2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Data-driven Iterative Learning Control for Continuous-Time Systems

Bing Chu, Paolo Rapisarda

Abstract— **We develop a data-driven iterative learning control design framework for continuous-time systems that does not require explicit or implicit identification of a system model. Using Chebyshev polynomial orthogonal bases, we show that all system trajectories can be characterised from sufficiently rich input/output data. Using this crucial result we develop a data-driven version of the model-based norm-optimal iterative learning control algorithm, and provide a computationally efficient implementation thereof. We rigorously analyse the convergence properties of the resulting design and also present a numerical example to illustrate its effectiveness.**

I. Introduction

Data-driven techniques are currently one of the most active research areas in control (see e.g. [1]–[5]). Much progress has recently been made also on data-driven Iterative Learning Control (ILC) techniques (see e.g. [6]–[10]). However, most attention in this area has been focused on discrete-time problems; research about data-driven ILC design for continuous-time systems has been relatively scarce. Traditional PID type ILC design (see [11], [12]) uses only previous trial's data (without model information) to update the next trial input. However, it requires a proper tuning of the PID parameters, and generally has limited performance. Adaptive ILC design (see e.g. [13]) requires knowledge of the model structure and an explicit or implicit parameter identification process. Moreover, its convergence performance (e.g., monotonic error convergence that is desirable in practice) is not guaranteed. Frequency-domain model-free ILC design (see e.g. [14], [15]) involves an explicit process to identify the frequency response of the system (or its inverse) from past data.

In this paper, we develop a data-driven ILC design framework for continuous-time systems that does not require *any* (explicit or implicit) model or parameter identification process, and that has *guaranteed convergence* performance (monotonic tracking error norm convergence). To this purpose we use approximation theory concepts (Chebyshev polynomial orthogonal bases; COPB in the following) and reference-trajectory tracking as an exemplary ILC problem.

Some features of COPB important for data-driven ILC are:

- Numerically accurate and computationally efficient algorithms are available to compute COPB representations from *samples* of continuous-time functions.
- The convergence rate of the series is faster the smoother the function is. Approximation error bounds are available for truncated series.
- Differentiation is reformulated as multiplication of the vector of COPB coefficients and a *differentiation matrix*.
- Error bounds for the approximation error of the derivative computed using a truncated series are computable directly from the function itself.

Crucial in our approach is the transformation of the continuous-time input and output trajectories into *discrete* sequences of the coefficients of their COPB representations. The COPB representation of the derivatives of these trajectories can be computed *directly* via linear algebra operations from the COPB representation of the trajectories themselves, and no direct measurement of the derivatives or their numerical approximation based on discretization is needed.

We exploit these advantages to reformulate the classical model-based continuous-time reference-tracking ILC problem (see [16], [17]) into *arbitrarily accurate*, *data-driven*, *finite-dimensional* approximations thereof, amenable to computationally efficient and numerically accurate computer treatment. We demonstrate the efficacy of COPB in data-driven continuous-time ILC with numerical simulations.

The paper is structured as follows. In section II we recall some basic definitions and concepts pertaining to iterative learning control (subsection II-A), and to COPB (subsection II-B). Section III contains the statement of a fundamental result relating COPB representations of "sufficiently rich" input-output data produced by a continuous-time LTI system, and the COPB representations of *all* system trajectories. We exploit such fundamental result in section IV, where we reformulate the classical ILC problem into a data-driven problem defined on a space of coefficient sequences. We show how to use standard quadratic optimization techniques to compute a solution and we rigorously analyse the convergence properties of the resulting design. We illustrate our approach with a numerical example in section V, and we discuss our contribution and current research in section VI.

Notation

We denote by $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{C}$ respectively the set of natural, real and complex numbers. $\mathbb{R}^n$, respectively $\mathbb{C}^n$, denote the space of n-dimensional vectors with real, respectively complex, entries. $\mathbb{R}^{n\times m}$ denotes the set of $n \times m$ matrices with real entries; $\mathbb{R}^{n\times\infty}$ the set of real matrices with n rows and an infinite number of columns; and $\mathbb{R}^{\infty\times\infty}$ the set of real matrices with an infinite number of rows and columns. The transpose of a matrix M is denoted by $M^\top$, and its pseudoinverse by $M^\dagger$. If A and B are two matrices with the same number of columns, we define $\mathrm{col}(A,B) := \begin{bmatrix} A^\top & B^\top \end{bmatrix}^\top$. The i-th row of a matrix M is denoted by $M_{i,:}$, and its i-th column denoted by $M_{:,i}$. For matrix $A \in \mathbb{R}^{n\times m}$, the column vector $vec(A)$ is defined by stacking all the columns of A, i.e., $vec(A) := \begin{bmatrix} A_{:,1}^\top & A_{:,2}^\top & \cdots & A_{:,n}^\top \end{bmatrix}^\top$. For a

School of Electronics and Computer Science, University of Southampton, UK, B.Chu@ecs.soton.ac.uk, pr3@ecs.soton.ac.uk

linear operator L, its kernel and range are denoted by $\ker[L]$ and $\mathcal{R}[L]$ respectively. We denote by $\mathcal{L}_2(\mathbb{I},\mathbb{R})$ the space of square-integrable real-valued functions defined on a finite interval $\mathbb{I} := [t_0, t_1] \subset \mathbb{R}$.

II. Background

A. Iterative Learning Control

Iterative learning control (ILC) is a control design method for systems working in a repetitive manner, e.g. a robotic arm on a car assembly line which performs the same task repeatedly, with high tracking accuracy requirements. The idea of ILC is that by learning from previous executions of the same task, improved tracking performance can be obtained. We formalize the ILC design problem as follows.

Consider the following n^{th} order linear time-invariant continuous-time system with m inputs and p outputs

$$\begin{aligned}\frac{d}{dt}x_k(t) &= Ax_k(t) + Bu_k(t)\\ y_k(t) &= Cx_k(t) + Du(t),\ x(t_0) = x_0\ ,\end{aligned} \tag{1}$$

where $x_k(t) \in \mathbb{R}^n, u_k(t) \in \mathbb{R}^m$ and $y_k(t) \in \mathbb{R}^p$ denote the state, input and output on k^{th} iteration (or trial) at time t; A, B, C, D are systems matrices with proper dimensions. The system output y_k is required to track a given reference signal r defined over a finite interval $\mathbb{I} := [t_0, t_1]$ and in a repetitive manner, i.e., at time $t = t_1$, the time index is reset to $t = t_0$, the state is reset to $x(t_0) = x_0$, and the system is required to track r over $\mathbb{I}$ again.

In operator form, the system (1) is represented as

$$y_k = S_p u_k + d\ ,$$

where $y_k \in \mathcal{L}_2(\mathbb{I},\mathbb{R}^p), u_k \in \mathcal{L}_2(\mathbb{I},\mathbb{R}^m), S_p$ is the convolution operator, and $d \in \mathcal{L}_2(\mathbb{I},\mathbb{R}^p)$ represents the effect of initial condition as given below

$$y_k(t) = \int_{t_0}^{t} Ce^{A(t-\tau)}Bu_k(\tau)d\tau + Du(t) + Ce^{At}x_0.$$

For simplicity we can assume that $d = 0$ by incorporating it into the reference – see [16], [17] for more details.

The ILC design task is to find a control-updating law as

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

where $e_k := r - y_k \in \mathcal{L}_2(\mathbb{I},\mathbb{R}^p)$ is the tracking error on trial k. The objective is to design the updating law so that a steadily improved tracking performance and ideally perfect tracking is achieved, i.e.,

$$\lim_{k\to\infty} e_k = 0\ .$$

A number of design methods have been proposed in the literature. Among them, optimisation-based design ones have attracted significant interests due to their guaranteed convergence performance. This is the focus of this paper. In particular, we consider the following so called *norm optimal ILC design* (NOILC), which in its simplest form at each trial, optimizes the following performance index

$$\begin{aligned} u_{k+1} \quad &= \quad \text{argmin}_{u_{k+1}}\{\|e_{k+1}\|_Q^2 + \|u_{k+1} - u_k\|_R^2\}\\ \text{s.t.} \quad & \quad y_{k+1} = S_p u_{k+1} \end{aligned} \tag{2}$$

where $e_{k+1} = r - y_{k+1}$ is the tracking error on trial $k+1$, and the norms are induced by the following inner products in the output space $y, f \in \mathcal{L}_2(\mathbb{I},\mathbb{R}^p)$:

$$\langle y, f\rangle_Q := \int_{t_0}^{t_1} y^\top(t)Qf(t)dt\ ;$$

and in the input space $u, v \in \mathcal{L}_2(\mathbb{I},\mathbb{R}^m)$:

$$\langle u, v\rangle_R := \int_{t_0}^{t_1} u^\top(t)Rv(t)dt$$

where $Q \succ 0$ and $R \succ 0$ have compatible dimensions.

The above NOILC algorithm has several appealing convergence properties, including guaranteed monotonic convergence of the tracking error norm, and (under mild conditions) perfect tracking of the reference signal. However, its formulation does require full model information (the convolution operator S_p) in the NOILC optimisation problem (2). In this paper, we develop a data-driven NOILC design such that *no model information* is needed, using latest developments from COPB based data-driven control for continuous-time systems. Before that, we first present some basic material on Chebyshev polynomial orthogonal bases.

B. Chebyshev polynomial orthogonal bases

In the rest of this paper we denote by $\mathbb{I}$ the interval $[-1, 1]$. This does not entail any loss of generality, as every interval $[t_0, t_1]$ can be transformed into $\mathbb{I}$ by linear translation and scaling. Define the *weight function* w by

$$w(t) := \frac{1}{\sqrt{1-t^2}}\ , t \in \mathbb{I}\ .$$

The *Chebyshev polynomials* are defined by $C_0(t) := 1$, $C_1(t) := t$, and $C_{n+1}(t) = 2tC_n(t) - C_{n-1}(t)$, $n \geq 1$. They are orthogonal with respect to the inner product defined by

$$\langle f, g\rangle_w := \int_{\mathbb{I}} f(t)g(t)w(t)dt\ ,$$

and they form a complete basis for $\mathcal{L}_2(\mathbb{I},\mathbb{R})$, i.e. their linear span is dense in $\mathcal{L}_2(\mathbb{I},\mathbb{R})$.

It can be shown, see Section 6 of [18], that if $f \in \mathcal{L}_2(\mathbb{I},\mathbb{R})$, there exist unique $\widetilde{f}_k \in \mathbb{R}$, $k \in \mathbb{N}$, such that the sequence $\left\{\sum_{k=0}^{n} \widetilde{f}_k C_k\right\}_{k\in\mathbb{N}}$ converges in the mean to f; if f is Lipschitz-continuous, then the sequence $\left\{\sum_{k=0}^{n} \widetilde{f}_k C_k\right\}_{k\in\mathbb{N}}$ converges absolutely and uniformly (see Theorem 3.1 p. 17 in [19]). Moreover, $\widetilde{f}_k = \langle f, C_k\rangle_w$ and $\{\widetilde{f}_k\}_{k\in\mathbb{N}} \in \ell_2(\mathbb{N},\mathbb{R})$. The coefficients $\widetilde{f}_k$ can be computed effectively and accurately using an interpolation procedure, see [19].

Denote by $\mathfrak{C}$ the infinite vector of polynomials

$$\mathfrak{C} := \begin{bmatrix} C_0 & C_1 & \dots \end{bmatrix}^\top\ , \tag{3}$$

and by $\widetilde{f}$ the infinite vector of coefficients, defined by

$$\widetilde{f} := \begin{bmatrix} \widetilde{f}_0 & \widetilde{f}_1 & \dots \end{bmatrix}\ . \tag{4}$$

With these positions, we write

$$f = \sum_{k=0}^{\infty} \widetilde{f}_k C_k = \widetilde{f}\mathfrak{C}\ . \tag{5}$$

We call the right-hand side of (5) the *polynomial transform* of f (see p. 69 of [20]).

For *vector* functions, we use the following notation. Denote by f_i, $i = 1, \ldots, n$ the i-th component of $f \in \mathcal{L}_2(\mathbb{I}, \mathbb{R}^n)$, and let $f_i = \sum_{k=0}^{\infty} \widetilde{f}_{i,k} C_k$ be its COPB representation; we write

$$f = \underbrace{\begin{bmatrix} \widetilde{f}_{1,0} & \widetilde{f}_{1,1} & \ldots \\ \vdots & \vdots & \ldots \\ \widetilde{f}_{n,0} & \widetilde{f}_{n,1} & \ldots \end{bmatrix}}_{=:\widetilde{f}} \mathfrak{C} \,. \tag{6}$$

If $f \in \mathcal{L}_2(\mathbb{I}, \mathbb{R})$ has a COPB representation (5), we call

$$\Pi_N(f) := \sum_{k=0}^{N} \widetilde{f}_k C_k \,, \tag{7}$$

the *truncation* or *projection* of the Chebyshev series for f *to degree* N. It can be shown that the *approximation error*

$$f - \Pi_N(f) = \sum_{k=N+1}^{\infty} \widetilde{f}_k C_k \,, \tag{8}$$

decays with N. For differentiable functions, the ∞-approximation error is $O(N^{-\nu})$, where ν is the order of differentiability, see Theorem 7.2 p. 53 of [19] and section 5.4.2 of [20], respectively. For analytic functions, the ∞-approximation error is $O(\rho^{-N})$ for some $0 < \rho < 1$, see Theorem 8.2 p. 57 of [19]. For $\mathcal{C}^{\infty}$-functions, the approximation error goes to zero faster than $O(N^{-k})$ for every finite k, see p. 47 of [20]. Similar results can be established for less regular functions defined in Sobolev spaces (see Appendix A.11 of [20]), and section 5.5.2 therein.

If $f \in \mathcal{L}_2(\mathbb{I}, \mathbb{R})$ is differentiable and $\frac{d}{dt} f \in \mathcal{L}_2(\mathbb{I}, \mathbb{R})$, then the *differentiation in the transform space* equality holds:

$$\frac{d}{dt} f = \sum_{k=0}^{\infty} \widetilde{f}_k \, \frac{d}{dt} C_k \,, \tag{9}$$

see p. 77 of [20]. Since C_k is a polynomial, $\frac{d}{dt} C_k$ is also a polynomial, and there exist $d_{k,j} \in \mathbb{R}$ such that

$$\frac{d}{dt} C_k = \sum_{j=0}^{\infty} d_{k,j} C_j \,, \; k \in \mathbb{N} \,. \tag{10}$$

Define

$$\left(\frac{d}{dt} \mathfrak{C} \right)^{\top} := \begin{bmatrix} \frac{d}{dt} C_0 & \frac{d}{dt} C_1 & \ldots \end{bmatrix} \,; \tag{11}$$

from (10), using some tedious calculations, it can be proved that the coefficients $d_{k,j}$ can be arranged in the infinite matrix

$$\mathcal{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \ldots \\ 1 & 0 & 0 & 0 & 0 & \ldots \\ 0 & 4 & 0 & 0 & 0 & \ldots \\ 3 & 0 & 6 & 0 & 0 & \ldots \\ 0 & 8 & 0 & 8 & 0 & \ldots \\ 5 & 0 & 10 & 0 & 10 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \,. \tag{12}$$

Note that the entries of $\mathcal{D}$ increase linearly with their indices.

We rewrite (10) as $\frac{d}{dt} \mathfrak{C} = \mathcal{D} \mathfrak{C}$, and (9) is equivalent with

$$\frac{d}{dt} f = \widetilde{f} \frac{d}{dt} \mathfrak{C} \quad = \widetilde{f} \mathcal{D} \mathfrak{C} \,. \tag{13}$$

In practice, finite truncations of the infinite vectors of coefficients and of the infinite differentiation matrices need to be used. Denote the coefficients of $\frac{d}{dt} f$ by $\widetilde{f}_k^{(1)}$:

$$\frac{d}{dt} f = \widetilde{f} \mathcal{D} \mathfrak{C} =: \underbrace{\begin{bmatrix} \widetilde{f}_0^{(1)} & \widetilde{f}_1^{(1)} & \ldots \end{bmatrix}}_{=:\widetilde{f}^{(1)}} \mathfrak{C} \,.$$

Let N be a fixed integer, partition

$$\widetilde{f} =: \begin{bmatrix} \Pi_N(f) & \widetilde{f}' \end{bmatrix} \quad \text{and } \widetilde{f}^{(1)} =: \begin{bmatrix} \Pi_N(\frac{d}{dt} f) & \widetilde{f}'^{(1)} \end{bmatrix} \,, \tag{14}$$

where $\widetilde{f}', \widetilde{f}'^{(1)} \in \mathbb{R}^{1 \times \infty}$; partition $\mathcal{D}$ accordingly:

$$\mathcal{D} =: \begin{bmatrix} \mathcal{D}_{11} & \mathcal{D}_{12} \\ \mathcal{D}_{21} & \mathcal{D}_{22} \end{bmatrix} \,.$$

Note that $\Pi_N(\frac{d}{dt} f) = \Pi_N(f) \mathcal{D}_{11} + \widetilde{f}' \mathcal{D}_{21}$ and that $\frac{d}{dt} (\Pi_N(f)) = \Pi_N(f) \mathcal{D}_{11}$; we call $\frac{d}{dt} (\Pi_N(f)) - \Pi_N \left(\frac{d}{dt} f \right)$ the *derivative approximation error*. In the following we assume that $\widetilde{f}_k$ goes quickly to zero relative to the linear increase of the entries of $\mathcal{D}$. It follows that a large enough N exists for which both $\widetilde{f}_k$ and $\widetilde{f}_k^{(1)}$ are below machine precision: then $\widetilde{f}$ and $\widetilde{f}^{(1)}$ can be considered to have compact support, and $\Pi_N(\frac{d}{dt} f) \simeq \frac{d}{dt} (\Pi_N(f)) = \Pi_N(f) \mathcal{D}_{11}$. For practical purposes the projection of the derivative equals the derivative of the projection, which is computable with finite-dimensional linear algebra operations.

III. All system trajectories from one

Consider an input-state-output representation

$$\begin{aligned} \frac{d}{dt} x &= Ax + Bu \\ y &= Cx + Du \,, \end{aligned} \tag{15}$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ and $D \in \mathbb{R}^{p \times m}$. Define

$$\mathcal{O}_k := \begin{cases} C & \text{if } k = 0 \\ \begin{bmatrix} \mathcal{O}_{k-1} \\ CA^k \end{bmatrix} & \text{if } k \geq 1 \end{cases} \,;$$

the *system lag* is defined by

$$\ell := \min\{k \in \mathbb{N} \mid \text{rank } \mathcal{O}_k = \text{rank } \mathcal{O}_{k-1}\} \,. \tag{16}$$

Evidently $\ell \leq n$; if (C, A) is observable, then ℓ is the observability index of the pair (C, A).

The following is Definition 1 in [1].

Definition 1: Let $\mathbb{I} = [-1, 1]$. $f : \mathbb{I} \to \mathbb{R}^m$ is *persistently exciting of order* k on $\mathbb{I}$ if

a) f is $(k-1)$-times continuously differentiable in $\mathbb{I}$;
b) For every $v := \begin{bmatrix} v_0 & \ldots & v_{k-1} \end{bmatrix} \in \mathbb{R}^{1 \times km}$ it holds that

$$v \begin{bmatrix} f(t) \\ f^{(1)}(t) \\ \vdots \\ f^{(k-1)}(t) \end{bmatrix} = 0 \; \forall \; t \in \mathbb{I} \Longrightarrow v_0, \ldots, v_{k-1} = 0 \,. \tag{17}$$

Associate to (15) its *external behavior*, defined by

$$\mathfrak{B} \;=\; \big\{\mathrm{col}(u,y) \in \mathcal{C}^\infty(\mathbb{I},\mathbb{R}^{m+p}) \mid \exists\; x \in \mathcal{C}^\infty(\mathbb{I},\mathbb{R}^n) \text{ s.t. } \mathrm{col}(u,x,y) \text{ satisfies (15)}\big\} \;. \tag{18}$$

Since the system (15) is linear, there is no finite-time escape to infinity, and consequently all trajectories of $\mathfrak{B}$ belong to $\mathcal{L}_2(\mathbb{I},\mathbb{R}^{m+p})$ and have a Chebyshev basis representation. We define $\Pi(\mathfrak{B})$, the *projection of* $\mathfrak{B}$ on the space of Chebyshev coefficient sequences, by

$$\Pi(\mathfrak{B}) \;:=\; \big\{\mathrm{col}(\widetilde{u},\widetilde{y}) \in \ell_2(\mathbb{N},\mathbb{R}^{m+p}) \mid \;\exists\; \mathrm{col}(u,y) \in \mathfrak{B} \text{ s.t. } \mathrm{col}(\widetilde{u},\widetilde{y}) = \Pi(\mathrm{col}(u,y))\big\} \;. \tag{19}$$

In the following, given the Chebyshev representation $\widetilde{f}$ of $f \in \mathcal{L}_2(\mathbb{I},\mathbb{R}^r)$, we denote by $\mathcal{W}_L(f)$ the $Lr \times \infty$ matrix

$$\mathcal{W}_L(\widetilde{f}) := \mathrm{col}\left(\widetilde{f}\mathcal{D}^j\right)_{j=0,\dots,L-1} \;. \tag{20}$$

The following is the continuous-time analogous of the main result of [3]. We denote the i-th row of a matrix M by $M_{i,:}$.

Theorem 1: Define $\mathfrak{B}$ by (18), and let $\mathrm{col}\,(u,y) \in \mathfrak{B}$. Assume that (A,B) is controllable, and that u is persistently exciting of order $L > \ell + n$. Define $\mathcal{W}_L(\widetilde{u})$, $\mathcal{W}_L(\widetilde{y})$ by (20). Then $\dim \mathcal{R}\,[\mathrm{col}(\mathcal{W}_L(\widetilde{u}),\mathcal{W}_L(\widetilde{y}))] = Lm + n =: d$.

Let $V_u \in \mathbb{R}^{Lm\times d}$ and $V_y \in \mathbb{R}^{Lp\times d}$ be such that $\mathrm{col}(V_u,V_y)$ is a basis matrix for $\mathcal{R}\,[\mathrm{col}(\mathcal{W}_L(\widetilde{u}),\mathcal{W}_L(\widetilde{y}))]$. Define $\Pi(\mathfrak{B})$ by (19). The following are equivalent:

1) $\mathrm{col}\,(\widetilde{u}',\widetilde{y}') \in \Pi(\mathfrak{B})$;
2) There exists $G \in \mathbb{R}^{d\times\infty}$ such that

$$\begin{bmatrix}\mathcal{W}(\widetilde{u}')\\ \mathcal{W}(\widetilde{y}')\end{bmatrix} \;=\; \begin{bmatrix}V_u\\ V_y\end{bmatrix} G \;, \tag{21}$$

3) There exists $G \in \mathbb{R}^{d\times\infty}$ such that

$$\begin{aligned}(V_uG)_{1,:} &= \widetilde{u}'\\ (V_yG)_{1,:} &= \widetilde{y}'\\ (V_uG)_{1,:}\mathcal{D}^i - (V_uG)_{i+1,:} &= 0\\ (V_yG)_{1,:}\mathcal{D}^i - (V_yG)_{i+1,:} &= 0 \;,\end{aligned} \tag{22}$$

$i = 1,\dots,L-1$.

Proof: See Theorem 4 p. 13 of [2]. ■

Theorem 1 shows how to use Chebyshev representations of a sufficiently informative trajectory to compute the Chebyshev representation of *all* possible system trajectories.

IV. A COPB BASED NOILC DESIGN FOR CONTINUOUS-TIME SYSTEMS

In this section we reformulate the NOILC design problem (2) without using any analytical model, only using 'sufficiently rich' input-output data.

A. Algorithm description and convergence properties

We first give a data-based characterization of all solutions of (2).

Proposition 1: Let $\mathrm{col}(u,y)$ be an input-output trajectory of (1). Assume that (A,B) is controllable, and that u is persistently exciting of order $L > \ell + n$. Define V_u and V_y as in Theorem 1, and denote the number of their columns by $d = Lm + n$.

u_{k+1} solves the NOILC design problem (2) on trial $k+1$ if and only if there exists $G_{k+1} \in \mathbb{R}^{d\times\infty}$ that solves the following optimisation problem

$$\begin{aligned}&\min_{G_{k+1}} \left\{\|r - (V_yG_{k+1})_{1,:}\mathfrak{C}\|_Q^2 + \|(V_uG_{k+1})_{1,:}\mathfrak{C} - u_k\|_R^2\right\}\\ &\text{s.t. } (V_uG_{k+1})_{1,:}\mathcal{D}^i - (V_uG_{k+1})_{i+1,:} = 0\\ &\quad\;\; (V_yG_{k+1})_{1,:}\mathcal{D}^i - (V_yG_{k+1})_{i+1,:} = 0 \;,\\ &\quad\;\; i = 1,\dots,L-1.\\ &\quad\;\; (V_yG_{k+1})_{1,:}\mathcal{D}^i\mathfrak{C}(-1) = 0,\; i = 0,\dots,q-1 \;.\end{aligned} \tag{23}$$

Moreover, the optimal input u_{k+1} on trial $k+1$ is

$$u_{k+1} = (V_uG_{k+1})_{1,:}\mathfrak{C} \;.$$

Proof: We first reformulate the NOILC optimization problem (2) on the $(k+1)^{th}$ trial substituting the definition of tracking error e_{k+1}:

$$\begin{aligned}&\mathrm{argmin}_{u_{k+1}} \left\{\|r - y_{k+1}\|_Q^2 + \|u_{k+1} - u_k\|_R^2\right\}\\ &s.t.\; \begin{bmatrix}-S_p & I\end{bmatrix}\begin{bmatrix}u_{k+1}\\ y_{k+1}\end{bmatrix} = 0\end{aligned} \tag{24}$$

The constraint appearing in (24) is equivalent to requiring that $\mathrm{col}(u_{k+1},y_{k+1})$ is an input-output trajectory of (1). Using Theorem 1 and the assumption of persistent excitation of u, $\mathrm{col}(u_{k+1},y_{k+1})$ is a system trajectory if and only if it satisfies (22). To complete the proof, recall that the system output is required to satisfy the zero initial condition, equivalently the last q constraints in (23). ■

Proposition 1 shows that solving the NOILC optimisation problem (2) for the input signal u_{k+1} in the original space $\mathcal{L}_2(\mathbb{I},\mathbb{R}^m)$ (using a system model) is equivalent to finding a solution of the transformed optimisation problem (23) using the input signal *COPB representation* computed from existing informative *data* as in Theorem 1. The resulting algorithm naturally inherits all the convergence properties of the model based NOILC algorithm, as shown in the following proposition (see [16], [17] for the proof).

Proposition 2: Denote by S_p^* the the Hilbert adjoint operator of the plant convolution operator S_p. The COPB based NOILC algorithm (23) has the following properties:

1) The tracking error norm converges monotonically

$$\|e_{k+1}\|_Q \le \|e_k\|_Q,\; \forall\; k \ge 0.$$

2) The tracking error sequence converges as follows

$$\lim_{k\to\infty} e_k = P_{\ker[S_p^*]}(e_0)$$

where $P_{\ker[S_p^*]}(e_0)$ is the orthogonal projection of the initial tracking error e_0 onto the subspace $\ker[S_p^*]$.

3) Consequently, if $r \in \mathcal{R}[S_p]$, or $\ker[S_p^*] = 0$, then perfect tracking of the reference trajectory is achieved:

$$\lim_{k\to\infty} e_k = 0.$$

4) Suppose that $r \in \mathcal{R}[S_p^*]$. The input sequence $\{u_k\}_{k\in\mathbb{N}}$ converges to the solution of the optimisation problem

$$\lim_{k\to\infty} u_k = \arg\min_u \left\{\|u - u_0\|_R^2 :\; r = S_pu\right\}$$

B. Implementation Procedure

To illustrate a procedure to compute the optimal input, we need to state some preliminary results.

Lemma 1: Let $C_m(\cdot)$, $C_n(\cdot)$ be two elements of the Chebyshev basis. Then

$$\int_{-1}^{1} C_m(t)C_n(t)dt = \begin{cases} 0, \text{ if } \ m+n \ \text{ is odd} \\ \frac{1}{1-(m-n)^2} + \frac{1}{1-(m+n)^2}, \text{ otherwise.} \end{cases}$$

Proof: This is equality 7.341.2 of [21]. ■

Lemma 2: Define $H \in \mathbb{R}^{\infty\times\infty}$ by

$$H_{m,n} := \int_{-1}^{1} C_m(t)C_n(t)dt \ , \ m,n \in \mathbb{N} \ . \tag{25}$$

Any finite principal submatrix of H is positive-definite.

Proof: Let $N \in \mathbb{N}$, and denote the $(N+1)\times(N+1)$ principal submatrix of H by H_N. H_N is the Gramian of the set $\{C_i(\cdot)\}_{i=0,\ldots,N}$ with respect to the standard $\mathcal{L}_2(\mathbb{I},\mathbb{R})$-inner product. Consequently, $H_N \succeq 0$. Moreover, since the elements of $\{C_i(\cdot)\}_{i=0,\ldots,N}$ are linearly independent, such Gramian is positive-definite. ■

Lemma 3: Let $R = R^\top \in \mathbb{R}^{m\times m}$, and define H by (25). Denote by H_N the $(N+1)\times(N+1)$ principal submatrix of H. If $R \succ 0$, then $H \otimes R \succ 0$.

Proof: Follows from the fact that the Kronecker product of two positive-definite matrices is also positive-definite. ■

We can now transform the optimization problem (23) in an equivalent version, easier to work with.

Proposition 3: Let $G_{k+1} \in \mathbb{R}^{d\times\infty}$. Denote by $\widetilde{u}_k \in \mathbb{R}^{m\times\infty}$ the infinite vector of Chebyshev coefficients of u_k, and by $\widetilde{r} \in \mathbb{R}^{p\times\infty}$ the infinite vector of Chebyshev coefficients of the reference trajectory r. Define

$$\begin{aligned} f(G_{k+1}) := & \, vec((V_uG_{k+1})_{1:})^\top (H\otimes R)\, vec((V_uG_{k+1})_{1,:}) \\ & + vec((V_yG_{k+1})_{1,:})^\top (H\otimes Q)\, vec((V_yG_{k+1})_{1,:}) \\ & - 2vec(\tilde{u_k})^\top (H\otimes R)\, vec((V_uG_{k+1})_{1,:}) \\ & - 2vec(\tilde{r})^\top (H\otimes Q)\, vec((V_yG_{k+1})_{1,:}) \ . \end{aligned} \tag{26}$$

The CPOB based NOILC optimisation problem (23) is equivalent to the following quadratic optimization problem with linear equality constraints:

$$\begin{aligned} & \min_{G_{k+1}\in\mathbb{R}^{d\times\infty}} f(G_{k+1}) \\ & \text{s.t. } (V_uG_{k+1})_{1,:}\mathcal{D}^i - (V_uG_{k+1})_{i+1,:} = 0 \\ & \quad\ \ (V_yG_{k+1})_{1,:}\mathcal{D}^i - (V_yG_{k+1})_{i+1,:} = 0 \ , \\ & \quad\ \ i = 1,\ldots,L-1, \\ & \quad\ \ (V_yG_{k+1})_{1,:}\mathcal{D}^i\mathfrak{C}(-1) = 0 \ , \\ & \quad\ \ i = 0,\ldots,q-1 \ . \end{aligned} \tag{27}$$

Proof: The claim is proved if we show that the performance index (26) is equivalent to (23). To show this, we expand the norm in (23); consider the second term in the performance index of (23):

$$\begin{aligned} & \|(V_uG_{k+1})_{1,:}\mathfrak{C} - u_k\|_R^2 \\ & = \langle (V_uG_{k+1})_{1,:}\mathfrak{C} - u_k, (V_uG_{k+1})_{1,:}\mathfrak{C} - u_k\rangle_R \\ & = \langle (V_uG_{k+1})_{1,:}\mathfrak{C}, (V_uG_{k+1})_{1,:}\mathfrak{C}\rangle_R + 2\langle (V_uG_{k+1})_{1,:}\mathfrak{C}, u_k\rangle_R \\ & \quad + \langle u_k, u_k\rangle_R \end{aligned}$$

The first term in the above equation is

$$\begin{aligned} & \int_{-1}^{1} ((V_uG_{K+1})_{1,:}\mathfrak{C})^\top R\ (V_uG_{K+1})_{1,:}\mathfrak{C}dt \\ & = \sum_{k=0}^{\infty}\sum_{j=0}^{\infty} \int_{-1}^{1} C_k(t)(V_uG_{k+1})_{1,k}^\top R (V_uG_{k+1})_{1,j} C_j(t)dt \\ & = \sum_{k=0}^{\infty}\sum_{j=0}^{\infty} (V_uG_{k+1})_{1,k}^\top R (V_uG_{k+1})_{1,j} \int_{-1}^{1} C_k(t)C_j(t)dt \\ & = \sum_{k=0}^{\infty}\sum_{j=0}^{\infty} (V_uG_{k+1})_{1,k}^\top R (V_uG_{k+1})_{1,j} H_{k,j} \\ & = vec((V_uG_{k+1})_{1,:})^\top \underbrace{\begin{bmatrix} RH_{0,0} & RH_{0,1} & RH_{0,2} & \ldots \\ RH_{1,0} & RH_{1,1} & RH_{1,2} & \ldots \\ \vdots & \vdots & \ddots & \ldots \end{bmatrix}}_{=H\otimes R} \\ & \quad \times vec((V_uG)_{1,:}) \ . \end{aligned}$$

Other terms in the performance index can be derived in a similar way. Also note that $\langle u_k, u_k\rangle_R$ and $\langle r, r\rangle_Q$ do not depend on the decision variable and thus can be neglected. This concludes the proof. ■

Standard computational tools can be used to solve the above quadratic optimization problem with linear constraints.

V. Numerical Example

To illustrate the design approach, consider the following simple dynamical system (from [1])

$$\frac{d}{dt}x = -x + u \ , \ y = x \ .$$

The system is operating over the time interval $\mathbb{I} = [-1, 1]$ with initial condition $x(-1) = 0$, and is required to track the reference trajectory $r(t) = \sin(\pi(t+1)/2)$ repeatedly. To solve the ILC design problem, we use the data-driven algorithm developed in this paper. The system has a lag of $\ell = 1$ and order $n = 1$. It is easy to verify that the input signal $u(t) = -3e^{-4t} - 2e^{-3t} - e^{-2t}$ is persistently exciting of order $L = 3 > \ell + n$ and consequently using the fundamental lemma described in Section III, all system trajectories can be obtained from $\text{col}(\tilde{u}, \tilde{y})$ where the output is $y(t) = e^{-4t} + e^{-3t} + e^{-2t} + e^{-t}$.

It can be shown that a basis of $\mathcal{R}\left[\text{col}(\mathcal{W}_L(\widetilde{u}), \mathcal{W}_L(\widetilde{y}))\right]$ (as defined in Theorem 1) is given by

$$\begin{bmatrix} V_u \\ V_y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Using this basis we can characterise all system trajectories, see (21).

To illustrate the performance of the developed COPB based NOILC algorithm, we simulate it with different choices of weighting parameters. The weighting matrices are chosen as $Q = I, R = 0.1, 0.05, 0.025, 0.0125, 0.00625$ and the initial input is chosen as $u_0 = 0$. The tracking error norms during the first 10 trials are given in Figure 1 to illustrate the performance of the algorithm and the effect of R. From the figure, it can be clearly seen that as predicted by Proposition 2, the tracking error norm does converge monotonically. Furthermore, decreasing R leads to faster error convergence as larger input change is allowed between trials. We have also simulated the algorithm over 1000 trials to show the long term performance. Inspection on the results on the 1000^{th} trial suggests that very good (almost perfect) tracking has been achieved, and the output signals are not distinguishable from the reference (figures are omitted here for space reasons).

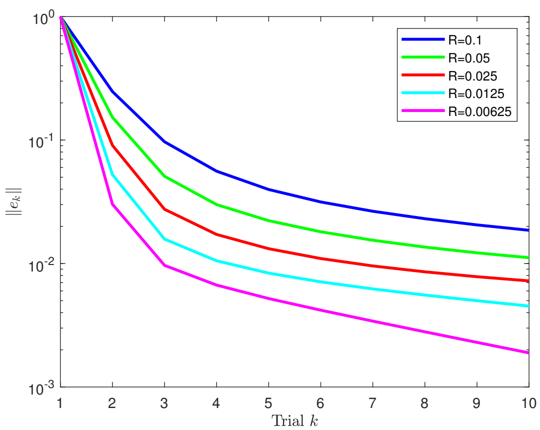

Fig. 1. Tracking error norm over 10 trials with different R

VI. Conclusions and Open Problems

We have developed a data-driven NOILC algorithm for continuous-time systems. Using Chebyshev polynomial bases, we can represent all system trajectories from one sufficiently rich trajectory. Using this representation, we eliminated the need for an explicit model in norm optimal iterative learning control design and gave a truly data-driven solution to the NOILC problem. We first stated an optimisation problem in a space of the Chebyshev coefficients of the underlying input and output signals. We then showed how to use standard quadratic optimization techniques to compute a solution and we rigorously analysed the convergence properties of the resulting design. Numerical simulations were presented to illustrate the performance of the design.

We presented only some of our preliminary results on data-driven ILC design for continuous-time systems: a number of other research topics are being pursued, including the incorporation of system constraints, consideration of nonlinear systems as well as noise and disturbances, extension to other ILC design problems (e.g. point-to-point tracking), and the use of other basis functions. These results will be reported elsewhere.

References

[1] P. Rapisarda, M. K. Camlibel, and H. J. van Waarde, "A persistency of excitation condition for continuous-time systems," *IEEE Control Systems Letters*, vol. 7, pp. 589–594, 2023.

[2] P. Rapisarda, H. J. van Waarde, and M. K. Camlibel, "A "fundamental lemma" for continuous-time systems, with applications to data-driven simulation," *Systems & Control Letters*, vol. 179, p. 105603, 2023.

[3] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[4] H. J. Van Waarde, J. Eising, H. L. Trentelman, and M. K. Camlibel, "Data informativity: a new perspective on data-driven analysis and control," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4753–4768, 2020.

[5] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2020.

[6] 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, 2012.

[7] J. Bolder, S. Kleinendorst, and T. Oomen, "Data-driven multivariable ILC: enhanced performance by eliminating L and Q filters," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 12, pp. 3728–3751, 2018.

[8] R. de Rozario and T. Oomen, "Data-driven iterative inversion-based control: Achieving robustness through nonlinear learning," *Automatica*, vol. 107, pp. 342–352, 2019.

[9] Z. Jiang and B. Chu, "Norm optimal iterative learning control: A data-driven approach," *IFAC-PapersOnLine*, vol. 55, no. 12, pp. 482–487, 2022.

[10] R. Chi, Z. Hou, S. Jin, and B. Huang, "An improved data-driven point-to-point ilc using additional on-line control inputs with experimental verification," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 4, pp. 687–696, 2017.

[11] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems," in *The 23rd IEEE Conference on Decision and Control*. IEEE, 1984, pp. 1064–1069.

[12] K. L. Moore, *Iterative learning control for deterministic systems*, ser. Advances in Industrial Control. Springer London, 2012.

[13] A. Tayebi, "Adaptive iterative learning control for robot manipulators," *Automatica*, vol. 40, no. 7, pp. 1195–1203, 2004.

[14] S. Devasia, "Iterative machine learning for output tracking," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 516–526, 2019.

[15] K.-S. Kim and Q. Zou, "A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1767–1777, 2013.

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

[17] N. Amann, D. 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.

[18] G. Sansone, *Orthogonal functions*. Interscience Publishers, 1959, vol. 9.

[19] L. N. Trefethen, *Approximation Theory and Approximation Practice, Extended Edition*. SIAM, 2019.

[20] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods: Fundamentals in Single Domains*. Springer-Verlag, 2006.

[21] I. Gradshteyn and I. Ryzhik, *Table of Integrals, Series, and Products*, 8th ed., D. Zwillinger, Ed. Elsevier Academic Press, Waltham, MA, 2015.