Spatial Iterative Learning Control: Output Tracking

Merid Lješnjanin [*], **Ying Tan** [*], **Denny Oetomo** [*] and **Christopher T. Freeman** [**]

[*] *University of Melbourne, Australia, (e-mail: x@unimelb.edu.au, x ∈ {merid.ljesnjanin, yingt, doetomo}).*
[**] *University of Southampton, UK, (e-mail: cf@ecs.soton.ac.uk)*

Abstract: This paper focuses on tracking *spatially repeatable* tasks. In addition, these tasks are *not* necessarily *temporally repeatable* in the sense that the *finite* length of the corresponding time interval may *change* with each repetition. Because of that, the standard Iterative Learning Control (ILC) framework is not directly applicable. Namely, the standing assumption that the *finite* length of the time interval is *fixed* with each repetition, is violated. Motivated by *human motor learning*, this paper proposes a Spatial ILC (SILC) framework which leverages the spatial repeatability. In particular, the concept of *spatial projection*, closely related to *temporal rescaling*, is proposed. This allows to *spatially* relate the relevant information from the past repetition to the present repetition. To demonstrate the proposed framework, a class of nonlinear time-varying systems with relative degree zero is selected. In particular, using contraction mapping technique, it is shown that under appropriate assumptions, the corresponding tracking error converges under the proposed SILC control law. Finally, simulation results support the obtained result.

Keywords: Spatial Iterative Learning Control, Spatial Projection, Output Tracking, Contraction Mapping.

1. INTRODUCTION

Iterative Learning Control (**ILC**) is a control paradigm which focuses on problems which involve tasks that *repeat*. One classical example is the problem of packaging assembly line which includes several repeatable tasks performed by a robot, such as placing products in a box and sticking labels on it. After first defining a *desirable* way of performing each task, the goal of ILC is then to improve the corresponding *transient* behavior and to achieve *perfect* tracking. This is done by exploiting task repeatability through application of an appropriate *learning* mechanism. *Knowledge* that is acquired via *learning* enables the construction of a control law which effectively and efficiently [1] accomplishes the transient and tracking goals. Vast literature associated with ILC exists and it spans both, the theory and the practice; e.g., cf.: survey papers Ahn et al. (2007); Wang et al. (2009); Xu (2011), books Xu and Tan (2003); Bien and Xu (2012); Moore (2012) and references therein.

Indeed, ILC has been successfully applied to various problems involving repeatable tasks, as documented in the literature above. However, it is important to notice that each repeatable task satisfies the ILC standing assumption. Namely, the *finite* length of the time interval over which the task evolves is *fixed* for each repetition. Unfortunately, there are many problems where repeatable tasks do not satisfy this assumption, rendering *standard* ILC inapplicable. To illustrate this consider a repeatable task which consists of tracking a straight line of finite length from a point A to a point B; in a two dimensional space. Indeed, there are (many) examples where this task can be performed within the *fixed* finite time interval at each repetition; e.g., many industrial robots are capable of performing such a task. However, for certain problems such as the rehabilitation process, this assumption cannot be satisfied; Shmuelof et al. (2012) demonstrates this observation in a similar setup even for healthy subjects. For instance, consider stroke rehabilitation. To recover, during each session, a patient performs the same task (movement) sufficiently many times. Now, imagine that the rehabilitation process includes tracking the straight line mentioned above *without* any robotic assistance. It is unreasonable to expect that at each repetition, the patient will execute this task (movement) in a *fixed* finite time interval. Depending on the injury related factors, such as pain, range of motion, motivation, focus, fatigue, and the stage of the rehabilitation, it is very likely that the duration of the time interval will vary with each repetition. Indeed, it is natural to expect that over sufficiently many repetitions the duration of the time interval will converge to some value. Hence, for *repeatable* tasks which have its *spatial* constituent *fixed* while its *temporal* constituent *variable*, the standing assumption needs to be *relaxed* in order to leverage the existing ILC results. A natural relaxation is to consider *bounded* instead of *fixed* time intervals; notice that *fixed* time intervals are always *bounded*. This relaxation enables one to formulate problems so that the focus is on the *spatial* component in the corresponding ILC analysis and design.

[1] Other control methodologies can be used to solve this problem. However, in most cases they do not fully exploit the repeatability feature, and thus result in a less efficient and effective control laws.

Related industrial research on Spatial ILC (SILC) is largely driven with particular industrial problems. For instance, these include switched reluctance motors Sahoo et al. (2007), nonlinear rotary systems Yang and Chen (2009) and micro-additive manufacturing Hoelzle and Barton (2014). Some theoretical results can be found in Janssens et al. (2013) and Moore et al. (2007). The former reference deals with the problem of minimization of the total tracking error of the repeated tasks with output constraints. On the other hand, the latter reference explores the freedom of not specifying temporal information related to the spatial movement and shows some significant practical yields. Neither reference provides a general analysis and design framework though.

This paper aims to address this need for a class of nonlinear time-varying systems with relative degree zero. Towards that goal, first a framework that focuses on the spatial aspect of the corresponding task is proposed. At the core of the proposed framework is the concept of *spatial projection* which is closely related to appropriate *temporal rescaling*. To the best of authors' knowledge the only reference that utilizes a similar idea in the ILC setting is reported in Kawamura and Sakagami (2002). Namely, in Kawamura and Sakagami (2002), ILC and time-scale transformation are used to identify added mass, drag and buoyancy in the dynamics of the underwater robots. The considered application is very specific and moreover, no general analysis and design framework is provided. In the present paper, after the introduction of the spatial projection, the corresponding SILC controller is proposed. The conducted convergence analysis exploits the contraction mapping idea, cf.: Xu and Tan (2003). It is shown that under the standard ILC assumptions (akin to those in (Xu and Tan, 2003, Chapter 2)) for the considered class of systems and the appropriate assumptions related to the *spatial projection*, spatial tracking is achieved. Finally, simulation results demonstrate the convergence, even under disturbances due to computer implementation.

The paper is organized as follows. In the sequel, first the mathematical preliminaries and notational conventions are provided which is followed with the formulation of the problem in Section 2. Then, in Section 3, the corresponding assumptions are provided and main results are stated. This is followed with Section 4, in which the simulations results demonstrate the claims of the main results. Finally, in Section 5, the concluding remarks are documented.

Preliminaries

Symbols $\mathbb{Z}$ and $\mathbb{R}$, respectively, denote the set of integer and real numbers. A set $\mathbb{D} \in \{\mathbb{Z}, \mathbb{R}\}$ which elements are $\diamond$-bounded, $\diamond \in \{\leq, <, >, \geq\}$, by an element $a \in \mathbb{D}$, is denoted with $\mathbb{D}_{\diamond a} := \{x \in \mathbb{D} : x \diamond a\}$. The set of natural numbers is then defined as $\mathbb{N} := \mathbb{Z}_{>0}$ while $\mathbb{N}_0 := \mathbb{Z}_{\geq 0}$. The Cartesian product between sets $\mathbb{D}_j$, $j \in \{1, \ldots, d\}$, $d \in \mathbb{N}$, is denoted as $\mathbb{D}_1 \times \cdots \times \mathbb{D}_d$. However, when $\mathbb{D}_j = \mathbb{D}$, $\forall j \in \{1, \ldots, d\}$, $d \in \mathbb{N}$, the shorthand notation $\mathbb{D}^d$ is used. Its elements are denoted as ordered d-tuples, i.e., $(x_1, \ldots, x_d)$ where $x_j \in \mathbb{D}$, $\forall j \in \{1, \ldots, d\}$, $d \in \mathbb{N}$, and throughout the document, wherever applicable, this notation is used to denote column vectors. Several norms are used throughout the document. First, $\|x\|_q^q := \sum_{j=1}^n |x_j|^q$, $(q, n) \in [1, \infty) \times \mathbb{N}$, where $|\cdot|$ denotes the standard Euclidean norm. In addition, for a given $[0, \mathsf{T}] \mapsto x(t) \in \mathbb{R}^n$, $\mathsf{T} > 0$, its the supremum norm is defined as $\|x\|_{\mathsf{s}} := \max_{t \in [0,\mathsf{T}]} \|x(t)\|_1$, while its time-weighted norm is defined as $\|x\|_\lambda := \max_{t \in [0,\mathsf{T}]} e^{-\lambda \cdot t} \cdot \|x(t)\|_1$, where $\lambda > 0$.

2. PROBLEM FORMULATION

For the sake of clear problem formulation, first, a generic nonlinear time-varying system with relative degree zero, within the context of a *traditional* ILC framework, is considered; e.g., cf. Xu and Tan (2003). The follow up discussion then illustrates why such approach is not necessarily applicable for *spatially* repeatable tasks. Then, a *spatial projection* mapping is defined and an example of how this concept might be useful in characterizing these tasks is provided. Finally, using the concept of the spatial projection, the problem formulation is provided.

So, consider the following system, evolving over a two dimensional temporal space [2]

$$\dot{x}(i,t) = f\big(t, x(i,t), u(i,t)\big), \tag{1a}$$
$$y(i,t) = h\big(t, x(i,t), u(i,t)\big), \tag{1b}$$

where,

$$\big(i,t\big) \in \mathbb{N}_0 \times [0, \mathsf{T}],\ \mathsf{T} \in \mathbb{R}_{>0}, \tag{2}$$

is an element of a two dimensional temporal space. Variables $x \in \mathbb{B}_x \subseteq \mathbb{R}^n$, $u \in \mathbb{B}_u \subseteq \mathbb{R}^m$ and $y \in \mathbb{B}_y \subseteq \mathbb{R}^p$, respectively, are the system state, input and output, while $(n, m, p) \in \mathbb{N}^3$. Further, respectively, $f : \mathbb{R}_{\geq 0} \times \mathbb{B}_x \times \mathbb{B}_u \to \mathbb{B}_x$ and $h : \mathbb{R}_{\geq 0} \times \mathbb{B}_x \times \mathbb{B}_u \to \mathbb{B}_y$, are the system state and output mappings. Correspondingly, consider a model of a desired behavior,

$$\dot{x}_{\mathsf{d}}(t) = f\big(t, x_{\mathsf{d}}(t), u_{\mathsf{d}}(t)\big), \tag{3a}$$
$$y_{\mathsf{d}}(t) = h\big(t, x_{\mathsf{d}}(t), u_{\mathsf{d}}(t)\big), \tag{3b}$$

where $x_{\mathsf{d}} \in \mathbb{B}_{x_{\mathsf{d}}} \subseteq \mathbb{R}^n$, $u_{\mathsf{d}} \in \mathbb{B}_{u_{\mathsf{d}}} \subseteq \mathbb{R}^m$ and $y_{\mathsf{d}} \in \mathbb{B}_{y_{\mathsf{d}}} \subseteq \mathbb{R}^p$, respectively, are the desired behavior state, input and output. Important to note in (3) is that *only* y_{d} is given and it is assumed that there *exists* u_{d} such that (3) holds; e.g., cf. (Xu and Tan, 2003, Assumption 2.2, Chapter 2).

Now, a standard ILC objective is to construct an ILC control law so that desired behavior is *learned* after sufficiently many iterations. One way of explicitly capturing this is by achieving $\lim_{i \to \infty} \Delta y(i,t) = (0, \ldots, 0)$, where $\Delta y(i,t) = y_{\mathsf{d}}(t) - y(i,t)$. For a relative degree zero systems, that satisfy standard ILC assumptions and some mild smoothness conditions (cf. (Xu and Tan, 2003, Chapter 2)), the so-called P-type ILC control law that achieves this limit is given as

$$u(i+1,t) = u(i,t) + \kappa \cdot \Delta y(i,t), \tag{4}$$

where $\kappa \in \mathbb{R}$ is the *learning control gain*.

Remark 1. (Relative Degree). Note that dynamic systems with arbitrary relative degree are addressed with different types of ILC control laws. For instance, the so-called D-type ILC control law can be used for dynamic systems with relative degree one; cf. Ahn et al. (2007); Wang et al. (2009); Xu (2011); Xu and Tan (2003); Bien and Xu (2012); Moore (2012) and references therein for different types of ILC control laws. □

[2] Notice that $\dot{x} := dx/dt$.

Now, consider again the rehabilitation example mentioned in Section 1. Namely, consider the example of a recovering patient. As explained, during each treatment (session) a patient repeats sufficiently many times a given movement in order to gradually recover. Recall that due to many factors, there is no guarantee that each repetition of a given movement is performed at the same rate and thus within the same time interval. *Traditional* ILC approach, succinctly introduced above, fails to address the problem in the rehabilitation example because (2) is not satisfied. Thus, one needs a more general framework which focuses on the *spatial* information. There are different ways to handle this. The approach pursued in this paper exploits the concept of *spatial projection* mapping defined below.

Definition 1. (Spatial Projection). Consider $s(t) \in \mathbb{R}^l$, $l \in \mathbb{N}$, where $t \in [0, T]$, $T > 0$. Given $\epsilon > 0$, a spatial projection mapping is defined as,

$$\sigma_\epsilon\left(s(t)\right) := s(\epsilon \cdot \tau), \tag{5}$$

where $\tau \in \left[0, \frac{T}{\epsilon}\right]$. □

Remark 2. (Spatial Projection). Notice that the spatial projection is obtained via *temporal rescaling* (*rate* ϵ) and manipulation of the original time domain. The following example provides the reasoning behind the idea of spatial projection. In addition, it serves as a prelude to the introduction of SILC problem formulation. □

Example 1. (Spatial Projection). To illustrate the application of the *spatial projection* for the characterization of spatially equivalent trajectories, consider the task of drawing a straight line, of finite length, from a fictional point A to the fictional point B; see Fig. 1. Instead of using a dynamical model which captures this task, to simplify the explanation consider just a static mapping. For instance, each point on a straight line (from A to B) can be defined as $p(t) := \exp(t)$. To capture the requirement of drawing a line of finite length $B - A$, let $t \in [0, T]$ where $T > 0$ is such that $\exp(T) = B - A$; note that because $t \in [0, T]$, $A = \exp(0) = 1$, however, this is not that relevant for the narrative. In Fig. 1 this corresponds to the green color. Let the corresponding line be a *nominal* line. The nominal line is drawn with a corresponding *rate* and it takes T amount of time to finish drawing it. Now, let $(\epsilon', \epsilon'') \in \mathbb{R}^2_{>0}$ be given and let $\epsilon' > 1 > \epsilon''$. It follows,

$$\sigma_{\epsilon'}(p(t)) = p(\epsilon' \cdot \tau),\ \tau \in \left[0, \frac{T}{\epsilon'}\right], \tag{6a}$$

$$\sigma_{\epsilon''}(p(t)) = p(\epsilon'' \cdot \tau),\ \tau \in \left[0, \frac{T}{\epsilon''}\right]. \tag{6b}$$

The line that corresponds to (6a) is associated with blue color while the line that corresponds to (6b) is associated with red color. As illustrated, in all three cases the *spatial* information is *preserved*; e.g., the straight line from A to B which is in the black color. However, *temporally*, they to differ. In particular, the difference is the rate at which each line is drawn and time it took to complete each line; e.g., it is possible to *temporally* characterize them. In particular, because $\epsilon'' < 1 < \epsilon'$, the line associated with the green color is drawn at a slower rate than the line associated with the blue color and at the faster rate than the line associated with the red color. To illustrate this even further, at the ("universal") time instant τ_*, the corresponding point on the straight line from A to B associated with the blue color is ahead of the corresponding point on the line from A to B associated with the green color which is ahead of the corresponding point on the line from A to B associated with the red color. Correspondingly the times it takes to complete the circle associated with the blue, green and red color, respectively, satisfy $T/\epsilon' < T < T/\epsilon''$. △

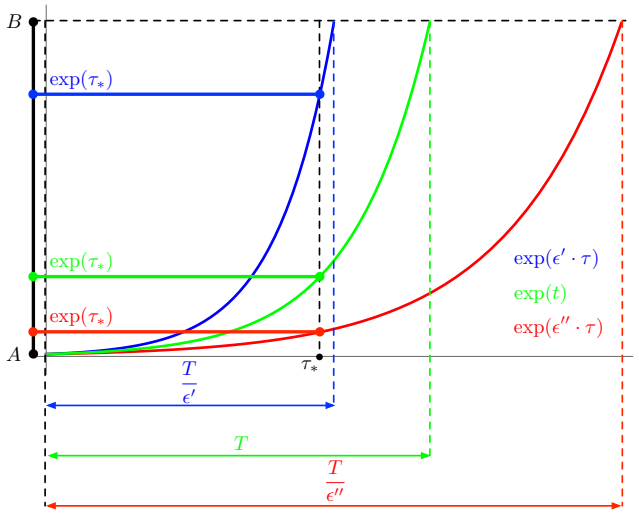

Fig. 1. Preserving spatial information.

Remark 3. The concept of *spatial projection* mapping (see Definition 1) has also been explored for a more specific class of nonlinear systems. In particular, in Lješnjanin, Merid and Tan, Ying and Oetomo, Denny and Freeman, T., Christopher (2016), a class of nonlinear uncertain systems with input saturation is considered. Even though the same concept of *spatial projection* mapping is being explored, the corresponding SILC problem formulation and convergence analysis differs from the present one. Namely, the proposed SILC controller is more complex while the convergence of the *state* error is demonstrated using a Composite Energy Function technique (e.g., cf. Xu and Tan (2003)). □

Now, consider the following dynamical system,

$$\dot{x}\big(i, \epsilon(i) \cdot t\big) = \epsilon(i) \cdot f\big(\epsilon(i) \cdot t, x(i, \epsilon(i) \cdot t), u(i, \epsilon(i) \cdot t)\big), \tag{7a}$$

$$y\big(i, \epsilon(i) \cdot t\big) = h\big(\epsilon(i) \cdot t, x(i, \epsilon(i) \cdot t), u(i, \epsilon(i) \cdot t)\big), \tag{7b}$$

where,

$$\big(i, t\big) \in \mathbb{N}_0 \times [0, \mathsf{T}_i], \tag{8}$$

is an element of a two dimensional temporal space with,

$$\mathsf{T}_i := \frac{\mathsf{T}}{\epsilon(i)},\ \mathsf{T} \in \mathbb{R}_{>0}, \tag{9}$$

and where,

$$\epsilon : \mathbb{N}_0 \to \mathbb{R}_{>0}, \tag{10}$$

is a mapping that captures at what rate the dynamics of (7a) evolves at each iteration i. More about this mapping is included below. Before proceeding further, notice that the system (7) can be interpreted as the spatial projection (see Definition 1 and Example 1) of the system (1) with respect to the mapping (10). To help the reader, in the sequel, the system (7) is succinctly represented as,

$$\dot{x}_{i|\epsilon_i} = \epsilon_i \cdot f(\epsilon_i \cdot t, x_{i|\epsilon_i}, u_{i|\epsilon_i}), \tag{11a}$$

$$y_{i|\epsilon_i} = h(\epsilon_i \cdot t, x_{i|\epsilon_i}, u_{i|\epsilon_i}). \tag{11b}$$

Correspondingly, consider a model of a desired behavior,

$$\dot{x}_{\mathsf{d}}\big(\epsilon_i(i)\cdot t\big) = \epsilon_i(i)\cdot f\big(\epsilon(i)\cdot t, x_{\mathsf{d}}(\epsilon_i(i)\cdot t), u_{\mathsf{d}}(\epsilon_i(i)\cdot t)\big), \quad (12a)$$

$$y_{\mathsf{d}}\big(\epsilon_i(i)\cdot t\big) = h\big(\epsilon(i)\cdot t, x_{\mathsf{d}}(\epsilon_i(i)\cdot t), u_{\mathsf{d}}(\epsilon_i(i)\cdot t)\big). \quad (12b)$$

Notice that the desired behavior model (12) is also accordingly spatially projected. Similarly as for (7), in the rest of the manuscript, the desired behavior model (12) is compactly represented as

$$\dot{x}_{\mathsf{d}|\epsilon_i} = \epsilon_i \cdot f(\epsilon_i \cdot t, x_{\mathsf{d}|\epsilon_i}, u_{\mathsf{d}|\epsilon_i}), \quad (13a)$$

$$y_{\mathsf{d}|\epsilon_i} = h(\epsilon_i \cdot t, x_{\mathsf{d}|\epsilon_i}, u_{\mathsf{d}|\epsilon_i}). \quad (13b)$$

Now, the ILC objective is the same. In particular, construct an ILC control law so that desired behavior is *learned* after sufficiently many iterations. However, unlike in the *traditional* ILC framework, now there is an extra variable ϵ (see (10)), which directly affects (7)-(9) and (13). Thus, before stating precisely the ILC objective it is necessary to elaborate more on it. Firstly, the variable ϵ (10) captures at what rate the dynamics (11a) evolves at the iteration i. In addition to that, it also directly affects the length of the corresponding time interval, see (9). Now, restricting attention to the problems similar to the rehabilitation (see the corresponding discussion in Section 1), it seems reasonable to assume that (10) converges to some value which is stated in the following assumptions.

Assumption 1. (Rate Convergence). Consider (10). There exists a tuple $(\epsilon_*, \eta) \in \mathbb{R}_{>0} \times (0,1)$, such that,

$$\|\epsilon_{i+1} - \epsilon_*\|_1 \leq \eta \cdot \|\epsilon_i - \epsilon_*\|_1, \quad (14)$$

holds for each $i \in \mathbb{N}_0$. □

Remark 4. (Rate Convergence). For instance, using again the patient example, the element ϵ_* can represent the rate of execution of a given task for a (healthy) subject. Section 4 explores different rate profiles which satisfy Assumption 1. □

Remark 5. (Rate Boundedness). Notice that because of Assumption 1 there exists $\bar{\epsilon} \in \mathbb{R}_{>0}$, such that $\epsilon_i \leq \bar{\epsilon}$ holds for each $i \in \mathbb{N}_0$. This means that the length of the corresponding time interval, see (9), is **upper bounded** as well. Also, note that the upper bound on rate ϵ means that considered dynamics cannot evolve infinitely fast, which definitely applies to humans. □

Remark 6. (Rate Dynamics). Notice that the rate ϵ, as defined in (10), is a static mapping which is assumed to be **known** for each iteration $i \in \mathbb{N}_0$. For rehabilitation-like examples, the Assumption 1 seems reasonable, even without the full knowledge of (10). However, the conducted convergence analysis and the proposed SILC control law (see (18)) depend on it. Thus, the future work will address the design and analysis where (10) is not fully available but estimates are. In addition to that, a more complex description of the rate ϵ will be considered, e.g., it will be represented as an output of a dynamical system which might be affected with data from the model (11) and/or (13). □

Finally, the ILC objective is to construct an ILC control law such that,

$$\lim_{i\to\infty} \Delta y_{i|\epsilon_i} = (0,\ldots,0), \quad (15)$$

where

$$\Delta y_{i|\epsilon_i} := y_{\mathsf{d}|\epsilon_i} - y_{i|\epsilon_i}. \quad (16)$$

Remark 7. (Error Limits). Notice that because of Assumption 1, it follows that the error limit (15), is equivalent to $\lim_{i\to\infty} \Delta y_{i|\epsilon_i} = \lim_{i\to\infty} \Delta y_{i|\epsilon_*} = (0,\ldots,0)$. □

One attempt in constructing such a law is to directly and correspondingly apply the reasoning behind (4). In particular, one might be tempted to apply the following ILC-like control law,

$$u_{i+1|\epsilon_{i+1}} := u_{i|\epsilon_i} + \kappa \cdot \Delta y_{i|\epsilon_i}. \quad (17)$$

However, unfortunately, when $\epsilon_i \neq c$, $c \in \mathbb{R}_{>0}$, $\forall i \in \mathbb{N}_0$, the length of $[0, \mathsf{T}_i]$ is not *fixed* (see (9)). Therefore, two consecutive time intervals are not necessarily the same length and using (17) to compute control values for the time interval $[0, \mathsf{T}_{i+1}]$ becomes problematic. Firstly, if $\mathsf{T}_{i+1} > \mathsf{T}_i$, then for the time interval $[\mathsf{T}_{i+1} - \mathsf{T}_i, \mathsf{T}_{i+1}]$ there are no control values from the time interval $[0, \mathsf{T}_i]$ that can be used; i.e., they are only defined for the time interval $[0, \mathsf{T}_i]$. On the other hand, if $\mathsf{T}_{i+1} < \mathsf{T}_i$, then the control values from the time interval $[\mathsf{T}_{i+1} - \mathsf{T}_i, \mathsf{T}_{i+1}]$ are not even use. To explain this more precisely, consider Example 1. For simplicity reasons, assume that the data associated to the blue color corresponds to the data from the time interval $[0, \mathsf{T}_i]$. Let $\mathsf{T}_{i+1} > \mathsf{T}_i$ and let the corresponding time interval $[0, \mathsf{T}_{i+1}]$ be associated to the green color; e.g., $\mathsf{T}_{i+1} = T$ while $\mathsf{T}_i = T/\epsilon'$. Now, using the "blue data" to compute the corresponding data for the time interval $[0, \mathsf{T}_{i+1}]$ becomes a problem because for the time interval $[\mathsf{T}_{i+1} - \mathsf{T}_i, \mathsf{T}_{i+1}]$ (in Example 1 this corresponds to $[T - T/\epsilon', T]$) the "blue data" is not defined. Conversely, now assume that the data associated to the green color corresponds to the data from the time interval $[0, \mathsf{T}_i]$. Let $\mathsf{T}_{i+1} < \mathsf{T}_i$ and let the corresponding time interval $[0, \mathsf{T}_{i+1}]$ be associated to the blue color; e.g., $\mathsf{T}_{i+1} = T/\epsilon'$ while $\mathsf{T}_i = T$. Now, using the "green data" to compute the corresponding data for the time interval $[0, \mathsf{T}_{i+1}]$ becomes also a problem. Namely, the "green data" that corresponds to the time interval $[\mathsf{T}_i - \mathsf{T}_{i+1}, \mathsf{T}_i]$ is not used. To address both issues the following ILC control law is proposed,

$$\begin{aligned} u_{i+1|\epsilon_{i+1}} &:= \sigma_{\epsilon_{i+1}}\left(u_{i|\epsilon_i} + \kappa \cdot \Delta y_{i|\epsilon_i}\right) \\ &= u_{i|\epsilon_{i+1}} + \kappa \cdot \Delta y_{i|\epsilon_{i+1}}. \end{aligned} \quad (18)$$

Remark 8. As indicated in Remark 6, the proposed SILC control law (18) depends on the knowledge of (10), i.e., it needs ϵ_{i+1}. The future work will explore different approaches of obtaining this quantity. □

The idea behind the ILC control law (18) is to preserve spatial reference. This appears to resemble human approach towards spatial tracking, i.e., when spatial reference is *fixed* and there are no time constraints.

Remark 9. (Relative Degree Zero). Notice that because relative degree zero systems are considered, the ILC control law (18) needs only to spatially project the data from the previous iteration. □

In the following section the proposed control law is analyzed in a more detail.

3. ANALYSIS

The analysis leverages the existing results in the ILC literature which use the contraction mapping approach to show convergence, e.g., cf. (Xu and Tan, 2003, Chapter 2). In fact, the analysis presented in this section follows closely the corresponding analysis from (Xu and Tan, 2003, Chapter 2). Hence, the proof is omitted for space reasons. In order to claim (15) under the proposed control

law (18), additional (sufficient) assumptions are stated below. The first one is related to the initial states of (11) and (13). This assumption, dubbed Identical Initialization Condition (IIC), is often used in the ILC literature.

Assumption 2. (IIC). Consider (7a). The following holds,

$$x(\epsilon_i \cdot 0, i) = x^{\circ}, \tag{19}$$

for each $i \in \mathbb{N}_0$. □

To simplify the presentation, attention is restricted to the case where $m = p = 1$, i.e., Single-Input-Single-Output. Extension to Multiple-Input-Multiple-Output, where $m = p$, can be obtained by following (Xu and Tan, 2003, Chapter 4), however for space and simplicity reasons it is omitted. Now, the mapping describing the state evolution of (11a) and (13a) is assumed to be Globally Lipshitz Continuous (GLC). In particular, the following holds.

Assumption 3. (GLC). Consider (11a) and (13a). There exists $\mathsf{L} > 0$ such that,

$$\|f(\epsilon\cdot t, \tilde{x}, \tilde{u}) - f(\epsilon\cdot t, \bar{x}, \bar{u})\|_1 \leq \mathsf{L}\cdot\big(\|\tilde{x}-\bar{x}\|_1 + \|\tilde{u}-\bar{u}\|_1\big), \tag{20}$$

holds for all $\big((\epsilon\cdot t, \tilde{x}, \tilde{u}), (\epsilon\cdot t, \bar{x}, \bar{u})\big) \in (\mathbb{R}_{\geq 0} \times \mathbb{R}^n \times \mathbb{R})^2$ and all $\epsilon \in \mathbb{R}_{>0}$. □

Certain smoothness properties of the output map in (11b) and (13b) are assumed.

Assumption 4. (h smoothness). Consider (11b) and (13b). There exists a tuple $(H_1, H_2, H_3) \in \mathbb{R}^3_{>0}$ such that,

$$\left\| \frac{\partial}{\partial x_{j|\epsilon_i}} h(\epsilon_i \cdot t, x_{j|\epsilon_i}, u_{j|\epsilon_i}) \right\|_M \leq H_1, \tag{21a}$$

$$H_2 \leq \frac{\partial}{\partial u_{j|\epsilon_i}} h(\epsilon_i \cdot t, x_{j|\epsilon_i}, u_{j|\epsilon_i}) \leq H_3, \tag{21b}$$

holds $\forall(\epsilon_i \cdot t, x_{j|\epsilon_i}, u_{j|\epsilon_i}) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}$, each $j \in \{\mathsf{d}, i\}$ and each $i \in \mathbb{N}_0$; $\|\cdot\|_M$ is an appropriate induced matrix norm [3]. □

The final sufficient assumption that enables convergence of (16) under the control law (18) is stated next.

Assumption 5. Consider (11b) and (13b). There exists $\kappa \in \mathbb{R}$ such that,

$$\gamma := \max_{H_2 \leq \frac{\partial h(\chi)}{\partial u_{j|\epsilon_j}} \leq H_3} \left\| 1 - \kappa \cdot \frac{\partial h(\chi)}{\partial u_{j|\epsilon_j}} \right\|_1 < 1, \tag{22}$$

holds for any $\chi = (\epsilon_j \cdot t, x_{j|\epsilon_i}, u_{j|\epsilon_i}) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}$ each $j \in \{\mathsf{d}, i\}$ and each $i \in \mathbb{N}_0$. □

Finally, the main result is stated below.

Theorem 1. (Convergence of $\Delta y_{i|\epsilon_i}$). Consider the dynamical model (11), the desired behavior model (13) and the corresponding ILC control law (18). Let the Assumptions 1-4 be satisfied and let $\kappa \in \mathbb{R}$ from (18) be such that Assumption 5 is satisfied. Then,

$$\left\|\Delta y_{i+1|\epsilon_{i+1}}\right\|_\lambda < \gamma \cdot \left\|\sigma_{\epsilon_{i+1}}\left(\Delta y_{i|\epsilon_i}\right)\right\|_\lambda, \tag{23}$$

holds. □

Proof: The proof follows similar steps as the proof of Theorem 2.2 in (Xu and Tan, 2003, Chapter 2) and thus is omitted. ■

[3] For definitions and corresponding theory, consult (Rudin, 1976, Chapter 9).

4. SIMULATIONS

To illustrate the convergence property of spatially repeatable tasks under the proposed ILC control law (18) two examples are utilized. One includes a linear time-invariant dynamic system while the other includes nonlinear dynamic control system, cf. (Xu and Tan, 2003, Chapter 2). Notice that both examples involve scalar systems because of simplicity. For both examples, first the corresponding dynamics is stated, followed with the desired behavior dynamics. Then, four different rate ϵ profiles, that satisfy Assumption 1, are depicted and the corresponding results are presented.

Example 2. (Linear system). Consider the following dynamical system,

$$\begin{aligned} \dot{x}_{i|\epsilon_i} &= \epsilon_i \cdot \left(-3 \cdot x_{i|\epsilon_i} - u_{i|\epsilon_i}\right), \\ y_{i|\epsilon_i} &= x_{i|\epsilon_i} - 2 \cdot u_{i|\epsilon_i}, \end{aligned}$$

where $x^{\circ} = 0.5$. Let the desired behavior be,

$$y_{\mathsf{d}|\epsilon_i} = \pi \cdot \sin(\pi \cdot \epsilon_i \cdot t) + \frac{1}{2 - \epsilon_i \cdot t},\ t \in [0, 1.5/\epsilon_i].$$

Now, consider rate ϵ profiles depicted in the Fig. 2. For instance, these shapes may capture rates of 4 different patients or one patient at 4 different sessions. Scenario where such information is not available will be considered in the future work. In particular, the corresponding rate estimates and the corresponding dynamical models will be considered. At this stage, the aim is to demonstrate that with rate profiles which converge to some value, the corresponding tracking error convergence is obtained using the proposed ILC control law (18). Now, the corresponding

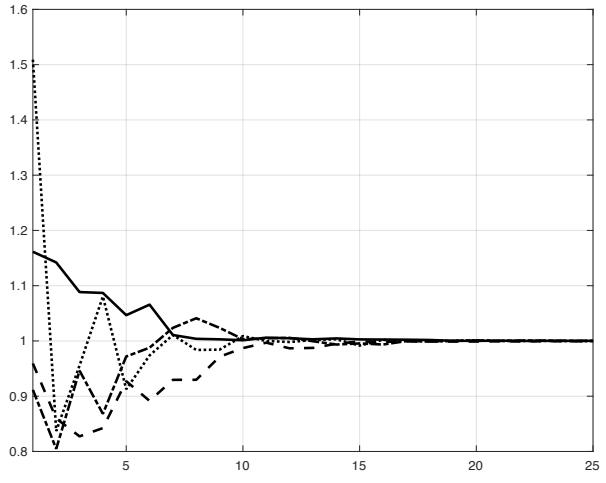

Fig. 2. Different line shapes correspond to different rate ϵ profiles. All four profiles converge to $\epsilon_* = 1$. Iteration time i is located on the horizontal axis while the vertical axis stores values.

results under the proposed ILC control law (18) are captured in the Fig. 3. As captured in Fig. 3, in all instances the predicted convergence is achieved. △

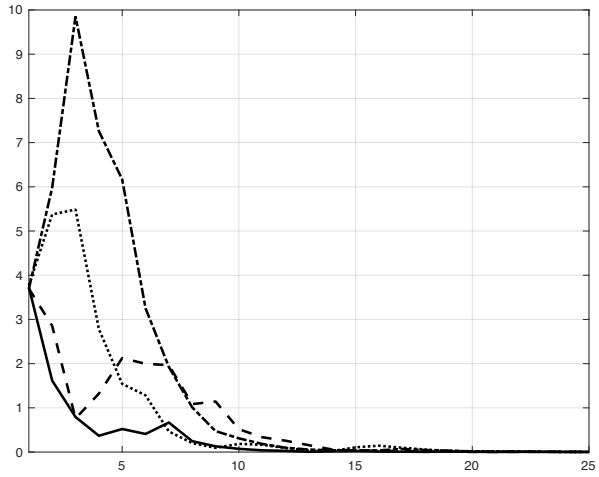

Fig. 3. Convergence of $\Delta y_{i|\epsilon_i}$ under (18) for rate profiles depicted in Fig. 2. The learning control gain in (18), $\kappa = 0.8$, while $u_{0|\epsilon_1} = 0$, for simplicity. Iteration i is located on the horizontal axis while the vertical axis stores the values of $\|\Delta y_{i|\epsilon_i}\|_{\mathsf{s}}$.

Example 3. (Nonlinear system). Similarly as above, consider the following system,

$$\dot{x}_{i|\epsilon_i} = \epsilon_i \cdot \left(0.5 \cdot \cos^2(x_{i|\epsilon_i}) + u_{i|\epsilon_i}\right),$$
$$y_{i|\epsilon_i} = 1.8 \cdot \frac{x^2_{i|\epsilon_i} + 0.1}{1 + x^2_{i|\epsilon_i}} \cdot \arctan\left(\frac{1}{2} \cdot \tan\left(\frac{u_{i|\epsilon_i}}{2}\right)\right),$$

where $x^{\circ} = 0.5$, and let the desired behavior be,

$$y_{\mathsf{d}|\epsilon_i} = (\epsilon_i \cdot t)^2, \ t \in [0, 1.5/\epsilon_i].$$

The considered rate ϵ profiles are depicted in the Fig. 4, while the corresponding results are captured in the Fig. 5. Similarly as for the previous case, Fig. 5 illustrates the predicted convergence. △

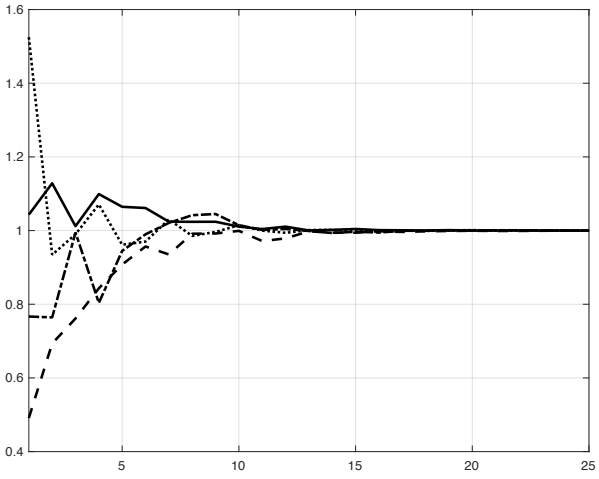

Fig. 4. Different line shapes correspond to different rate ϵ profiles. All four profiles converge to $\epsilon_* = 1$. Iteration time i is located on the horizontal axis while the vertical axis stores values.

Remark 10. (Simulation Implementation). Before proceeding to the next section, few remarks regarding the implementation of the control law (18) are provided. To implement simulations, MATLAB and Simulink are used. Due

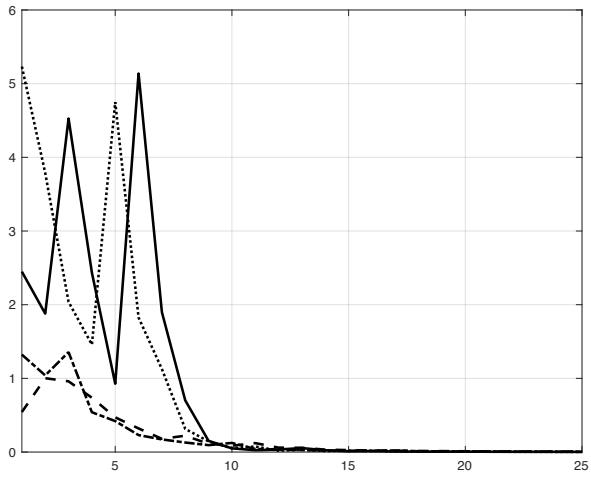

Fig. 5. Convergence of $\Delta y_{i|\epsilon_i}$ under (18) for rate profiles depicted in Fig. 4. The learning control gain in (18), $\kappa = 0.8$, while $u_{0|\epsilon_1} = 0$, for simplicity. Iteration i is located on the horizontal axis while the vertical axis stores the values of $\|\Delta y_{i|\epsilon_i}\|_{\mathsf{s}}$.

to *spatial projection*, implementation of the proposed ILC control law (18) involves first sampling the relevant data from the previous iteration; in particular, 1500 samples are taken. Then, these values are equally spread along the time interval that corresponds to the next iteration, e.g., $i+1$. Depending on the solver used in the Simulink, different interpolation methods are used. For Example 2, for its dash-dot rate profile, Fig. 6 shows the control related data for $i = 4$ (blue color) and its projection for $i = 5$ (red color); which forms the control value for that iteration. In addition, there is also control related data for $i = 50$ (black color) which is the same as its projection to $i = 51$; because the rate profile has converged to the value of one. □

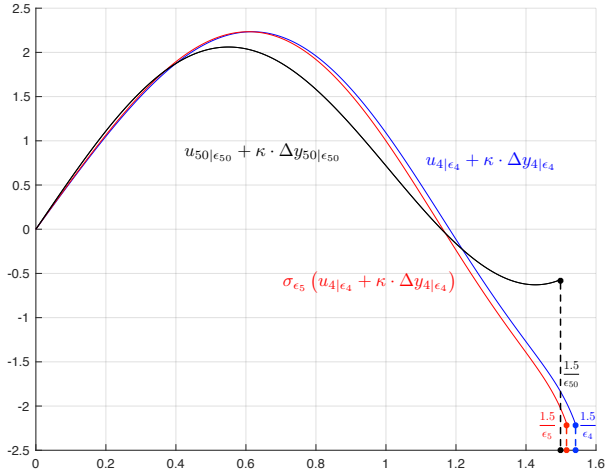

Fig. 6. Control signal $u_{i|\epsilon_i}$ for Example 2 for its dash-dot rate profile. The vertical axis stores control values while the horizontal axis stores the ("universal") time.

5. CONCLUSION

A SILC framework that utilizes the concept of the spatial projection is proposed. A class of nonlinear time-varying systems with relative degree zero is considered and it is shown that the proposed SILC controller achieves spatial tracking. Simulation results demonstrate that the corresponding tracking errors converge even under the imperfections introduced with the simulation environment. There are several technical future directions such as relaxing Assumption 3 to Locally Lipschitz Continuous functions, considering multiple inputs and considering systems with higher relative degree. Most importantly, the future work includes analysis and design where the full knowledge of the rate profile ϵ is not available.

REFERENCES

Ahn, H.S., Chen, Y., and Moore, K.L. (2007). Iterative learning control: brief survey and categorization. *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, 37(6), 1099.

Bien, Z. and Xu, J.X. (2012). *Iterative learning control: analysis, design, integration and applications*. Springer Science & Business Media.

Hoelzle, D.J. and Barton, K.L. (2014). A new spatial iterative learning control approach for improved micro-additive manufacturing. In *2014 American Control Conference*, 1805–1810. IEEE.

Janssens, P., Van Loock, W., Pipeleers, G., Debrouwere, F., and Swevers, J. (2013). Iterative learning control for optimal path following problems. In *52nd IEEE Conference on Decision and Control*, 6670–6675. IEEE.

Kawamura, S. and Sakagami, N. (2002). Analysis on dynamics of underwater robot manipulators based on iterative learning control and time-scale transformation. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, 1088–1094. IEEE.

Lješnjanin, Merid and Tan, Ying and Oetomo, Denny and Freeman, T., Christopher (2016). Spatial iterative learning control: Systems with input saturation. Submitted to *American Control Conference*, Seattle, WA, USA.

Moore, K.L. (2012). *Iterative learning control for deterministic systems*. Springer Science.

Moore, K.L., Ghosh, M., and Chen, Y.Q. (2007). Spatial-based iterative learning control for motion control applications. *Meccanica*, 42(2), 167–175.

Rudin, W. (1976). *Principles of mathematical analysis*, volume 3. McGraw-Hill New York.

Sahoo, S., Panda, S., and Xu, J. (2007). Application of spatial iterative learning control for direct torque control of switched reluctance motor drive. In *Power Engineering Society General Meeting, 2007. IEEE*, 1–7. IEEE.

Shmuelof, L., Krakauer, J.W., and Mazzoni, P. (2012). How is a motor skill learned? Change and invariance at the levels of task success and trajectory control. *Journal of neurophysiology*, 108(2), 578–594.

Wang, Y., Gao, F., and Doyle, F.J. (2009). Survey on iterative learning control, repetitive control, and run-to-run control. *Journal of Process Control*, 19(10), 1589–1600.

Xu, J.X. (2011). A survey on iterative learning control for nonlinear systems. *International Journal of Control*, 84(7), 1275–1294.

Xu, J.X. and Tan, Y. (2003). *Linear and nonlinear iterative learning control*, volume 291. Springer.

Yang, Y.H. and Chen, C.L. (2009). Spatial-based adaptive iterative learning control of nonlinear rotary systems with spatially periodic parametric variation. In *Asian Control Conference, 2009. ASCC 2009. 7th*, 698–703. IEEE.