On Iterative Learning Control for Nonlinear Discrete-Time Systems with Input and Output Constraints

1st Zeyu Li
Department of Mechanical Engineering
The University of Melbourne
Parkville VIC 3010 Australia
zeyul5@student.unimelb.edu.au

2nd Gijo Sebastian
Department of Mechanical Engineering
The University of Melbourne
Parkville VIC 3010 Australia
gijos@unimelb.edu.au

3rd Ying Tan
Department of Mechanical Engineering
The University of Melbourne
Parkville VIC 3010 Australia
yingt@unimelb.edu.au

4th Denny Oetomo
Department of Mechanical Engineering
The University of Melbourne
Parkville VIC 3010 Australia
doetomo@unimelb.edu.au

***Abstract*—Iterative learning control (ILC) is a class of data-driven techniques, which can "*learn*" the unknown desired control input signal to accurately track a desired output trajectory using input and output measurements when the tracking task is performed over a finite time interval repetitively. Although many ILC algorithms have been proposed and successfully applied to various engineered systems, not much work has been done to handle input and output hard constraints simultaneously when the dynamic models are unknown. This paper proposes a novel gradient estimator to minimise a cost function whose analytical expression is unknown, but measurable. By incorporating both input constraints and output constraints into the cost function, the proposed method can achieve a uniform semi-global practical asymptotic convergence of the tracking performance by tuning the parameters of a family of dithers carefully with constraints satisfactions. Simulation results are presented to support the theoretical findings.**

I. Introduction

Iterative learning control (ILC) learns to perform a task through repetitions by incorporating the error learned from the previous iterations. It was first proposed by Uchiyama [1] and soon attracted much attention after the pioneering works by Arimoto *et.al* [2]. Due to its model-free nature and simplicity in implementation, a large number of applications have been reported in the literature, such as precision motion control [3], [4], wind energy generation [5], robotic rehabilitation [6] [7], traffic flow control [8], high-speed trains [9], modelling of human motor control [7], see more applications in survey papers [10]–[12] and references therein.

In practical implementations of ILC algorithms, hard constraints in terms of input and output signals always exist. For example, both sensors (for the output) and actuators (for the input) have the upper and lower limit. When the system model is unknown, ensuring the convergence of the tracking error as well as satisfying both input and output constraints becomes challenging.

For continuous-time plants, the input constraints have been addressed in the design of ILC for a certain class of linear/nonlinear continuous-time systems using CEF based design [13]–[16]. Barrier functions like Lyapunov functions (BF-LFs) have been used in the stability analysis of nonlinear systems to deal with output constraints [17]. The BF-LF can be extended to the learning control design, see for example, [18]–[23] and references therein.

When the system of interest is in discrete-time, by using the super vector formulation, the ILC with constraints can be converted into a constrained optimization problem in a high dimensional space, see for example, [24]–[26] for input constraints and output constraints [27]–[30] by using typical optimization techniques provided that the analytical form of the cost is **known**, leading to a standard gradient based iterative algorithms. However, in most of ILC applications, such an analytical form for the cost is unknown. This work extends the existing work without assuming the knowledge of the engineered systems. Utilizing the input/output data, the gradient of the cost function can be estimated from the measured data instead of computing from the analytic form of the cost function.

In this work, the gradient of the unknown nonlinear mapping is estimated using a suitable family of discrete-time dither signals. This idea is motivated from the continuous-time version of dither-based gradient estimators [31]. The closeness of solution between the the proposed dither-based gradient estimator and the gradient is shown by extending the existing discrete-time averaging theory [32] to a new setting, in which the averaged system is semi-globally practically asymptotically (SPA) stable. This concludes that the proposed algorithm can converge uniformly semi-globally practically asymptotically to the optimal solutions.

II. Preliminaries and Problem Formulation

A. Notations and Preliminaries

Let $\mathcal{N}$ represent the set of all non-negative integers. The notation $\mathcal{R}$ represents the set of all real numbers. For any vector $\mathbf{x} \in \mathcal{R}^n$, $|\mathbf{x}|$ represents its Euclidean norm, defined as $|\mathbf{x}| \triangleq \sqrt{\mathbf{x}^\intercal \mathbf{x}}$, where $(\cdot)^\intercal$ represents the transpose. For a matrix $A \in \mathcal{R}^{n\times m}$, $|A|$ is the induced matrix norm. I_m represents a identity matrix for size $m \times m$.

For a mapping $\mathbf{f}(\cdot) : \mathcal{R} \rightarrow \mathcal{R}^m$, if there exist positive constants $k, \eta^* \in \mathcal{R}$ such that and for any $\eta \in [0, \eta^*]$ the inequality $|\mathbf{f}(\eta)| \leq k\eta$ holds. The function $\mathbf{f}(\eta)$ is the order of η and denoted by $\mathcal{O}(\eta)$ [33, Definition 10.1].

The set of n times continuously differentiable functions is represented by $\mathcal{C}^n$ with $n \in \mathcal{N}$. For a nonlinear mapping $f(\cdot) : \mathcal{R}^n \to \mathcal{R}$ satisfying $f(\cdot) \in \mathcal{C}^1$, its gradient is defined as

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x(1)} & \frac{\partial f(\mathbf{x})}{\partial x(2)} & \cdots & \frac{\partial f(\mathbf{x})}{\partial x(n)} \end{bmatrix}.$$

B. Problem formulation

A class of multiple-input-multiple-output (MIMO) discrete-time nonlinear non-affine systems is considered. At the i^{th} iteration, it has the following form

$$\begin{aligned} \mathbf{x}_i[k+1] &= \mathbf{f}(\mathbf{x}_i[k], \mathbf{u}_i[k]), \qquad \mathbf{x}_i[0] \in \mathcal{R}^n \\ \mathbf{y}_i[k] &= \mathbf{g}(\mathbf{x}_i[k], \mathbf{u}_i[k]), \end{aligned} \tag{1}$$

where $k \in [0,\ 1,\ 2,\ \cdots, N-1]$ represents time steps in discrete-time and $i \in \mathcal{N}$ represent the iteration (or trial) number. The vector $\mathbf{x} \in \mathcal{R}^n$ represents the state. The output is $\mathbf{y} \in \mathcal{R}^p$ and the input is $\mathbf{u} \in \mathcal{R}^m$. Here nonlinear **unknown** mappings $\mathbf{f}(\cdot,\cdot) : \mathcal{R}^n \times \mathcal{R}^m \to \mathcal{R}^n$, $\mathbf{g}(\cdot,\cdot) : \mathcal{R}^n \times \mathcal{R}^m \to \mathcal{R}^m$ are assumed to be locally Lipschitz continuous. Although the nonlinear mappings are unknown, the output signal $\mathbf{y}_i[k]$ is measurable. A super vector is introduced to describe the input and output sequences

$$\begin{aligned} \widetilde{\mathbf{u}}_i &= [\mathbf{u}_i[0]^\intercal, \mathbf{u}_i[1]^\intercal, \ldots, \mathbf{u}_i[N-1]^\intercal]^\intercal \in \mathcal{R}^{m\cdot N}, \\ \widetilde{\mathbf{y}}_i &= [\mathbf{y}_i[0]^\intercal, \mathbf{y}_i[1]^\intercal, \ldots, \mathbf{y}_i[N-1]^\intercal]^\intercal \in \mathcal{R}^{p\cdot N}, \end{aligned} \tag{2}$$

so that the nonlinear model (1) can be represented as the following unknown nonlinear static mapping $\boldsymbol{\phi} : \mathcal{R}^{m\cdot N} \to \mathcal{R}^{p\cdot N}$

$$\widetilde{\mathbf{y}}_i = \boldsymbol{\phi}\left(\widetilde{\mathbf{u}}_i, \mathbf{y}_i(0)\right). \tag{3}$$

For a given reference trajectory $\mathbf{y}_r[k], \forall k \in [0, \ldots, N-1]$, the ILC design tries to track this reference trajectory. For convenience of notation, we also define $\widetilde{\mathbf{y}}_r = [\mathbf{y}_r[0]^\intercal, \mathbf{y}_r[1]^\intercal, \cdots, \mathbf{y}_r[N-1]^\intercal]^\intercal$ and the error vector is defined accordingly.

$$\widetilde{\mathbf{e}}_i \triangleq \widetilde{\mathbf{y}}_r - \widetilde{\mathbf{y}}_i, \qquad i \in \mathcal{N}. \tag{4}$$

The input constraints and the output constraints are:

$$\begin{aligned} \boldsymbol{\psi}_1(\mathbf{y}[k]) &\geq \mathbf{0}_{n_1}, \qquad k = 0, \ldots, N-1 && (5) \\ \boldsymbol{\psi}_2(\mathbf{u}[k]) &\geq \mathbf{0}_{n_2}, \qquad k = 0, \ldots, N-1, && (6) \end{aligned}$$

where $\boldsymbol{\psi}_1(\cdot)$ and $\boldsymbol{\psi}_2(\cdot)$ are smooth nonlinear functions with appropriate dimensions. Here $n_1, n_2 \in \mathcal{N}$ and $\mathbf{0}_n$ is a vector in $\mathcal{R}^n$ in which each element of the vector is zero. For simplicity of analysis, the following identical initial condition [34] is assumed.

Assumption 1: The system (1) satisfies $\mathbf{y}_i[0] = \mathbf{y}_r[0]$, for all $i \in \mathcal{N}$, which is an identical initial condition [34]. Moreover, the system input and output at first iteration $\widetilde{\mathbf{u}}_0, \widetilde{\mathbf{y}}_0$ are assumed to satisfy the constraints defined in (5) and (6). □

Control Objective is to find a sequence of input signals $\{\widetilde{\mathbf{u}}_i\}_{i\in\mathcal{N}}$ such that $\lim_{i\to\infty} |\widetilde{\mathbf{e}}_i| = 0$ by satisfying both input constraints (5) and output constraints (6).

We can formulate the problem as the following optimization:

$$\begin{aligned} &\min_{\widetilde{\mathbf{u}} \in \mathcal{R}^{m\cdot N}} \widetilde{\mathbf{e}}^\intercal \widetilde{\mathbf{e}}, && (7) \\ &\textit{s.t.} \quad \begin{cases} \boldsymbol{\psi}_1(\mathbf{y}[k]) \geq \mathbf{0}_{n_1}, k = 0, \ldots, N-1, \\ \boldsymbol{\psi}_2(\mathbf{u}[k]) \geq \mathbf{0}_{n_2}, k = 0, \ldots, N-1 \end{cases}. \end{aligned}$$

Next assumption assumes that the optimization problem (7) is feasible.

Assumption 2: For the given $\widetilde{\mathbf{y}}_r$, which satisfies the output constraints (6), there exists a unique $\widetilde{\mathbf{u}}_r$, which is the solution of the optimization problem (7) and also satisfied input constraints (5). □

Based on the optimization formulation (7), the following cost function is constructed

$$\begin{aligned} J(\widetilde{\mathbf{u}}_i) = \widetilde{\mathbf{e}}_i^\intercal \widetilde{\mathbf{e}}_i \quad &- \sum_{k=0}^{N-1} \sum_{s=1}^{n_1} V^y\Big(\boldsymbol{\psi}_{1,s}(\mathbf{y}_i[k]), \boldsymbol{\theta}_y\Big) \\ &- \sum_{k=0}^{N-1} \sum_{s=1}^{n_2} V^u\Big(\boldsymbol{\psi}_{2,s}(\mathbf{u}_i[k]), \boldsymbol{\theta}_u\Big), \end{aligned} \tag{8}$$

where V^y, V^u are smooth barrier functions parameterized by some design parameters $\boldsymbol{\theta}_y$ and $\boldsymbol{\theta}_u$ respectively. These two barrier functions have the standard form:

$$V(\boldsymbol{\psi}(\mathbf{y}), \boldsymbol{\theta}_y) = \begin{cases} \mu_1(\mathbf{y}, \boldsymbol{\theta}_y) & \text{if } \ \mathbf{y} \in \mathcal{I}_1(\boldsymbol{\theta}_y) \\ \mu_2(\mathbf{y}, \boldsymbol{\theta}_y) & \text{if } \ \mathbf{y} \in \mathcal{I}_2(\boldsymbol{\theta}_y) \\ 0 & \text{if } \ \mathbf{y} \in \mathcal{I}_3(\boldsymbol{\theta}_y), \end{cases} \tag{9}$$

where the sets $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$ are parameterized by the parameter $\boldsymbol{\theta}_y$. The set $\mathcal{I}_3$ is a compact subset of the original set satisfying the constraints, while $\mathcal{I}_2$ is a transition set to ensure the needed smoothness requirement of the barrier function. When constraints are violated, $\mu_1(\cdot,\cdot)$ will approach to infinity. This type of barrier function will not introduce an addition bias in terms of the tracking error. More precisely, the optimal solution of (8) is $\widetilde{\mathbf{e}} = \mathbf{0}$. As the reference output $\mathbf{y}_r$ is known, it can be used to design an appropriate parameter $\boldsymbol{\theta}_y$ and three sets. However, as $\mathbf{u}_r$ is unknown, for simplicity of the presentation, the following assumption is used to ensure that the $\mathbf{u}_r$ is within the set $\mathcal{I}_3$.

Assumption 3: The unknown reference input $\mathbf{u}_r$ in Assumption 2 is compatible with the form of the proposed barrier function (9), i.e., there exists an appropriate parameter $\boldsymbol{\theta}_u$ such that $\mathbf{u}_r \in \mathcal{I}_3(\boldsymbol{\theta}_u)$ holds. □

In simulations, we will show an example of such barrier functions.

If the nonlinear mapping $\boldsymbol{\phi}$ in (3) is known, the analytic form of the cost function in (8) is known. In the sequel, the following gradient-descent algorithm is usually used, see, for example, [28] and references therein,

$$\widetilde{\mathbf{u}}_{i+1} = \widetilde{\mathbf{u}}_i - \varepsilon \nabla J(\widetilde{\mathbf{u}}_i), \tag{10}$$

where $0 < \varepsilon < 1$ represents the step size of the gradient descent method. The following assumption is used.

Assumption 4: For the cost defined in (8), which satisfies Assumptions 1–3, the gradient-descent algorithm proposed in (10) can achieve a global asymptotic convergence, uniformly in small ε. More precisely, there exists a class $\mathcal{KL}$ function $\beta(\cdot,\cdot)$ such that there exists $\varepsilon_0^* > 0$ such that for any $\varepsilon \in (0, \varepsilon_0^*)$, the solutions of (10) satisfy the following inequality:

$$|\widetilde{\mathbf{u}}_i - \widetilde{\mathbf{u}}_r| \leq \beta\left(|\widetilde{\mathbf{u}}_0 - \widetilde{\mathbf{u}}_r|, \varepsilon i\right) \tag{11}$$

for any $|\widetilde{\mathbf{u}}_0 - \widetilde{\mathbf{u}}_r| \in \mathcal{R}^{m\times N}$. Note that the convergence of the input $\widetilde{\mathbf{u}}$ indicate that the cost will always be bounded, it also means the constraints are satisfied. □

Remark 1: Gradient-descent methods are one of the most widely used optimization techniques in practice. Although it works locally, if the cost function $J(\cdot) : \mathcal{R}^{m\cdot N} \to \mathcal{R}_{\geq 0}$ is convex[1] and differentiable for any $\widetilde{\mathbf{u}} \in \mathcal{R}^{m\times N}$, we can obtain global convergence. A similar analysis can be easily extended to the cases when the updating law (10) converges locally asymptotically. ◦

Remark 2: The proposed method can be extended to other gradient-based optimization techniques, for example, Newton-method, which uses both the first order gradient and the second order gradient (Hessian matrix). The idea is similar to [35], which implements any gradient-based optimization algorithm, which relies on the known form of the cost, by using a class of gradient estimators to estimate the needed gradient. Our future work will exploit how to generate a similar framework in the area of ILC convergence. ◦

Although there exists an optimization algorithm (10) that can work well, it requires the explicit analytical form of the cost function. As the nonlinear mapping $\boldsymbol{\phi}(\cdot,\cdot)$ in (3) is unknown, the analytic form of the cost function is not available. Therefore, we can not compute the gradient $\nabla J(\widetilde{\mathbf{u}})$. In this work, the measured input and the cost will be used to estimate the gradient. Hence, we will design a data-driven gradient estimator to obtain $\widehat{\nabla J(\widetilde{\mathbf{u}})}$, which is an approximation of the needed gradient.

III. Main Results

As ILC algorithms are usually model-free technique without knowing the model information, this work assumes that the analytical form of system nonlinear mappings $\mathbf{f}(\cdot,\cdot), \mathbf{g}(\cdot,\cdot)$ is not available. We can only measure the cost $J(\widetilde{\mathbf{u}}_i)$ at each iteration. Thus a data-driven method is required to approximate the gradient needed in (10).

It is well known that the gradient of a static mapping can be estimated by some dither-based techniques, such as the extremum seeking method [36], Lie-bracket approximation [31] in continuous-time domain. The key idea is to inject a family of high frequency dither signals to probe and extract the gradient information. It is not easy to extend such method directly to the iteration domain. On the one hand, high frequency dither signals cannot be applied to the iteration domain. The fastest possible periodic signal (not constant) in iteration domain repeats itself in two iteration. In order to keep an appropriate time scale separation, we need to slow down the updating law to make the periodic signal "fast" compared with the slow updating law. This leads to the following form of updating law, which is served as the gradient estimator.

$$\widetilde{\mathbf{u}}^a_{i+1} = \widetilde{\mathbf{u}}^a_i + \sqrt{\varepsilon}\Gamma\Big(\boldsymbol{\zeta}_i + \boldsymbol{\xi}_i\Big), \quad (12)$$

$$\boldsymbol{\zeta}_i = \sin(J(\widetilde{\mathbf{u}}^a_i)) \cdot \mathbf{d}_i \quad (13)$$

$$\boldsymbol{\xi}_i = \cos(J(\widetilde{\mathbf{u}}^a_i)) \cdot \mathbf{q}_i, \quad (14)$$

where $\mathbf{d}_i, \mathbf{q}_i$ are dither signals and $\Gamma \in \mathcal{R}^{m\cdot N\times m\cdot N}$ is a diagonal matrix with all positive elements. The choice of Γ depends on the selected dither signals. Here ε comes from (10) satisfying Assumption 4.

[1]A differentiable function $J(\cdot) : \mathcal{R}^{m\cdot N} \to \mathcal{R}$ is said to be convex, if for any $\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2 \in \mathcal{R}^{m\cdot N}$ the following inequality holds: $J(\widetilde{\mathbf{x}}_1) \geq J(\widetilde{\mathbf{x}}_2) + \nabla_{\mathbf{x}} J(\widetilde{\mathbf{x}}_2)(\widetilde{\mathbf{x}}_1 - \widetilde{\mathbf{x}}_2)$

The role of the parameter ε is two-fold. On one hand, it is the step size of gradient descent method (10). On the other hand, it serves as a time scale separation parameter so that the updating law is slower than the periodicity of dither signals so that the standard discrete-time averaging technique [32] is applicable. Our main result will show the closeness of solutions between the gradient descent system (10) and its estimation (12).

Dither signals $\mathbf{d}_i, \mathbf{q}_i \in \mathcal{R}^{m\cdot N}$ play an important role in the proposed gradient estimation algorithm (12). In general, they need to be "rich" enough or persistently excited (PE) to estimate a high dimensional parameter as pointed out in [37]. In order to guarantee the needed PE condition, the dithers need to satisfy the following assumption:

Assumption 5: Let $d_{i,l}(d_{j,l})$, $q_{i,l}(q_{j,l})$ be the $l^{th}(j^{th})$ element of $\mathbf{d}_i$, $\mathbf{q}_i$ respectively.

(1) $d_{i,l}, q_{i,l}$ are bounded periodical signals with the same period $K_l \in \mathcal{N}$. That is, $d_{i+K_l,l} = d_{i,l}$ and $q_{i+k_l,l} = q_{i,l}$ for all $i \in \mathcal{N}$.
(2) Both $d_{i,l}$ and $q_{i,l}$ have zero mean, i.e., $\sum_{i=0}^{K_l-1} d_{i,l} = 0$ and $\sum_{i=0}^{K_l-1} q_{i,l} = 0$.
(3) Let K be the least common multiple of $K_l, l = 1,\ldots,m\cdot N$. There exist positive constants $c_{1,l}$ and $c_{2,l}$ such that the following conditions hold

$$\sum_{i=0}^{K-1}\left(q_{i,j}\sum_{s=0}^{i} q_{s,l}\right) = \begin{cases} c_{1,l}, & \text{if } j = l \\ 0, & \text{otherwise} \end{cases}, \quad (15)$$

$$\sum_{i=0}^{K-1}\left(q_{i,j}\sum_{s=0}^{i} d_{s,l}\right) = \begin{cases} c_{2,l}, & \text{if } j = l \\ 0, & \text{otherwise} \end{cases}, \quad (16)$$

$$\sum_{i=0}^{K-1}\left(q_{i,j}\sum_{s=0}^{k} d_{s,l}\right) = -\sum_{i=0}^{K-1}\left(d_{i,j}\sum_{s=0}^{i} q_{s,l}\right), \quad (17)$$

$$\sum_{i=0}^{K-1}\left(d_{i,j}\sum_{s=0}^{i} d_{s,l}\right) = \sum_{i=0}^{K-1}\left(q_{i,j}\sum_{s=0}^{i} q_{s,l}\right). \quad (18)$$

For the convenience of analysis, the diagonal matrices $C_1, C_2 \in \mathcal{R}^{m\cdot N\times m\cdot N}$ are defined with elements $c_{1,l}$ and $c_{2,l}$.

□

Remark 3: As the dimension of the input of the optimization problem (8) is very high, i.e., $\widetilde{\mathbf{u}} \in \mathcal{R}^{m\times N}$, the choice of dithers becomes quite challenging as the number of frequencies used is proportional to the number of parameters to estimate. At the same time, when more dither frequencies are used, the parameter ε needs to be sufficiently small to keep the time scale separation, leading to a slower convergence speed. In our future work, we will explore optimization techniques that are less sensitive to the high dimensional input. ◦

A. Closeness of solutions

Next will show that if Assumption 5 holds, the proposed updating law $\widetilde{\mathbf{u}}^a_i$ well approximates the gradient descent ILC $\widetilde{\mathbf{u}}_i$ by selecting appropriate ε.

Theorem 1: Considering the gradient descent algorithm (10) and its approximation (12)–(14). If Assumptions 1–5 hold, for any positive integer $K_1 \in \mathcal{N}$ and K, C_2 coming from Assumption 5, there exists a constant $0 < \varepsilon_1^* < \varepsilon_0^*$, where ε_0

comes from Assumption 4, such that for any $\varepsilon \in (0, \varepsilon_1^*]$, the following inequality is satisfied with $\Gamma = KC_2^{-1}$

$$|\widetilde{\mathbf{u}}_i^a - \widetilde{\mathbf{u}}_i| = \mathcal{O}(\varepsilon), \tag{19}$$

for all $i \in [0, 1, \dots, \lfloor \frac{K_1}{\varepsilon K} \rfloor K]$ and $\widetilde{\mathbf{u}}_0^a = \widetilde{\mathbf{u}}_0$. ■

Proof: We first analyze the trajectories of (12) in a finite time interval $[0, K]$, where K comes from Assumption 5. It is noted that Assumption 4 indicates that for the finite time interval $[0, K]$, the trajectories of (10) is bounded. The trajectories of the updating law (12) at the K^{th} step are:

$$\widetilde{\mathbf{u}}_K^a = \widetilde{\mathbf{u}}_0 + \sqrt{\varepsilon\Gamma} \sum_{s=0}^{K-1} \Big(\boldsymbol{\zeta}_s + \boldsymbol{\xi}_s \Big). \tag{20}$$

Let denote $J(\widetilde{\mathbf{u}}_i^a)$ as J_i. By using summation by parts, the second term in (20) becomes:

$$\begin{aligned}
\sum_{s=0}^{K-1} \boldsymbol{\xi}_s &= \cos(J_{K-1}) \sum_{s=0}^{K-1} \mathbf{q}_s \\
&\quad - \sum_{s=0}^{K-2} \left(\Big(\cos(J_{s+1}) - \cos(J_s) \Big) \sum_{j=0}^{s} \mathbf{q}_j \right) \\
&= \cos(J_{K-1}) \sum_{s=0}^{K-1} \mathbf{q}_s \\
&\quad - \sum_{s=0}^{K-1} \left(\Big(\cos(J_{s+1}) - \cos(J_s) \Big) \sum_{j=0}^{s} \mathbf{q}_j \right) \\
&\quad + \Big(\cos(J_K) - \cos(J_{K-1}) \Big) \sum_{j=0}^{K-1} \mathbf{q}_j \\
&= \cos(J_K) \sum_{s=0}^{K-1} \mathbf{q}_s \\
&\quad - \sum_{s=0}^{K-1} \left(\Big(\cos(J_{s+1}) - \cos(J_s) \Big) \sum_{j=0}^{s} \mathbf{q}_j \right).
\end{aligned} \tag{21}$$

Due to the second item in Assumption 5, the first term in (21) is zero. By using Taylor's expansion in [38], the second term in (21) can be written as:

$$J_{s+1} - J_s = \nabla J_s^\intercal \sqrt{\varepsilon\Gamma} \Big(\boldsymbol{\zeta}_s + \boldsymbol{\xi}_s \Big) + \mathcal{O}_1(\varepsilon), \tag{22}$$

where $\mathcal{O}_1(\varepsilon)$ is the residue. Using Taylor's expansion again, it leads to

$$\begin{aligned}
&\cos(J_{s+1}) - \cos(J_s) \\
=\ & -\sqrt{\varepsilon\Gamma} \sin(J_s) \nabla J_s^\intercal \Big(\sin(J_s)\mathbf{d}_s + \cos(J_s)\mathbf{q}_s \Big) \\
+\ & \mathcal{O}_2(\varepsilon),
\end{aligned} \tag{23}$$

Substituting(24) back to equation (21), we have:

$$\begin{aligned}
\sum_{s=0}^{K-1} \boldsymbol{\xi}_s &= \sqrt{\varepsilon\Gamma} \sum_{s=0}^{K-1} \left(\Big(\sin(J_s)^2 \nabla J_s^\intercal \mathbf{d}_s \Big) \sum_{j=0}^{s} \mathbf{q}_j \right) \\
&+ \sqrt{\varepsilon\Gamma} \sum_{s=0}^{K-1} \left(\Big(\sin(J_s)\cos(J_s) \nabla J_s^\intercal \mathbf{q}_s \Big) \sum_{j=0}^{s} \mathbf{q}_j \right) \\
&+ \mathcal{O}_3(\varepsilon).
\end{aligned} \tag{24}$$

Similarly, we can compute $\sum_{s=0}^{K-1} \boldsymbol{\xi}_s$, resulting in the following bound of the solutions (12) at $i = K$

$$\begin{aligned}
&\widetilde{\mathbf{u}}_K^a = \widetilde{\mathbf{u}}_0^a \\
&+ \varepsilon\Gamma \sum_{s=0}^{K-1} \left(\Big(-\sin(J_s)\cos(J_s) \nabla J_s^\intercal \mathbf{d}_s \Big) \sum_{j=0}^{s} \mathbf{d}_j \right) \\
&+ \varepsilon\Gamma \sum_{s=0}^{K-1} \left(\Big(-\cos(J_s)^2 \nabla J_s^\intercal \mathbf{q}_s \Big) \sum_{j=0}^{s} \mathbf{d}_j \right) \\
&+ \varepsilon\Gamma \sum_{s=0}^{K-1} \left(\Big(\sin(J_s)^2 \nabla J_s^\intercal \mathbf{d}_s \Big) \sum_{j=0}^{s} \mathbf{q}_j \right) \\
&+ \varepsilon\Gamma \sum_{s=0}^{K-1} \left(\Big(\sin(J_s)\cos(J_s) \nabla J_s^\intercal \mathbf{q}_s \Big) \sum_{j=0}^{s} \mathbf{q}_j \right) \\
&+ \mathcal{O}_4(\varepsilon^{\frac{3}{2}}).
\end{aligned} \tag{25}$$

For convenience, we denote $\chi_{1,s} = -\sin(J_s)\cos(J_s), \eta_{1,s} = \sum_{j=0}^{s} \mathbf{d}_j \mathbf{d}_s^\intercal$, $\chi_{2,s} = -\cos(J_s)^2, \eta_{2,s} = \sum_{j=0}^{s} \mathbf{d}_j \mathbf{q}_s^\intercal$, $\chi_{3,s} = \sin(J_s)^2, \eta_{3,s} = \sum_{j=0}^{s} \mathbf{q}_j \mathbf{d}_s^\intercal$, and $\chi_{4,s} = \sin(J_s)\cos(J_s), \eta_{4,s} = \sum_{j=0}^{s} \mathbf{q}_j \mathbf{q}_s^\intercal$. By repeating the above procedure, at any $(n+1)K, n \in \mathcal{N}$, with the initial condition $\widetilde{\mathbf{u}}_0$, the solutions of updating law (10) can be re-written as

$$\widetilde{\mathbf{u}}_{(n+1)K}^a = \widetilde{\mathbf{u}}_0^a + \varepsilon\Gamma \sum_{i=0}^{(n+1)K-1} \mathbf{h}(i, \widetilde{\mathbf{u}}_i, \varepsilon), \tag{26}$$

where

$$\mathbf{h}(i, \widetilde{\mathbf{u}}_i, \varepsilon) = \Big(\sum_{s=1}^{4} \chi_{s,i}\eta_{s,i} + \mathcal{O}_5(\varepsilon^{\frac{1}{2}}) \Big) \nabla J_i. \tag{27}$$

As pointed out in the [32], for a given positive integer K_1 the trajectory of $\widetilde{\mathbf{u}}_i^a$ is $\mathcal{O}(\varepsilon)$ close to its averaged system trajectory $\widetilde{\mathbf{u}}_i^{av}$, for any $i \in \lfloor \frac{K_1}{\varepsilon} \rfloor$. The averaged system is defined as:

$$\widetilde{\mathbf{u}}_{i+1}^{av} = \widetilde{\mathbf{u}}_0^{av} + \varepsilon\Gamma \mathbf{h}_{av}(\widetilde{\mathbf{u}}_i^{av}), \tag{28}$$

$$\mathbf{h}_{av}(\widetilde{\mathbf{u}}) = \frac{1}{K} \sum_{i=0}^{K-1} \mathbf{h}(i, \widetilde{\mathbf{u}}, 0). \tag{29}$$

By using the properties from Assumption 5 term 3, it has

$$\begin{aligned}\mathbf{h}_{av}(\widetilde{\mathbf{u}}) = & -\frac{C_1}{K}\sin(J(\widetilde{\mathbf{u}}))\cos(J(\widetilde{\mathbf{u}}))\nabla J(\widetilde{\mathbf{u}}) \\ & -\frac{C_2}{K}\cos(J(\widetilde{\mathbf{u}}))^2\nabla J(\widetilde{\mathbf{u}}) \\ & -\frac{C_2}{K}\sin(J(\widetilde{\mathbf{u}}))^2\nabla J(\widetilde{\mathbf{u}}) \\ & +\frac{C_1}{K}\sin(J(\widetilde{\mathbf{u}}))\cos(J(\widetilde{\mathbf{u}}))\nabla J(\widetilde{\mathbf{u}}) \\ = & -\frac{C_2}{K}\nabla J(\widetilde{\mathbf{u}}),\end{aligned} \quad (30)$$

where C_1, C_2 come from Assumption (5). Note that the averaged system is equivalent to the gradient descent ILC algorithm (10) with $\Gamma = \dfrac{K}{C_2}$. In the sequel, we can show the closeness between the trajectories of (10) and (12) , i.e.,

$$|\widetilde{\mathbf{u}}_i^a - \widetilde{\mathbf{u}}_i| = \mathcal{O}(\varepsilon), \quad (31)$$

for all $i \in [0, 1, \ldots, \lfloor \frac{K_1}{\varepsilon K} \rfloor K]$. The completes the proof. ■

With the closeness result in Theorem 1, if (10) is UGA, uniformly in small ε, next result can conclude the convergence property of the system (12). The proof follows the standard trajectory-based analysis as shown in [39]. Due to space limitation, the proof is omitted.

Theorem 2: Considering the gradient descent algorithm (10) and its approximation (12)–(14). If Assumptions 1– 5 hold, for any positive pair (Δ, δ), there exists positive ε_2^*, such that for any $\varepsilon \in (0, \varepsilon]$, the solution of (12) satisfies

$$|\widetilde{\mathbf{u}}_i^a - \widetilde{\mathbf{u}}_r| \leq \max\{\beta_1(|\widetilde{\mathbf{u}}_0 - \widetilde{\mathbf{u}}_r|, \varepsilon i), \delta\}, \quad (32)$$

for all $i \in \mathcal{N}$ and $|\widetilde{\mathbf{u}}_0 - \widetilde{\mathbf{u}}_r| \leq \Delta$. Here the class $\mathcal{KL}$ function comes from Assumption 4. ■

IV. Simulation Results

Consider a nonlinear single-input-single-output discrete system:

$$\begin{aligned} x[k+1] &= \sin(x[k]) + u[k] \\ y[k] &= x[k] + u[k], \end{aligned} \quad (33)$$

with a reference input $u_r[k] = \sin(2k), \forall k \in [0, 1, \ldots, 6]$, initial condition $x[0] = 0.8$. The corresponding reference output y_r is generated via the system (33). A specific choice of the dithers which satisfy the Assumption 5 is selected as

$$d_{i,l} = \begin{cases} 1, \text{ if } & \bmod(i, K_l) \in \left[0, K_l' - 1\right] \\ 0, \text{ if } & \bmod(i, K_l) \in \left[K_l', 2K_l' - 1\right] \\ -1, \text{ if } & \bmod(i, K_l) \in \left[2K_l', 3K_l' - 1\right] \\ 0, \text{ if } & \bmod(i, K_l) \in \left[3K_l', 4K_l' - 1\right] \end{cases}, \quad (34)$$

$$q_{i,l} = \begin{cases} 0, \text{ if } & \bmod(i, K_l) \in \left[0, K_l' - 1\right] \\ 1, \text{ if } & \bmod(i, K_l) \in \left[K_l', 2K_l' - 1\right] \\ 0, \text{ if } & \bmod(i, K_l) \in \left[2K_l', 3K_l' - 1\right] \\ -1, \text{ if } & \bmod(i, K_l) \in \left[3K_l', 4K_l' - 1\right] \end{cases},$$

where $K_l = 4K_l' = 2^{l+1}, \forall l \in [1, 2, \ldots, 7]$ and $K = 2^8$, it can be checked that the term 3 in Assumption 5 is satisfied with $c_{1,l} = \frac{K_l}{4}, c_{2,l} = \frac{KK_l}{16}$.

In order to illustrate the role of ε in Theorem 1, different choices of ε are selected: $\varepsilon = 0.0002$ and $\varepsilon = 0.00005$. In this case, an unconstrained tracking problem is considered, such that

$$J(\widetilde{\mathbf{u}}) = \widetilde{\mathbf{e}}^{\intercal}\widetilde{\mathbf{e}} \quad (35)$$

It can be seen that the cost J converge to its minimum as shown in Fig.1. Moreover, by selecting a smaller ε, the convergence speed of the cost J is slowing down while the ultimate bound is getting smaller. This clearly shows the performance trade-off in terms of convergence speed and the accuracy or the ultimate bound.

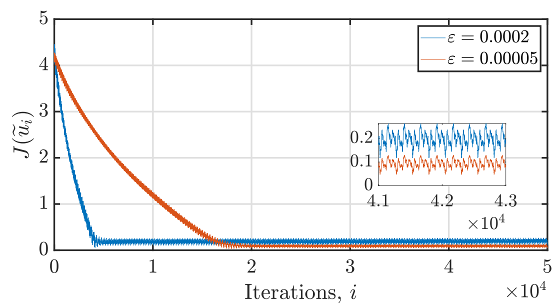

Fig. 1. Unconstrained cost J_i with $\varepsilon = 0.0002, 0.00005$

An upper bound for input $u_b = 1.05$ and a lower bound of output $y_b = 0.3$ are considered, such that:

$$\begin{aligned} u_b - u[k] &\geq 0, \quad \forall k \in [0, 1, \ldots, 6] \\ y[k] - y_b &\geq 0, \quad \forall k \in [0, 1, \ldots, 6]. \end{aligned} \quad (36)$$

Considering the limitation of the space, the figures are omitted here, but it can be checked that both the input and output constrains are violated at some trails when the parameter ε is selected as $\varepsilon = 0.00005$.

When constraints are considered, the parameter is selected as $\varepsilon = 0.00005$. It is assumed that the boundedness of the reference input $b_1 = 1$. The simulation results demonstrate the effectiveness of the proposed algorithm.

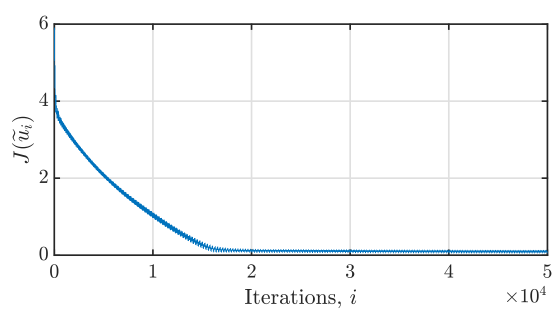

Fig. 2. Cost $J(\widetilde{u}_i)$ with constraints considered

V. Conclusion and Future Work

This work provides a framework for ILC to handle input and output hard constrains by formulating the tracking task into a constrained optimization problem. A data-driven gradient estimation algorithm, which utilizes the dither signals to estimate the gradient, is proposed find the optimal solution. The main result shows that the proposed algorithm can semi-globally practically asymptotically converge to the optimal solution, under appropriate conditions. Our future work will focus on how to reduce the complexity coming from data-driven gradient estimation.

References

[1] M. Uchiyama, "Formation of high-speed motion pattern of a mechanical arm by trial," *Transactions of the Society of Instrument and Control Engineers*, vol. 14, no. 6, pp. 706–712, 1978.

[2] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.

[3] K. L. Barton and A. G. Alleyne, "A cross-coupled iterative learning control design for precision motion control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1218–1231, 2008.

[4] T. Oomen, "Advanced motion control for precision mechatronics: Control, identification, and learning of complex systems," *IEEJ Journal of Industry Applications*, vol. 7, no. 2, pp. 127–140, 2018.

[5] O. Tutty, M. Blackwell, E. Rogers, and R. Sandberg, "Iterative learning control for improved aerodynamic load performance of wind turbines with smart rotors," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 967–979, 2013.

[6] C. T. Freeman, E. Rogers, A.-M. Hughes, J. H. Burridge, and K. L. Meadmore, "Iterative learning control in health care: Electrical stimulation and robotic-assisted upper-limb stroke rehabilitation," *IEEE Control Systems Magazine*, vol. 32, no. 1, pp. 18–43, 2012.

[7] S.-H. Zhou, J. Fong, V. Crocher, Y. Tan, D. Oetomo, and I. Mareels, "Learning control in robot-assisted rehabilitation of motor skills–a review," *Journal of Control and Decision*, vol. 3, no. 1, pp. 19–43, 2016.

[8] Z. Hou, J.-X. Xu, and J. Yan, "An iterative learning approach for density control of freeway traffic flow via ramp metering," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 1, pp. 71–97, 2008.

[9] D. Huang, W. Yang, T. Huang, N. Qin, Y. Chen, and Y. Tan, "Iterative learning operation control of high-speed trains with adhesion dynamics," *IEEE Transactions on Control Systems Technology*, 2021.

[10] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: brief survey and categorization," *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, vol. 37, no. 6, p. 1099, 2007.

[11] D. A. Bristow and J. R. Singler, "Analysis of transient growth in iterative learning control using pseudospectra," in *Proceedings of the Symposium on Learning Control at IEEE CDC*, 2009.

[12] D. Shen, "Iterative learning control with incomplete information: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 5, pp. 885–901, 2018.

[13] J.-X. Xu, Y. Tan, and T.-H. Lee, "Iterative learning control design based on composite energy function with input saturation," *Automatica*, vol. 40, no. 8, pp. 1371–1377, aug 2004.

[14] M. Sun, D. Wang, and P. Chen, "Repetitive learning control of nonlinear systems over finite intervals," *Science in China Series F: Information Sciences*, vol. 53, no. 1, pp. 115–128, 2010.

[15] Y. Tan, H.-H. Dai, D. Huang, and J.-X. Xu, "Unified iterative learning control schemes for nonlinear dynamic systems with nonlinear input uncertainties," *Automatica*, vol. 48, no. 12, pp. 3173–3182, 2012.

[16] G. Sebastian, Y. Tan, D. Oetomo, and I. Mareels, "On input and output constraints in iterative learning control design for robotic manipulators," *Unmanned Systems*, vol. 06, no. 03, pp. 197–208, 2018.

[17] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier Lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp. 918–927, 2009.

[18] J.-X. Xu and X. Jin, "State-constrained iterative learning control for a class of MIMO systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 5, pp. 1322–1327, 2013.

[19] X. Jin and J.-X. Xu, "Iterative learning control for output-constrained systems with both parametric and nonparametric uncertainties," *Automatica*, vol. 49, no. 8, pp. 2508–2516, 2013.

[20] X. Jin and J.-X. Xu, "A barrier composite energy function approach for robot manipulators under alignment condition with position constraints," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 17, pp. 2840–2851, 2014.

[21] K. Yovchev, K. Delchev, and E. Krastev, "State space constrained iterative learning control for robotic manipulators," *Asian Journal of Control*, vol. 20, no. 3, pp. 1145–1150, 2018.

[22] G. Sebastian, Y. Tan, D. Oetomo, and I. Mareels, "Feedback-based iterative learning design and synthesis with output constraints for robotic manipulators," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 513–518, July 2018.

[23] X. Jin, "Fault-tolerant iterative learning control for mobile robots non-repetitive trajectory tracking with output constraints," *Automatica*, vol. 94, no. 7, pp. 63–71, 2018.

[24] B. Chu and D. H. Owens, "Iterative learning control for constrained linear systems," *International Journal of Control*, vol. 83, no. 7, pp. 1397–1413, 2010.

[25] P. Janssens, G. Pipeleers, and J. Swevers, "A data-driven constrained norm-optimal iterative learning control framework for LTI systems," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 546–551, 2013.

[26] M.-B. Radac, R.-E. Precup, and E. M. Petriu, "Constrained data-driven model-free ILC-based reference input tuning algorithm," *Acta Polytechnica Hungarica*, vol. 12, no. 1, pp. 137–160, 2015.

[27] A. P. Schoellig, F. L. Mueller, and R. D'Andrea, "Optimization-based iterative learning for precise quadrocopter trajectory tracking," *Autonomous Robots*, vol. 33, no. 1-2, pp. 103–127, 2012.

[28] C. T. Freeman and Y. Tan, "Iterative learning control with mixed constraints for point-to-point tracking," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 604–616, 2013.

[29] M. Volckaert, M. Diehl, and J. Swevers, "Generalization of norm optimal ILC for nonlinear systems with constraints," *Mechanical Systems and Signal Processing*, vol. 39, no. 1-2, pp. 280–296, 2013.

[30] D. H. Owens, *Iterative learning control: an optimization paradigm*. Springer, 2015.

[31] H.-B. Dürr, M. S. Stanković, C. Ebenbauer, and K. H. Johansson, "Lie bracket approximation of extremum seeking systems," *Automatica*, vol. 49, no. 6, pp. 1538–1552, 2013.

[32] E.-W. Bai, L.-C. Fu, and S. S. Sastry, "Averaging analysis for discrete time and sampled data adaptive systems," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 2, pp. 137–148, 1988.

[33] H. K. Khalil, *Nonlinear systems*, 2nd ed. Prentice Hall, Upper Saddle River, NJ 07458, 1996.

[34] J.-X. Xu and Y. Tan, *Linear and nonlinear iterative learning control*. Springer, 2003, vol. 291.

[35] D. Nesić, Y. Tan, W. H. Moase, and C. Manzie, "A unifying approach to extremum seeking: Adaptive schemes based on estimation of derivatives," in *Proceedings of IEEE conference on decision and control*, 2010, pp. 4625–4630.

[36] S. Z. Khong, D. Nešić, and M. Krstić, "Iterative learning control based on extremum seeking," *Automatica*, vol. 66, pp. 238–245, 2016.

[37] M. Guay, D. Dochain, and M. Perrier, "Adaptive extremum seeking control of continuous stirred tank bioreactors with unknown growth kinetics," *Automatica*, vol. 40, no. 5, pp. 881–888, 2004.

[38] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[39] D. S. Laila, D. Nešić, and A. Astolfi, "3 sampled-data control of nonlinear systems," in *Advanced topics in control systems theory*. Springer, 2006, pp. 91–137.