IEEE Control Systems Letters paper presented at
2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

A trust-region method for data-driven iterative learning control of nonlinear systems

Jia Wang, Leander Hemelhof, Ivan Markovsky, and Panagiotis Patrinos

***Abstract*—This paper employs a derivative-free trust-region method to solve the norm-optimal iterative learning control problem for nonlinear systems with unknown dynamics. The iteration process is composed by two kinds of trials: main and additional trials. The tracking error is reduced in each main trial, and the additional trials explore the nonlinear dynamics around the main trial input. Then the trust-region subproblem is constructed based on the additional trial data, and solved to generate the next main trial input. The convergence of the tracking error is proved under mild assumptions. Our method is illustrated in simulations.**

***Index Terms*—Data-driven control, iterative learning control, nonlinear system control, derivative-free optimization**

I. Introduction

Norm-optimal iterative learning control (ILC) is a prevalent control method for repetitive tasks [1], [2], and has been successfully applied in many industrial sectors, for example free electron lasers [3], rehabilitation robotics [4] and batch reactors [5]. Based on optimization theory, norm-optimal ILC can learn from the previous trials, and update the input to steer the output to be as close as possible to the reference. More comprehensive review of norm-optimal ILC is in [6].

Access to an accurate model is crucial for norm-optimal ILC. As it can be difficult to obtain inexpensive but accurate representations of repetitive processes, data-driven norm-optimal ILC methods have been proposed. In [7], in order to improve the nominal model, the model parameters are estimated at the beginning of each trial based on previous data. Then the norm-optimal ILC method is used on the updated nominal model. In [5], the nonlinear dynamics between two trials is linearized by solving a least squares problem, which is used in the norm-optimal ILC design. However, above mentioned methods still require some prior knowledge of the actual system, for example the initial nominal model [7] and the sign of the partial derivative of unknown dynamics with respect to the input [5]. With the rapid development of technology, industrial processes become more and more complex. Consequently, identifying models or obtaining prior knowledge becomes more challenging. Thanks to the development of digital technology, a vast amount of process data can now be collected and stored. Therefore, we attempt to design a control law directly from this data to reduce the dependence of the identified model or prior knowledge, which is the main motivation of this paper.

The computation and implementation of norm-optimal ILC has been widely discussed. For example, algorithms such as steepest descent and Newton-Raphson [8], and more recently Quasi-Newton [9] and Nesterov acceleration [10] have been suggested to solve the norm-optimal ILC problem. The above mentioned methods require an identified or nominal model to evaluate the objective gradient. If the norm-optimal ILC can be implemented in a derivative-free manner, the model requirement can be removed from the controller design. However, little research has explored derivative-free implementations of norm-optimal ILC. There has been growing interest in the derivative-free optimization, which only requires objective values during the solution process. In [11], SPSA was proposed. The gradient is estimated based on the data and then the steepest descent method is used. In [12], GLIS was presented. The minimum of the objective is approximated by minimizing a sequence of data-driven surrogate functions. Although the above mentioned methods can be used to implement the norm-optimal ILC, there are some limitations. SPSA may converge slowly due to a lack of Hessian information, and GLIS may be time-consuming due to complicated surrogate functions. Recently, a derivative-free Gauss-Newton method was proposed in [13]. The local nonlinearity is captured by the linear interpolation, based on which a series of trust-region subproblems is constructed and solved. Second-order information is captured due to the estimated Jacobian of the tracking error, which can potentially result in a better convergence performance. Inspired by [13], we design a data-driven norm-optimal ILC method based on derivative-free trust-region methods to reduce the dependence of the model information. This constitutes our main contribution.

II. Problem statement

Consider a nonlinear discrete-time system:

$$\begin{cases} x_j(k+1) = f(x_j(k), u_j(k)) \\ y_j(k) = h(x_j(k)), \end{cases} \tag{1}$$

where $k = 0, \dots, N-1$ is the time index with N a positive integer (the length of one trial) and j is the trial index, and $x_j(k) \in \mathbb{R}^n$, $u_j(k) \in \mathbb{R}^m$ and $y_j(k) \in \mathbb{R}^p$ are the state, input and output. The mapping $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are assumed to be continuously differentiable but unknown.

The research leading to these results has received funding from: Fond for Scientific Research Vlaanderen (FWO) projects G033822N, G081222N, G0A0920N. European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 953348.

J. Wang, L. Hemelhof and P. Patrinos are with the Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, 3001 Leuven, Belgium. I. Markovsky is with the International Centre for Numerical Methods in Engineering (CIMNE), Gran Capità, 08034 Barcelona, Spain, and Catalan Institution for Research and Advanced Studies (ICREA), Pg. Lluis Companys 23, 08010 Barcelona, Spain. I. Markovsky is an ICREA professor at the CIMNE, Barcelona. (Email: jia.wang@esat.kuleuven.be; leander.hemelhof@kuleuven.be; panos.patrinos@esat.kuleuven.be; imarkovsky@cimne.upc.edu).

Assumption 1. *The initial value $x_j(0)$ is identical for each trial, i.e., $x_j(0) = x_0,\ \forall j \geq 0$.*

Denoting $\mathbf{U}_j = (u_j(0), \ldots, u_j(N-1)) \in \mathbb{R}^{mN}$ and $\mathbf{Y}_j = (y_j(1), \ldots, y_j(N)) \in \mathbb{R}^{pN}$, one obtains the lifted model:

$$\mathbf{Y}_j = \mathbf{G}(x_0, \mathbf{U}_j), \tag{2}$$

by propagating (1) starting from x_0, in which $\mathbf{G} : \mathbb{R}^n \times \mathbb{R}^{mN} \to \mathbb{R}^{pN}$ is an unknown nonlinear mapping. The reference trajectory is given as $\mathbf{R} = (r(1), \ldots, r(N)) \in \mathbb{R}^{pN}$ and the tracking error is defined by $\mathbf{E}_j := \mathbf{E}(\mathbf{U}_j) = \mathbf{R} - \mathbf{Y}_j = \mathbf{R} - \mathbf{G}(x_0, \mathbf{U}_j)$. If the nonlinear mapping $\mathbf{G}$ and the initial condition x_0 are known, the optimal tracking problem can be solved as a nonlinear least squares problem:

$$\underset{\mathbf{U}}{\text{minimize}}\ \ \ell(\mathbf{U}) = \frac{1}{2}\left\|\mathbf{R} - \mathbf{G}(x_0, \mathbf{U})\right\|^2 \tag{3}$$

in which $\|\cdot\|$ denotes the Euclidean norm. The minimizer of (3) is denoted as $\mathbf{U}^*$, which is the optimal input trajectory. However, $\mathbf{G}$ is usually unknown in practice, hence, the aim of this work is to design a data-driven ILC algorithm that updates $\mathbf{U}_{j+1}$ by using $\mathbf{U}_j$ and its *surrounding information*:

$$\mathbf{U}_{j+1} = C(\underbrace{\mathbf{U}_j}_{\substack{\downarrow\\ \text{main trial input}}}, \underbrace{\mathbf{U}_j^{(1)}, \ldots, \mathbf{U}_j^{(mN)}}_{\substack{\text{surrounding information:}\\ \text{additional trial input}}}, \underbrace{\mathbf{E}_j}_{\substack{\downarrow\\ \text{main trial error}}}, \underbrace{\mathbf{E}_j^{(1)}, \ldots, \mathbf{E}_j^{(mN)}}_{\substack{\text{surrounding information:}\\ \text{additional trial error}}}),$$

where the nonlinear mapping $C(\cdot)$ is to be designed. Denote the closed ball as $\mathbb{B}(\mathbf{z}_0, r) := \{\mathbf{z} \in \mathbb{R}^{mN} | \|\mathbf{z} - \mathbf{z}_0\| \leq r\}$. The additional trial input $\mathbf{U}_j^{(i)}$ belongs to the closed ball $\mathbb{B}(\mathbf{U}_j, \Delta_j)$, which is centered around $\mathbf{U}_j$ with radius $0 < \Delta_j \leq \Delta_{\max}$. Applying $\mathbf{U}_j^{(i)}$ to (1), the additional trial tracking error is obtained as $\mathbf{E}_j^{(i)} = \mathbf{E}(\mathbf{U}_j^{(i)})$. The pairs $(\mathbf{U}_j^{(i)}, \mathbf{E}_j^{(i)})$ constitute the surrounding information of $\mathbf{U}_j$. The generation of $\mathbf{U}_j^{(i)}$ will be discussed in the next section. The control goal is to drive the main trial tracking error to 0 by using the ILC algorithm.

III. Data-driven representation

A. Dynamical linearization

The dynamical relationship between two successive trials is

$$\mathbf{E}_{j+1} - \mathbf{E}_j = -\left(\mathbf{G}(x_0, \mathbf{U}_{j+1}) - \mathbf{G}(x_0, \mathbf{U}_j)\right). \tag{4}$$

The dynamical relationship (4) can be formulated as

$$\Delta\mathbf{E}_j = -\frac{\partial \mathbf{G}}{\partial \mathbf{U}}(\bar{\mathbf{U}}_j)\Delta\mathbf{U}_j$$

by using the differential mean value theorem, in which $\Delta\mathbf{E}_j = \mathbf{E}_{j+1} - \mathbf{E}_j$, $\Delta\mathbf{U}_j = \mathbf{U}_{j+1} - \mathbf{U}_j$, and $\bar{\mathbf{U}}_j$ is an unknown vector belonging to the interval $[\mathbf{U}_j, \mathbf{U}_{j+1}]$. The Jacobian matrix $\frac{\partial \mathbf{G}}{\partial \mathbf{U}}$ has a block lower triangular structure due to the causality of nonlinear system (1). Letting $\Phi_j = -\frac{\partial \mathbf{G}}{\partial \mathbf{U}}(\bar{\mathbf{U}}_j)$, we have

$$\mathbf{E}_{j+1} = \mathbf{E}_j + \Phi_j \Delta\mathbf{U}_j.$$

Since the matrix Φ_j is unknown, we use the first-order Taylor expansion to approximate $\mathbf{E}_{j+1}$ at the current input $\mathbf{U}_j$:

$$\mathbf{E}_{j+1} = \mathbf{E}(\mathbf{U}_j + \Delta\mathbf{U}_j) \approx \mathbf{E}_j + J(\mathbf{U}_j)\Delta\mathbf{U}_j,$$

where $J(\mathbf{U}_j) = -\frac{\partial \mathbf{G}}{\partial \mathbf{U}}(\mathbf{U}_j)$. Then, we replace $J(\mathbf{U}_j)$ with its estimate $J_j \in \mathbb{R}^{pN \times mN}$ as follows.

$$\mathbf{E}_{j+1} \approx \tilde{\mathbf{E}}_j(\Delta\mathbf{U}_j) := \mathbf{E}_j + J_j\Delta\mathbf{U}_j. \tag{5}$$

The construction of J_j from data is discussed in the next subsection. Denoting $\mathbb{L}(\mathbf{U}_0) = \{\mathbf{U} \in \mathbb{R}^{mN} | \ell(\mathbf{U}) \leq \ell(\mathbf{U}_0)\}$ and $\mathbb{B}(\mathbf{U}_0) = \bigcup_{\mathbf{U} \in \mathbb{L}(\mathbf{U}_0)} \mathbb{B}(\mathbf{U}, \Delta_{\max})$ for a given initial input $\mathbf{U}_0$, the following assumption is introduced.

Assumption 2. *The nonlinear dynamics $\mathbf{G}(x_0, \mathbf{U})$ is continuously differentiable and the Jacobian matrix $J(\mathbf{U})$ is Lipschitz continuous with constant L_J for all $\mathbf{U} \in \mathbb{B}(\mathbf{U}_0)$. The tracking error $\mathbf{E}(\mathbf{U})$ and Jacobian matrix $J(\mathbf{U})$ are uniformly bounded, i.e., $\|\mathbf{E}(\mathbf{U})\| \leq E_{\max}$ and $\|J(\mathbf{U})\| \leq J_{\max}$ in $\mathbb{B}(\mathbf{U}_0)$.*

Remark 1. *According to Lemma 3.2 in [13], Assumption 2 implies that the objective gradient $\nabla\ell(\mathbf{U})$ is Lipschitz continuous in $\mathbb{B}(\mathbf{U}_0)$ with constant $L_E := E_{\max}L_J + J_{\max}^2$.*

B. Linear interpolation

The surrounding information of $\mathbf{U}_j$ is denoted as the input set $\mathcal{U}_j := \{\mathbf{U}_j, \mathbf{U}_j^{(1)}, \ldots, \mathbf{U}_j^{(mN)}\}$ and the tracking error set $\mathcal{E}_j := \{\mathbf{E}_j, \mathbf{E}_j^{(1)}, \ldots, \mathbf{E}_j^{(mN)}\}$. Linear interpolation is used to capture the dynamics behind $\mathcal{U}_j$ and $\mathcal{E}_j$. The interpolation conditions are $\mathbf{E}(\mathbf{U}_j^{(i)}) = \tilde{\mathbf{E}}_j(\mathbf{U}_j^{(i)} - \mathbf{U}_j)$, or equivalently:

$$J_j(\mathbf{U}_j^{(i)} - \mathbf{U}_j) = \mathbf{E}_j^{(i)} - \mathbf{E}_j,\ i = 1, \ldots, mN \tag{6}$$

In this way, J_j can be obtained by solving a linear system:

$$\underbrace{\begin{pmatrix} (\mathbf{U}_j^{(1)} - \mathbf{U}_j)^T \\ \vdots \\ (\mathbf{U}_j^{(mN)} - \mathbf{U}_j)^T \end{pmatrix}}_{\mathbf{W}_j} J_j^T = \begin{pmatrix} (\mathbf{E}_j^{(1)} - \mathbf{E}_j)^T \\ \vdots \\ (\mathbf{E}_j^{(mN)} - \mathbf{E}_j)^T \end{pmatrix}. \tag{7}$$

If the square matrix $\mathbf{W}_j$ is invertible, J_j is determined uniquely, which needs the vectors $\{\mathbf{U}_j^{(1)} - \mathbf{U}_j, \ldots, \mathbf{U}_j^{(mN)} - \mathbf{U}_j\}$ to be linearly independent and such input set $\mathcal{U}_j$ is called "poised" [13]. In order to build a poised input set, we select $\mathbf{U}_j^{(i)}$ as

$$\mathbf{U}_j^{(i)} = \mathbf{U}_j + \xi^{(i)}\Delta_j\mathbf{e}^{(i)},\ i = 1, \ldots, mN \tag{8}$$

where $\xi^{(i)} = 1$ or -1 is chosen with equal probability, and $\mathbf{e}^{(i)} \in \mathbb{R}^{mN}$ is the unit vector which has 1 in the i-th element. Eq. (8) guarantees that: (i) the additional trial input $\mathbf{U}_j^{(i)}$ belongs to $\mathbb{B}(\mathbf{U}_j, \Delta_j)$ since $\|\mathbf{U}_j^{(i)} - \mathbf{U}_j\| = \Delta_j$, (ii) the matrix $\mathbf{W}_j$ is invertible since $\mathbf{U}_j^{(i)} - \mathbf{U}_j$ is a unit vector multiplied by scalar $\xi^{(i)}\Delta_j$, which makes $\mathbf{W}_j$ a diagonal matrix with non-zero diagonal elements, and (iii) the matrix J_j preserves the block lower triangular structure since each element in $\mathbf{U}_j$ is perturbed sequentially, which makes the matrix on the right-hand side of (7) block upper triangular due to the causality. According to Lemma 3.2 in [14], such invertibility is equivalent to the existence of the Lagrange polynomials. Since the Lagrange polynomials of the input set $\mathcal{U}_j$ play an important role in the ILC performance analysis, we introduce them as follows.

Definition 1. *Let $\mathbf{U}_j^{(0)} = \mathbf{U}_j$, consider the linear interpolation of a nonlinear function, the Lagrange polynomials of input*

set $\mathcal{U}_j = \{\mathbf{U}_j^{(0)}, \mathbf{U}_j^{(1)}, \ldots, \mathbf{U}_j^{(mN)}\}$ *with* $\mathbf{U}_j^{(i)} \in \mathbb{R}^{mN}$*, are the basis* $\{\mathbf{l}_0(\mathbf{U}), \ldots, \mathbf{l}_{mN}(\mathbf{U})\}$ *defined by*

$$\mathbf{l}_t(\mathbf{U}_j^{(i)}) = \begin{cases} 1, & \text{if } i = t \\ 0, & \text{if } i \neq t. \end{cases}, \quad i, t = 0, \ldots, mN$$

Algorithms 6.1 and 6.2 in [14] provide a recursive way to compute the Lagrange polynomials. On the other hand, a poised set $\mathcal{U}_j$ has the Λ-poisedness property, see the following definition, which is useful to bound $\|\mathbf{E}_{j+1} - \tilde{\mathbf{E}}_j(\Delta\mathbf{U}_j)\|$.

Definition 2. *Let* $\Lambda > 0$*. A poised input set* $\mathcal{U}_j$ *is* Λ*-poised in* $\mathbb{B}(\mathbf{U}_j, \Delta_j)$ *if the basis of Lagrange polynomials has*

$$\Lambda \geq \max_{t=0,\ldots,mN} \max_{\mathbf{U} \in \mathbb{B}(\mathbf{U}_j, \Delta_j)} |\mathbf{l}_t(\mathbf{U})|,$$

in the context of linear interpolation.

Since $\Delta_j \leq \Delta_{\max}$ for $j \geq 0$, there exists a constant $\Lambda > 0$ such that $\Lambda \geq \max\limits_{j \geq 0} \max\limits_{t=0,\ldots,mN} \max\limits_{\mathbf{U} \in \mathbb{B}(\mathbf{U}_j, \Delta_j)} |\mathbf{l}_t(\mathbf{U})|$. Therefore, the input set $\mathcal{U}_j$ is Λ-poised for all trials. After obtaining the matrix J_j from (7), we have the Gauss-Newton subproblem:

$$\underset{\Delta\mathbf{U}_j}{\text{minimize}} \ \tilde{\ell}_j(\Delta\mathbf{U}_j) = \frac{1}{2}\left\|\mathbf{E}_j + J_j\Delta\mathbf{U}_j\right\|^2,$$

which is equivalent to

$$\underset{\Delta\mathbf{U}_j}{\text{minimize}} \ \frac{1}{2}\|\mathbf{E}_j\|^2 + \left(J_j^T\mathbf{E}_j\right)^T\Delta\mathbf{U}_j + \frac{1}{2}\Delta\mathbf{U}_j^T J_j^T J_j \Delta\mathbf{U}_j. \quad (9)$$

The problem (9) will be used to construct the trust-region subproblem in the next section. In the following, we use notations $\ell(\mathbf{U}_j) := \frac{1}{2}\|\mathbf{E}_j\|^2$, $\tilde{\ell}_j(\Delta\mathbf{U}_j) := \frac{1}{2}\|\tilde{\mathbf{E}}_j(\Delta\mathbf{U}_j)\|^2$, $\mathbf{g}_j := J_j^T\mathbf{E}_j$ and $\mathbf{H}_j := J_j^T J_j$. Note that $\mathbf{g}_j$ is the estimated gradient at $\mathbf{U}_j$, i.e., $\mathbf{g}_j \approx \nabla\ell(\mathbf{U}_j) = J(\mathbf{U}_j)^T\mathbf{E}_j$.

Theorem 1. *Under Assumption 2, suppose the input set* $\mathcal{U}_j \subset \mathbb{B}(\mathbf{U}_j, \Delta_j)$ *is* Λ*-poised, then we have*

$$\left\|\mathbf{E}_{j+1} - \tilde{\mathbf{E}}_j(\Delta\mathbf{U}_j)\right\| \leq \alpha_1\Delta_j^2 \quad (10a)$$
$$\left\|J_j - J(\mathbf{U}_j + \Delta\mathbf{U}_j)\right\| \leq \alpha_2\Delta_j, \quad (10b)$$

for all $\|\Delta\mathbf{U}_j\| \leq \Delta_j$*, in which* $\alpha_1 = \frac{L_J}{2} + \alpha_2$*,* $\alpha_2 = \frac{L_J mN\Lambda}{2\sigma} + L_J$ *and a constant* $\sigma > 0$ *is independent of* $\mathcal{U}_j$ *and* Λ*.*

Proof. According to Assumption 2, one obtains

$$\left\|\mathbf{E}_{j+1} - \mathbf{E}_j - J(\mathbf{U}_j)\Delta\mathbf{U}_j\right\| \leq \frac{1}{2}L_J\left\|\Delta\mathbf{U}_j\right\|^2, \quad (11a)$$
$$\left\|\mathbf{E}_j^{(i)} - \mathbf{E}_j - J(\mathbf{U}_j)(\mathbf{U}_j^{(i)} - \mathbf{U}_j)\right\| \leq \frac{1}{2}L_J\left\|\mathbf{U}_j^{(i)} - \mathbf{U}_j\right\|^2 \leq \frac{1}{2}L_J\Delta_j^2, \quad (11b)$$

for $i = 1, \ldots, mN$. Substituting (6) to (11b) and multiplying both sides by $1/\Delta_j$, one obtains

$$\left\|\left(J_j - J(\mathbf{U}_j)\right)\frac{(\mathbf{U}_j^{(i)} - \mathbf{U}_j)}{\Delta_j}\right\| \leq \frac{1}{2}L_J\Delta_j. \quad (12)$$

Define the scaled data matrix as

$$M_j := \begin{pmatrix} \frac{\mathbf{U}_j^{(1)} - \mathbf{U}_j}{\Delta_j} & \frac{\mathbf{U}_j^{(2)} - \mathbf{U}_j}{\Delta_j} & \cdots & \frac{\mathbf{U}_j^{(mN)} - \mathbf{U}_j}{\Delta_j} \end{pmatrix} \in \mathbb{R}^{mN \times mN},$$

which implies that columns of M_j belong to $\mathbb{B}(\mathbf{0}, 1)$. We have

$$\begin{aligned} \left\|\left(J_j - J(\mathbf{U}_j)\right)M_j\right\|^2 &\leq \left\|\left(J_j - J(\mathbf{U}_j)\right)M_j\right\|_F^2 \\ &= \sum_{i=1}^{mN}\left\|\left(J_j - J(\mathbf{U}_j)\right)\frac{\mathbf{U}_j^{(i)} - \mathbf{U}_i}{\Delta_j}\right\|^2, \end{aligned} \quad (13)$$

and then

$$\begin{aligned} \left\|J_j - J(\mathbf{U}_j)\right\| &= \left\|\left(J_j - J(\mathbf{U}_j)\right)M_jM_j^{-1}\right\| && (14a)\\ &\leq \left\|\left(J_j - J(\mathbf{U}_j)\right)M_j\right\|\left\|M_j^{-1}\right\| && (14b)\\ &\overset{(13)}{\leq} \sqrt{\sum_{i=1}^{mN}\left\|\left(J_j - J(\mathbf{U}_j)\right)\frac{\mathbf{U}_j^{(i)} - \mathbf{U}_i}{\Delta_j}\right\|^2}\left\|M_j^{-1}\right\| && (14c)\\ &\overset{(12)}{\leq} \frac{1}{2}L_J\sqrt{mN}\left\|M_j^{-1}\right\|\Delta_j, && (14d) \end{aligned}$$

in which the invertibility of M_j follows from (8). The bound of $\|M_j^{-1}\|$ is discussed as follows. Define a mapping $\phi : \mathbb{R}^{mN} \to \mathbb{R}^{mN+1}$ as $\phi(\mathbf{U}) := (1, \mathbf{U})$, which in fact is the natural basis with polynomial degree 1, see formula (3.1) in [14]. Given an input trajectory $\tilde{\mathbf{U}} \in \mathbb{B}(\mathbf{U}_j, \Delta_j)$, according to Definition 3.6 in [14], the Λ-poisedness property of the input set $\mathcal{U}_j$ implies the existence of a vector $\tilde{\varrho} := (\tilde{\varrho}_0, \ldots, \tilde{\varrho}_{mN}) \in \mathbb{R}^{mN+1}$ with $\|\tilde{\varrho}\|_\infty \leq \Lambda$ such that $\sum_{i=0}^{mN}\phi(\mathbf{U}_j^{(i)})\tilde{\varrho}_i = \phi(\tilde{\mathbf{U}})$, i.e.,

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{U}_j & \mathbf{U}_j^{(1)} & \cdots & \mathbf{U}_j^{(mN)} \end{pmatrix}\tilde{\varrho} = \begin{pmatrix} 1 \\ \tilde{\mathbf{U}} \end{pmatrix}. \quad (15)$$

On the other hand, Lemmas 3.8 and 3.9 in [14] show that the scaling and shift of the input set $\mathcal{U}_j$ do not change the value of $\tilde{\varrho}$, hence, using $1/\Delta_j$ and $\mathbf{U}_j$ for scaling and shift, the vector $\tilde{\varrho}$ in (15) also holds

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \frac{\mathbf{U}_j - \mathbf{U}_j}{\Delta_j} & \frac{\mathbf{U}_j^{(1)} - \mathbf{U}_j}{\Delta_j} & \cdots & \frac{\mathbf{U}_j^{(mN)} - \mathbf{U}_j}{\Delta_j} \end{pmatrix}\tilde{\varrho} = \begin{pmatrix} 1 \\ \frac{\tilde{\mathbf{U}} - \mathbf{U}_j}{\Delta_j} \end{pmatrix},$$

which implies that there exists a vector $\varrho := (\tilde{\varrho}_1, \ldots, \tilde{\varrho}_{mN}) \in \mathbb{R}^{mN}$ such that $M_j\varrho = \mathbf{U}$ with $\|\varrho\|_\infty \leq \Lambda$ for a given $\mathbf{U} = \frac{\tilde{\mathbf{U}} - \mathbf{U}_j}{\Delta_j} \in \mathbb{B}(\mathbf{0}, 1)$, then we have $\|\varrho\|_\infty = \|M_j^{-1}\mathbf{U}\|_\infty \leq \Lambda$. Define a function $\psi(v) := \max\limits_{\mathbf{U} \in \mathbb{B}(\mathbf{0},1)} |v^T\mathbf{U}|$ with $v \in \mathbb{R}^{mN}$ and a positive constant $\sigma := \min\limits_{\|v\|=1} \psi(v)$. Let $\|\bar{v}\| = 1$, $\tilde{v} = \beta\bar{v}$ and $\beta > 0$, we have $\psi(\tilde{v}) = \psi(\beta\bar{v}) = \beta\psi(\bar{v}) = \|\tilde{v}\|\psi(\bar{v}) \geq \|\tilde{v}\|\sigma$ for all $\tilde{v} \in \mathbb{R}^{mN}$, which implies that there exists $\mathbf{U} \in \mathbb{B}(\mathbf{0}, 1)$ such that $|\tilde{v}^T\mathbf{U}| \geq \sigma$ if $\|\tilde{v}\| = 1$. Let $\tilde{v}$ be the normalized eigenvector of M_j^{-1} corresponding to the largest singular value, Lemma 3.13 in [14] shows that $\|M_j^{-1}\mathbf{U}\| \geq |\tilde{v}^T\mathbf{U}|\|M_j^{-1}\| \geq \sigma\|M_j^{-1}\|$, where the left item can be bounded by $\|M_j^{-1}\mathbf{U}\| \leq \sqrt{mN}\|M_j^{-1}\mathbf{U}\|_\infty$. In this way, we have $\|M_j^{-1}\| \leq \sqrt{mN}\Lambda/\sigma$. Combing with (14d), the estimate error of Jacobian is

$$\left\|J_j - J(\mathbf{U}_j)\right\| \leq \frac{L_J mN\Lambda}{2\sigma}\Delta_j, \quad (16)$$

which leads to

$$\begin{aligned} \left\|J_j - J(\mathbf{U})\right\| &\leq \left\|J_j - J(\mathbf{U}_j)\right\| + \left\|J(\mathbf{U}_j) - J(\mathbf{U})\right\| && (17a)\\ &\leq \left(\tfrac{L_J mN\Lambda}{2\sigma} + L_J\right)\Delta_j, && (17b) \end{aligned}$$

for all $\mathbf{U} \in \mathbb{B}(\mathbf{U}_j, \Delta_j)$. As for the tracking error, we have

$$\begin{aligned} &\|\mathbf{E}_{j+1} - \tilde{\mathbf{E}}_j(\Delta\mathbf{U}_j)\| \\ &\overset{(5)}{\leq} \|\mathbf{E}_{j+1} - \mathbf{E}_j - J(\mathbf{U}_j)\Delta\mathbf{U}_j\| + \|J(\mathbf{U}_j)\Delta\mathbf{U}_j - J_j\Delta\mathbf{U}_j\| \\ &\overset{(11a)}{\leq} \frac{1}{2}L_J\|\Delta\mathbf{U}_j\|^2 + \|J(\mathbf{U}_j) - J_j\|\|\Delta\mathbf{U}_j\| \\ &\overset{(17b)}{\leq} \frac{1}{2}L_J\Delta_j^2 + \left(\tfrac{L_J mN\Lambda}{2\sigma} + L_J\right)\Delta_j^2. \end{aligned}$$

The proof is completed. □

Let $\Delta\mathbf{U}_j = \mathbf{0}$ in (10b), Theorem 1 immediately leads to

$$\|\nabla\ell(\mathbf{U}_j) - \mathbf{g}_j\| \le \|J(\mathbf{U}_j) - J_j\|\|\mathbf{E}_j\| \le \alpha_2 E_{\max}\Delta_j. \quad (18)$$

Let $\alpha_3 = \alpha_2 E_{\max}$, the inequality (18) will be used to prove the convergence of trust-region ILC in the next section.

IV. ILC DESIGN

In this work, the norm-optimal ILC updates the main trial input trajectory by solving the trust-region subproblem:

$$\underset{\Delta\mathbf{U}_j}{\text{minimize}} \ \frac{1}{2}\Delta\mathbf{U}_j^T\mathbf{H}_j\Delta\mathbf{U}_j + \mathbf{g}_j^T\Delta\mathbf{U}_j \ \text{ s.t. } \|\Delta\mathbf{U}_j\| \le \Delta_j. \quad (19)$$

Then apply $\mathbf{U}_j + \Delta\mathbf{U}_j$ to (1), if the resulting performance is acceptable, the candidate input moves from $\mathbf{U}_j$ to $\mathbf{U}_j + \Delta\mathbf{U}_j$, otherwise, keep the candidate input as $\mathbf{U}_j$ and reduce the radius Δ_j. The tracking error reduction is measured by

$$r_j = \frac{\ell(\mathbf{U}_j) - \ell(\mathbf{U}_j + \Delta\mathbf{U}_j)}{\tilde{\ell}_j(\mathbf{0}) - \tilde{\ell}_j(\Delta\mathbf{U}_j)}. \quad (20)$$

The following assumption is required for convergence, which is usually satisfied by solving subproblem (19) properly [13].

Assumption 3. *Let $\Delta\mathbf{U}_j$ be the descent direction after solving (19) at the j-th trial, then*

$$\tilde{\ell}_j(\mathbf{0}) - \tilde{\ell}_j(\Delta\mathbf{U}_j) \ge \frac{c}{2}\|\mathbf{g}_j\| \min\left\{\Delta_j, \frac{\|\mathbf{g}_j\|}{\|\mathbf{H}_j\|}\right\}, \ c \in (0, 1].$$

The trust-region-based norm-optimal ILC (TRILC) is summarized in Algorithm 1. Step 3 calculates the estimated gradient $\mathbf{g}_j^{\text{ini}}$ and Hessian $\mathbf{H}_j^{\text{ini}}$ from surrounding information. If $\|\mathbf{g}_j^{\text{ini}}\|$ is smaller than a threshold ε_g, the *criticality phase* in steps $5-9$ is invoked, which is explained as follows. If the estimated gradient is close to $\mathbf{0}$, then the true gradient $\nabla\ell(\mathbf{U}_j)$ could potentially be small, which implies the main trial input $\mathbf{U}_j$ may be close to a stationary point. Hence, the local exploration should be restricted to a small area, in order to result in a more accurate data-driven representation. After obtained $\mathbf{g}_j$ and $\mathbf{H}_j$, the subproblem (19) is solved to satisfy Assumption 3. Step 17 manages the trust-region radius and input for the next main trial according to r_j: If $r_j \ge \eta_2$, the data-driven representation describes the local nonlinearity well, so $\Delta\mathbf{U}_j$ is a successful input increment and it is safe to enlarge Δ_j to $\gamma_{\text{inc}}\Delta_j$. If $\eta_1 \le r_j < \eta_2$, the data-driven representation cannot capture the local nonlinearity accurately, but the tracking error is still reduced, so the input increment $\Delta\mathbf{U}_j$ is acceptable, but Δ_j is reduced to $\gamma_{\text{dec}}\Delta_j$. If $r_j < \eta_1$, the data-driven representation is invalid, so the exploration for the next main trial is still centered around $\mathbf{U}_j$, but with a reduced radius $\gamma_{\text{dec}}\Delta_j$.

Lemma 1. *If $\|\nabla\ell(\mathbf{U}_j)\| \ne 0$, steps $5-9$ in Algorithm 1 will terminate in a finite number of loops.*

Proof. Assume steps $5-9$ is running infinitely, we have $(\omega_c)^d\Delta_j^{\text{ini}} \ge \mu\|\mathbf{g}_j^{[d]}\|$ for all $d \ge 0$. From (18), it holds that $\|\nabla\ell(\mathbf{U}_j) - \mathbf{g}_j^{[d]}\| \le \alpha_3(\omega_c)^d\Delta_j^{\text{ini}}$ for all $d \ge 0$, then we have

$$\|\nabla\ell(\mathbf{U}_j)\| \le \|\nabla\ell(\mathbf{U}_j) - \mathbf{g}_j^{[d]}\| + \|\mathbf{g}_j^{[d]}\| \le \alpha_3(\omega_c)^d\Delta_j^{\text{ini}} + \frac{1}{\mu}(\omega_c)^d\Delta_j^{\text{ini}},$$

which implies that $\|\nabla\ell(\mathbf{U}_j)\| = 0$ since $\omega_c \in (0,1)$ and $d \to \infty$. A contradiction occurs with $\|\nabla\ell(\mathbf{U}_j)\| \ne 0$. Therefore, steps $5-9$ will terminate finitely. □

Algorithm 1 Trust-region-based norm-optimal ILC (TRILC)

Input: Initial input $\mathbf{U}_0$, initial trust-region radius $\Delta_0^{\text{ini}} > 0$, radius upper bound $\Delta_{\max}$, radius scalings $0 < \gamma_{\text{dec}} < 1 < \gamma_{\text{inc}}$, gradient threshold ε_g, gradient scaling $\mu > 0$, criticality radius reduction $\omega_c \in (0,1)$, acceptance threshold $0 < \eta_1 \le \eta_2 < 1$ and accuracy tolerance $\varepsilon > 0$.

Output: ILC input sequence $\{\mathbf{U}_j\}$.

1: **for** $j = 0, 1, 2, \ldots$ **do**
2: Construct input set $\mathcal{U}_j$ in $\mathbb{B}(\mathbf{U}_j, \Delta_j^{\text{ini}})$, apply inputs in $\mathcal{U}_j$ to system (1) and construct tracking error set $\mathcal{E}_j$.
3: Solve (7), and obtain $\mathbf{g}_j^{\text{ini}}$ and $\mathbf{H}_j^{\text{ini}}$.
4: **if** $\|\mathbf{g}_j^{\text{ini}}\| \le \varepsilon_g$ **then** ▷ criticality phase
5: **for** $d = 0, 1, 2, \ldots$ **do**
6: If $(\omega_c)^d\Delta_j^{\text{ini}} \le \mu\|\mathbf{g}_j^{[d]}\|$, then stop and let $\mathcal{U}_j = \mathcal{U}_j^{[d]}$, $\mathbf{g}_j = \mathbf{g}_j^{[d]}$, $\mathbf{H}_j = \mathbf{H}_j^{[d]}$ and $\Delta_j = (\omega_c)^d\Delta_j^{\text{ini}}$ (note that $\mathbf{g}_j^{[0]} = \mathbf{g}_j^{\text{ini}}$ and $\mathbf{H}_j^{[0]} = \mathbf{H}_j^{\text{ini}}$).
7: Construct $\mathcal{U}_j^{[d+1]}$ via (8) in $\mathbb{B}(\mathbf{U}_j, (\omega_c)^{d+1}\Delta_j^{\text{ini}})$, apply inputs in $\mathcal{U}_j^{[d+1]}$ to (1) and construct $\mathcal{E}_j^{[d+1]}$.
8: Solve (7), obtain $\mathbf{g}_j^{[d+1]}$ and $\mathbf{H}_j^{[d+1]}$.
9: **end for**
10: **else**
11: $\mathbf{g}_j = \mathbf{g}_j^{\text{ini}}$, $\mathbf{H}_j = \mathbf{H}_j^{\text{ini}}$ and $\Delta_j = \Delta_j^{\text{ini}}$.
12: **end if**
13: Solve trust-region subproblem (19) and obtain $\Delta\mathbf{U}_j$.
14: Apply $\mathbf{U}_j + \Delta\mathbf{U}_j$ to (1) and obtain $\mathbf{E}(\mathbf{U}_j + \Delta\mathbf{U}_j)$.
15: If $\ell(\mathbf{U}_j + \Delta\mathbf{U}_j) \le \varepsilon$, then stop algorithm.
16: Calculate ratio r_j by (20). ▷ evaluation of $\Delta\mathbf{U}_j$
17: Update input and trust-region radius as

$$(\mathbf{U}_{j+1}, \Delta_{j+1}^{\text{ini}}) = \begin{cases} (\mathbf{U}_j + \Delta\mathbf{U}_j, \min\{\gamma_{\text{inc}}\Delta_j, \Delta_{\max}\}), \ \text{if } r_j \ge \eta_2 \\ \qquad\qquad \triangleright \text{ successful input increment } \Delta\mathbf{U}_j \\ (\mathbf{U}_j + \Delta\mathbf{U}_j, \gamma_{\text{dec}}\Delta_j), \ \text{if } \eta_1 \le r_j < \eta_2 \\ \qquad\qquad \triangleright \text{ acceptable input increment } \Delta\mathbf{U}_j \\ (\mathbf{U}_j, \gamma_{\text{dec}}\Delta_j), \ \text{if } r_j < \eta_1 \\ \qquad\qquad \triangleright \text{ unacceptable input increment } \Delta\mathbf{U}_j \end{cases}$$

18: **end for**

Note that algorithms in [13] and [14] are designed for general optimization problems, in contrast, Algorithm 1 is a tailored method for data-driven ILC. There are two main differences between them from the engineering point of view. (i) Geometry-improving phase is removed from Algorithm 1, i.e., the value of Λ (see Definition 2) is not suppressed to below a given value. (ii) Safety phase is removed from Algorithm 1, i.e., the input $\mathbf{U}_j + \Delta\mathbf{U}_j$ is used even if $\|\Delta\mathbf{U}_j\|$ is small. To see (i), steps 2 and 7 construct the input set by (8), without running Algorithm 6.3 in [14]. The rationale behind this modification is to reduce the computation time in each main trial. If using Algorithm 6.3

to generate an input set that has a small Λ value, at least mN subproblems have to be solved in steps 2 and 7, which can be time-consuming for some applications. Hence, we replace this sophisticatedly tuned Λ with an unknown but bounded one, which simplifies the ILC algorithm without breaking the global convergence. As for (ii), although the input $\mathbf{U}_j + \Delta\mathbf{U}_j$ may not improve the performance sufficiently, the tracking error is still possibly reduced, which can be meaningful in practice, therefore, we keep the small increment $\Delta\mathbf{U}_j$. However, if $\mathbf{U}_j + \Delta\mathbf{U}_j$ gives an unacceptable performance, it will be discarded and the input will be reset to $\mathbf{U}_j$, see step 17.

The main trial convergence of TRILC is presented next.

Theorem 2. *Under Assumptions 1, 2 and 3, suppose the sequence $\{\mathbf{U}_j\}$ is generated by Algorithm 1, then we have*

$$\lim_{j\to\infty} \|\nabla\ell(\mathbf{U}_j)\| = 0. \tag{21}$$

Proof. First, we show (21) holds in the case that the index set $\mathcal{R} = \{j | r_j \geq \eta_2\}$ is finite. Let i be any main trial index after $\bar{j} = \max\{j | j \in \mathcal{R}\}$, and note that Δ_i can only be reduced if $r_j < \eta_2$, we have $\lim_{i\to\infty} \|\mathbf{U}_{\bar{j}} - \mathbf{U}_i\| = 0$ and $\lim_{i\to\infty} \Delta_i = 0$. Consider the inequality:

$$\begin{aligned}\|\nabla\ell(\mathbf{U}_{\bar{j}})\| &\leq \|\nabla\ell(\mathbf{U}_{\bar{j}}) - \nabla\ell(\mathbf{U}_i)\| + \|\nabla\ell(\mathbf{U}_i) - \mathbf{g}_i\| + \|\mathbf{g}_i\| \\ &\overset{(18)}{\leq} L_E\|\mathbf{U}_{\bar{j}} - \mathbf{U}_i\| + \alpha_3\Delta_i + \|\mathbf{g}_i\|,\end{aligned}$$

in which the first and second terms on the right-hand converge to 0. If the third term $\|\mathbf{g}_i\|$ does not converge to 0, according to Lemma 10.7 in [14] and Assumption 3, there exists $\bar{\kappa} > 0$ such that $\lim_{i\to\infty} \Delta_i \geq \bar{\kappa}$, which yields a contradiction with $\lim_{i\to\infty} \Delta_i = 0$. Thus, Eq. (21) holds in the case that $\mathcal{R} = \{j | r_j \geq \eta_2\}$ is finite. According to Theorem 1 and Assumption 2, the estimated Jacobian matrix J_j can be bounded as

$$\|J_j\| \leq \|J_j - J(\mathbf{U}_j)\| + \|J(\mathbf{U}_j)\| \overset{(16)}{\leq} \frac{L_J mN\Lambda}{2\sigma}\Delta_{\max} + J_{\max},$$

thus, we know that $\|\mathbf{H}_j\|$ is uniformly bounded for all main trial j since $\mathbf{H}_j = J_j^T J_j$, i.e., $\exists L_H > 0$ such that $\|\mathbf{H}_j\| \leq L_H$. Then, combining the above results with Theorem 10.13 in [14], Eq. (21) in the case that $\mathcal{R} = \{j | r_j \geq \eta_2\}$ is infinite can be proved, which leads to the result in Theorem 2. □

Algorithm 2 provides a heuristic way to reduce additional trials. The idea is to reuse the previous surrounding information to describe the local nonlinear dynamics around $\mathbf{U}_j$. With slight abuse of notation, the element $\mathbf{U}_{j-1}^{(i)}$ in the data set $\mathcal{U}_j$ is updated via (8) if $\mathbf{U}_{j-1}^{(i)} \notin \mathbb{B}(\mathbf{U}_j, \Delta_j)$, see steps $1-5$. In such case, the previous additional trial input $\mathbf{U}_{j-1}^{(i)}$ is far away from $\mathbf{U}_j$ and may not be able to capture the local nonlinearity. If all previous additional inputs $\mathbf{U}_{j-1}^{(i)}$ do not belong to $\mathbb{B}(\mathbf{U}_j, \Delta_j)$, mN additional trials are required (unsuccessful reduction). If only partial elements are updated in steps $1-5$, the invertibility of $\mathbf{W}_j$ is checked. An invertible $\mathbf{W}_j$ implies that the set $\mathcal{U}_j$ is poised, then no more additional trials are required (successful reduction). Otherwise, step 11 constructs the column index set $\mathcal{I}_j$ for linearly independent elements in $\mathcal{U}_j$, step 12 updates the input $\mathbf{U}_{j-1}^{(i)}$ via (8) for those indexes $i \notin \mathcal{I}_j$. Step 12 requires $mN - |\mathcal{I}_j|$ additional trials, if additional trials run by steps $1-5$ is less than $|\mathcal{I}_j|$, then the trial reduction is successful.

Algorithm 2 Additional trial reduction for the j-th main trial

Input: Data sets $\mathcal{U}_j = \{\mathbf{U}_j, \mathbf{U}_{j-1}^{(1)} \ldots, \mathbf{U}_{j-1}^{(mN)}\}$ and $\mathcal{E}_j = \{\mathbf{E}_j, \mathbf{E}_{j-1}^{(1)}, \ldots, \mathbf{E}_{j-1}^{(mN)}\}$, and trust-region radius Δ_j.
Output: Updated data sets $\mathcal{U}_j$ and $\mathcal{E}_j$.
1: **for** $i = 1, \ldots, mN$ **do**
2: **if** $\|\mathbf{U}_{j-1}^{(i)} - \mathbf{U}_j\| > \Delta_j$ **then** ▷ far away from $\mathbf{U}_j$
3: Replace $\mathbf{U}_{j-1}^{(i)}$ with $\mathbf{U}_j^{(i)} = \mathbf{U}_j + \xi^{(i)}\Delta_j\mathbf{e}^{(i)}$ in $\mathcal{U}_j$, apply $\mathbf{U}_j^{(i)}$ to (1), replace $\mathbf{E}_{j-1}^{(i)}$ with $\mathbf{E}_j^{(i)}$ in $\mathcal{E}_j$.
4: **end if**
5: **end for**
6: If all elements in $\mathcal{U}_j$ and $\mathcal{E}_j$ are updated, then stop algorithm, return $\mathcal{U}_j$ and $\mathcal{E}_j$. ▷ unsuccessful
7: Calculate square matrix $\mathbf{W}_j$ in (7) based on $\mathcal{U}_j$ and $\mathcal{E}_j$.
8: **if** square matrix $\mathbf{W}_j$ has full rank **then** ▷ successful
9: Stop algorithm, return $\mathcal{U}_j$ and $\mathcal{E}_j$.
10: **else** ▷ possibly successful
11: Record linearly independent column indexes to set $\mathcal{I}_j$.
12: For $i = 1, \ldots, mN$, if $i \notin \mathcal{I}_j$, then run step 3.
13: Return $\mathcal{U}_j$ and $\mathcal{E}_j$.
14: **end if**

Remark 2. *Some additional trials are required to stimulate the system, this is the cost to pay if no model information is used. Potential application scenarios for TRILC are industrial plants with complex nonlinearity or relatively fast trials, for example, a high-DOF robotic arm in the manufacturing industry [15], or a micro-batch reactor in chemical engineering [16].*

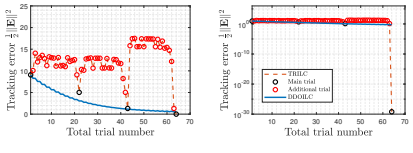
V. Simulation

A. Linear time-invariant system

The LTI system in [5] is considered as follows.

$$\begin{cases} x_j(k+1) = \begin{pmatrix} 0 & 1 \\ -0.5 & -0.5 \end{pmatrix} x_j(k) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u_j(k) \\ y_j(k) = \begin{pmatrix} 1, 0 \end{pmatrix} x_j(k). \end{cases}$$

The reference is $r(k) = 10^{-6}(k-1)^3(4 - 0.03(k-1))$ and $N = 20$. The initial states are $x_{1,0} = 1$ and $x_{2,0} = 0$. The parameters in Algorithm 1 are selected as follows: $\varepsilon = 0.01$, $\Delta_0^{\text{ini}} = 2$, $\Delta_{\max} = 10$, $\eta_1 = 0.01$, $\eta_2 = 0.9$, $\gamma_{\text{dec}} = 0.5$, $\gamma_{\text{inc}} = 1.5$, $\varepsilon_g = 0.01$, $\mu = 1$ and $\omega_c = 0.9$.



(a) (b)

Fig. 1. Comparison between TRILC and DDOILC. (a) Tracking error in linear scale. (b) Tracking error in logarithmic scale.

In Fig. 1, TRILC and data-driven optimal ILC (DDOILC) in [5] are compared. The tracking errors of TRILC in main

and additional trials are denoted by black and red dots. Note that the goal of the additional trials is to explore the local behavior of the system. Only the main trials are expected to improve the tracking error, which interprets the worse transition performance compared to DDOILC. However, DDOILC reduces the tracking error slowly after the 60-th trial. In contrast, TRILC reduces the tracking error significantly at the 64-th trial and then achieves the accuracy tolerance.

B. Continuous stirred tank reactor

In this part, TRILC is tested on nonlinear CSTR model:

$$\begin{cases} x_{1,j}(k+1) = (1-T_s\theta_s)x_{1,j}(k) + T_sD(1-x_{1,j}(k))e^{\frac{x_{2,j}(k)}{1+\frac{x_{2,j}(k)}{\kappa}}} \\ x_{2,j}(k+1) = (1-T_s\theta_s)x_{2,j}(k) + T_sBD(1-x_{1,j}(k))e^{\frac{x_{2,j}(k)}{1+\frac{x_{2,j}(k)}{\kappa}}} \\ \qquad\qquad - T_s\chi x_{2,j}(k) + T_s\chi u_j(k-\tau_d), \end{cases}$$

where $T_s = 0.1$, $\tau_d = 1$, $\theta_s = 1$, $\chi = 0.3$, $\kappa = 20$, $B = 1$ and $D = 0.072$. The state $x_{2,j}$ is the system output, and the goal is driving $x_{2,j}$ to a set-point by adjusting u_j. The above nonlinear model is obtained by using a sampling time of 0.1s and Euler approximation, which is a widely used benchmark for industrial control problems. For details see [17].

The set-point is $r = 1.96$, initial states are $x_{1,0} = 0.57$ and $x_{2,0} = 0.3$, and $N = 100$. Note that the nonlinear model is continuously differentiable, and the nonlinearity comes from exponential terms. Therefore, it is easy to show that Assumption 2 is valid on CSTR model. Let $\Delta_0^{\text{ini}} = 10$ and $\Delta_{\max} = 30$. Other parameters are the same as in the above subsection.

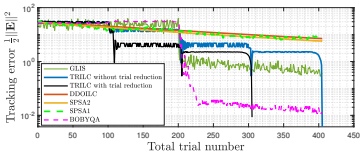


Fig. 2. Reactor temperature tracking performance with different methods.

Seven methods are compared: TRILC (with/without trial reduction), DDOILC, two points SPSA (SPSA2) [11], one point SPSA (SPSA1) [18], GLIS [12] and BOBYQA [19], as shown in Fig. 2. Since the information from only one and two trials are used by DDOILC and SPSA2, the nonlinearity may not be described accurately, hence, their convergence can be slow. SPSA1 only uses one additional trial to estimate gradients, which is less robust than SPSA2 in CSTR benchmark.

GLIS requires $2mN$ initial trials to start the optimization, i.e., trials 1 to 200, in which initial inputs are selected randomly. After initialization, GLIS shows a better performance compared to additional trials of TRILC. However, GLIS reduces the tracking error slowly and cannot outperform the main trials of TRILC. At the 405-th trial, TRILC (without trial reduction) stops with the tracking error 0.0038, but GLIS yields 0.356.

BOBYQA uses the quadratic interpolation to capture the local nonlinearity. The tracking error is reduced significantly after the 201-st trial. However, BOBYQA reduces the tracking error slowly from trial 228 to 405. At the 405-th trial, BOBYQA yields a tracking error 0.016, which is worse than TRILC.

The original TRILC is also compared to TRILC with trial reduction. The first batch (trial 1 to 101) of two methods is the same, which leads to the same input for the 2-nd main trial. Starting from the 102-nd trial, TRILC (without trial reduction) runs 100 new additional trials, and then reduces the tracking error at the 203-rd trial. In contrast, TRILC (with trial reduction) reuses the data set of the first batch, only 5 additional trials are required and the tracking error is reduced significantly at the 108-th trial. TRILC (with trial reduction) saves 100 trials in total and the effectiveness of Algorithm 2 is shown.

REFERENCES

[1] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control using optimal feedback and feedforward actions," *International Journal of Control*, vol. 65, no. 2, pp. 277–293, 1996.

[2] ——, "Iterative learning control for discrete-time systems with exponential rate of convergence," *IEE Proceedings-Control Theory and Applications*, vol. 143, no. 2, p. 217–224, 1996.

[3] S. Kichhoff, C. Schmidt, G. Lichtenberg, and H. Werner, "An iterative learning algorithm for control of an accelerator based free electron laser," in *47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008, pp. 3032–3037.

[4] C. T. Freeman, "Upper limb electrical stimulation using input-output linearization and iterative learning control," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, p. 1546–1554, 2015.

[5] R. Chi, Z. Hou, B. Huang, and S. Jin, "A unified data-driven design framework of optimality-based generalized iterative learning control," *Computers & Chemical Engineering*, vol. 77, pp. 10–23, 2015.

[6] E. Rogers, D. H. Owens, H. Werner, and et al., "Norm optimal iterative learning control with application to problems in accelerator based free electron lasers and rehabilitation robotics," *European Journal of Control*, vol. 5, p. 496–521, 2010.

[7] M. Volckaert, M. Diehl, and J. Swevers, "Generalization of norm optimal ILC for nonlinear systems with constraints," *Mechanical Systems and Signal Processing*, vol. 39, pp. 280–296, 2013.

[8] M. Togai and O. Yamano, "Analysis and design of an optimal learning control scheme for industrial robots: A discrete system approach," in *24th IEEE Conference on Decision and Control*, Fort Lauderdale, USA, 1985, pp. 1399–1404.

[9] H. Tao, W. Paszke, E. Rogers, and et al., "Modified Newton method based iterative learning control design for discrete nonlinear systems with constraints," *Systems & Control Letters*, vol. 118, pp. 35–43, 2018.

[10] J. Wang, L. Hemelhof, I. Markovsky, and P. Patrinos, "Fast data-driven iterative learning control for linear system with output disturbance," *arXiv:2312.14326*, 2023.

[11] J. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, 1992.

[12] A. Bemporad, "Global optimization via inverse distance weighting and radial basis functions," *Computational Optimization and Applications*, vol. 77, p. 571–595, 2020.

[13] C. Cartis and L. Roberts, "A derivative-free Gauss–Newton method," *Mathematical Programming Computation*, vol. 11, p. 631–674, 2019.

[14] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*. Philadelphia: MPS/SIAM, 2009.

[15] F. Rothling, R. Haschke, J. J. Steil, and H. Ritter, "Platform portable anthropomorphic grasping with the bielefeld 20-DOF shadow and 9-DOF TUM hand," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007, pp. 2951–2956.

[16] M. Kashid, O. Detraz, M. S. Moya, and et al., "Micro-batch reactor for catching intermediates and monitoring kinetics of rapid and exothermic homogeneous reactions," *Chemical Engineering Journal*, vol. 214, p. 149–156, 2013.

[17] F. Wu, "LMI-based robust model predictive control and its application to an industrial CSTR problem," *Journal of Process Control*, vol. 11, no. 6, pp. 649–659, 2001.

[18] J. Spall, "A one-measurement form of simultaneous perturbation stochastic approximation," *Automatica*, vol. 33, no. 1, pp. 109–112, 1997.

[19] M. J. D. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives," *Cambridge NA Report NA2009/06, University of Cambridge*, 2009.