Michael Meindl\textsuperscript{1}, Simon Bachhuber\textsuperscript{2}, and Thomas Seel\textsuperscript{1}

\textsuperscript{1}Institute of Mechatronic Systems, Leibniz Universität Hannover
\textsuperscript{2}Affiliation not available

June 07, 2024
Iterative Model Learning and Dual Iterative Learning Control: A Unified Framework for Data-Driven Iterative Learning Control

Michael Meindl∗, Simon Bachhuber, and Thomas Seel

Abstract—Accurate reference tracking is essential in control tasks in a variety of applications, and, in repetitive systems, model-based Iterative Learning Control (ILC) is a standard solution that is underpinned by formal stability and convergence guarantees. To overcome the limitations of model-based ILC design, data-driven ILC methods have been introduced which lead to a patchwork of several, very different approaches that do not preserve the modularity and theoretical guarantees of the many, well-established model-based ILC methods. We, for the first time, propose a unifying framework for data-driven ILC that preserves the modularity and formally proven properties of model-based ILC. Specifically, we propose Iterative Model Learning (IML) and Dual Iterative Learning Control (DILC). The IML framework enables iterative learning of unknown dynamics in repetitive systems using input/output trajectory pairs. We formally prove the duality between IML and ILC, i.e., an IML system is equivalent to an ILC system with a trial-varying reference and trial-varying but known dynamics. We further provide generic conditions to guarantee the convergence of the model and prove the duality of monotonic convergence in ILC and IML. This duality allows arbitrary, established, model-based ILC methods to be effectively applied within the IML framework to learn models of unknown dynamics.

Unlike existing data-driven ILC approaches which only combine specific model learning schemes with specific model-based ILC approaches, the proposed DILC framework combines IML with model-based ILC such that established, model-based ILC methods can be arbitrarily combined for simultaneous model and control learning with a formally proven separation principle. The proposed DILC is a unified framework for data-driven ILC that not only relieves model-based ILC of a priori model requirement but also preserves the modularity and theoretical guarantees of model-based ILC. Both IML and DILC are validated by extensive simulations and real-world experiments.

Index Terms—autonomous systems, iterative learning control, reference tracking, monotonic convergence

I. INTRODUCTION

Iterative Learning Control (ILC) is a learning control method that solves reference tracking tasks in repetitive systems by iteratively updating a feedforward control input based on the error information of previous trials [1]. ILC has greatly impacted the field of control because it not only reduces manual tuning efforts by learning autonomously [2], but also because it is grounded on thorough experimental validation and a rich theoretical framework with guarantees for stability and convergence.

To solve ILC problems, a variety of methods have been proposed [3]–[5] which make model assumptions in order to design purpose-built learning control laws. Examples include model-based tuning of scalar learning gains in P-ILC [6], gradient-ILC (G-ILC) [7], [8], norm-optimal ILC (NO-ILC) [9], [10], or $\mathcal{H}_\infty$-ILC [11], [12]. All of these methods are embedded in a theoretical framework that provides conditions for desirable properties such as asymptotic stability, monotonic error convergence, and robustness. Furthermore, model-based ILC methods have been successfully applied to a variety of real-world problems with applications ranging from biomedical engineering to industrial robotics [13]–[16].

However, model-based ILC methods are limited by trade-offs that originate from model uncertainties [17] and by manual modeling efforts that compromise the goal of autonomous control [18]. By learning the required model from data, data-driven ILC (DD-ILC) methods attempt to alleviate these limitations and can be categorized into three groups. First, there are DD-ILC methods [19]–[21] in which a second trial is performed on each iteration to estimate the gradient of the tracking errors with respect to the input trajectory and update the latter. Second, there are DD-ILC methods that identify a model of the plant based on an initial set of input/output trajectory pairs and use the model in a norm-optimal ILC design step [22]–[24]. Third, there are DD-ILC methods [25]–[29] that iteratively estimate the dynamic linearization of the plant dynamics and also employ the learned model in a norm-optimal ILC update. While these DD-ILC methods are capable of learning without a priori model information, there is a common pattern that spans across them all: A DD-ILC method is the combination of a specific model learning scheme and a specific model-based ILC method. As a consequence, existing DD-ILC methods do not support the modular employment of
arbitrary model-based ILC methods, i.e., the model-based ILC method can not be freely chosen to fit the demands of a given application. Furthermore, existing DD-ILC methods require novel derivations of guarantees for properties such as stability or monotonic convergence because they do not preserve the formally proven characteristics of established, model-based ILC methods.

To address these issues, we propose the two learning frameworks called Iterative Model Learning (IML) and Dual Iterative Learning Control (DILC). The first learning framework, namely IML, enables the iterative learning of unknown dynamics from input/output trajectory pairs. We formally prove duality of IML and ILC, i.e., an IML system is equivalent to an ILC system with trial-varying reference and trial-varying but known dynamics. As a consequence, established, model-based ILC methods can be modularly employed in the IML framework to learn models of unknown dynamics. We formally derive generic conditions to guarantee monotonic convergence of the model learning and also prove that monotonic convergence in IML and ILC is dual, i.e., IML preserves the monotonic convergence property of model-based ILC methods.

The second learning framework, namely DILC, combines IML with model-based ILC to relieve the latter of its a priori model requirements. The DILC framework enables the modular combination of arbitrary, model-based ILC methods in both the model and control learning, and a separation principle of the model and control learning is formally proven, i.e., DILC preserves the monotonic convergence property of model-based ILC methods. Hence, DILC is a unified framework for data-driven ILC that preserves both the modular flexibility and formally proven characteristics of model-based ILC.

Both the IML and DILC frameworks are validated by simulations of 25 randomly generated systems with unknown, linear dynamics while employing different, model-based ILC design approaches. Furthermore, the DILC framework is also validated by real-world experiments on a pendulum system with unknown, nonlinear dynamics.

A. Notation

The set of real numbers is denoted by $\mathbb{R}$, the set of positive natural numbers is denoted by $\mathbb{N}_{\geq 0}$, and the set of natural numbers including zero is denoted by $\mathbb{N}$. Vectors are denoted by lower-case letters in bold, e.g., $\mathbf{v}$, and matrices are denoted by upper-case letters in bold, e.g., $\mathbf{A}$. The Euclidean norm of a vector $\mathbf{v}$ is denoted by $||\mathbf{v}||$, and the induced Euclidean norm of a matrix $\mathbf{A}$ is denoted by $||\mathbf{A}||$.

II. PRELIMINARIES

Consider a repetitive, single-input/single-output, discrete-time system that, on trial $j \in \mathbb{N}_{\geq 0}$ and sample $n \in \mathbb{N}_{\geq 0}$, has the input variable $u_j(n) \in \mathbb{R}$ and output variable $y_j(n) \in \mathbb{R}$. Each trial consists of $N \in \mathbb{N}_{\geq 0}$ samples, and the samples of the input, respectively output, variable are collected in the so called input, respectively output trajectory, i.e., $\forall j \in \mathbb{N}_{\geq 0}$,

$$\mathbf{u}_j := [u_j(1) \quad u_j(2) \quad \ldots \quad u_j(N)]^T \quad (1)$$

$\mathbf{y}_j := [y_j(1 + m) \quad y_j(2 + m) \quad y_j(N + m)]^T$, (2)

whereby $m \in \mathbb{N}_{\geq 0}$ is the system’s relative degree. The system’s linear, time-invariant dynamics can be written in the lifted form

$$\forall j \in \mathbb{N}_{\geq 0}, \quad \mathbf{y}_j = \mathbf{P} \mathbf{u}_j \quad (3)$$

where $\mathbf{P} \in \mathbb{R}^{N \times N}$ is the plant matrix of the form

$$\mathbf{P} = \begin{bmatrix} p_1 & 0 & \ldots & 0 \\ p_2 & p_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ p_N & p_{N-1} & \ldots & p_1 \end{bmatrix} \quad (4)$$

The values, $\forall n \in [1, N]$, $p_n \in \mathbb{R}$ are the Markov parameters and can, amongst others, be obtained from a state-space model or transfer function [3].

The control task consists in having the output trajectory track a desired reference $\mathbf{r} \in \mathbb{R}^N$, and the error trajectory is defined as

$$\forall j \in \mathbb{N}_{\geq 0}, \quad \mathbf{e}_j := \mathbf{r} - \mathbf{y}_j \quad (5)$$

The learning task consists in updating the input trajectory from trial to trial such that the error trajectory is minimized. To this end, we consider the linear update law of form

$$\forall j \in \mathbb{N}_{\geq 0}, \quad \mathbf{u}_{j+1} = \mathbf{u}_j + L_{j} \mathbf{e}_j \quad (6)$$

whereby $L_j \in \mathbb{R}^{N \times N}$ is a possibly trial-varying - learning gain matrix. In model-based ILC, we assume that a possibly trial-varying - model $\mathbf{M}_j$ of the plant matrix $\mathbf{P}$ is available. Based on this model and additional parameters, the learning-gain matrix is designed. Formally, let $\mathcal{D} : \mathbb{R}^{N \times N} \mapsto \mathbb{R}^{N \times N}$ be a design function such that

$$\forall j \in \mathbb{N}_{\geq 0}, \quad L_j = \mathcal{D}(\mathbf{M}_j) \quad (7)$$

For the remainder of the manuscript, it is assumed that all design functions $\mathcal{D}$ are continuous, i.e., for every $\epsilon > 0$, there exists $\delta > 0$ such that

$$||\mathbf{X} - \mathbf{Y}|| < \epsilon \quad \implies \quad ||\mathcal{D}(\mathbf{X}) - \mathcal{D}(\mathbf{Y})|| < \delta. \quad (8)$$

Definition 1 (ILC System): The combination of dynamics (3), error trajectory (5), update law (6), and design function (7) is called an ILC system.

Given an ILC system, a core question is whether the tracking error converges. This motivates the concept of monotonic tracking convergence.

Definition 2 (Monotonic Tracking Convergence): An ILC system is called monotonically tracking convergent if and only if there exists $\alpha \in [0, 1)$ such that

$$\forall j \in \mathbb{N}_{\geq 0}, \quad ||\mathbf{e}_{j+1}|| \leq \alpha ||\mathbf{e}_j|| \quad (9)$$

In order to guarantee monotonic convergence, a variety of model-based design functions have been proposed. Two prominent examples are given by so-called gradient ILC and norm-optimal ILC.

Definition 3 (Gradient ILC): An ILC system with design function

$$\mathcal{D}_G(\mathbf{M}_j, \beta) = \beta \mathbf{M}_j^T \quad (10)$$

is called Gradient ILC [30].
Definition 4 (Norm-Optimal ILC): An ILC system with design function

\[ D_{\text{NO}}(M_j, Q, S_j) = (M_j^T Q M_j + S_j)^{-1} M_j^T Q , \]

whereby \( Q \in \mathbb{R}^{N \times N} \) and \( S_j \in \mathbb{R}^{N \times N} \) are symmetric, positive definite matrices, is called norm-optimal ILC \[31\].

III. PROBLEM FORMULATION

In this paper, the tightly coupled problems of learning a model of the dynamics (3) and learning an input trajectory to track a reference trajectory despite the dynamics being unknown are considered. For the remainder of the paper, the linear, trial-invariant dynamics (3) are considered and it is assumed that the plant matrix \( P \) is unknown.

A. Model Learning Problem

The first problem, called the model learning problem, consists in iteratively updating a model matrix \( M_j \) based on a sequence of input/output trajectories \( \{u_j, y_j\} \) that stem from the dynamics (3) such that, for some given input trajectory \( u \), the model matrix predicts the output trajectory according to

\[ \forall j \in \mathbb{N}_{\geq 0}, \quad \hat{y} = M_j u . \]

The model matrix \( M_j \) must converge, ideally in a monotonic fashion, to the unknown plant matrix \( P \). Convergence is judged based on the input-dependent prediction error

\[ \forall j \in \mathbb{N}_{\geq 0}, \quad \hat{e}_j(u) := y - M_j u, \]

whose norm must converge to zero.

To this end, all input trajectories are assumed to satisfy the excitation condition

\[ \forall j \in \mathbb{N}_{\geq 0}, \quad u_j(1) \neq 0 , \]

which ensures that all entries of the plant matrix \( P \) affect the output trajectory and, hence, are possible to be estimated.

B. Control Learning Problem

The second problem, called the control learning problem, consists in iteratively updating the input trajectory \( u_j \) such that the output trajectory \( y_j \) tracks the reference trajectory \( r \) despite the plant matrix \( P \) being unknown. Tracking performance is judged based on the error trajectory \( e_j \) whose norm must, ideally in a monotonic fashion, converge to zero.

IV. ITERATIVE MODEL LEARNING

To solve the model learning problem, we introduce the framework of Iterative Model Learning (IML) which is built on two major components. First, we introduce a lifting operator that allows us to prove duality, i.e., the model learning problem can be transformed to be equivalent to the control learning problem but with known dynamics. Second, we employ tools of ILC to analyze and solve the model learning problem.

A. Lifting Operator & Problem Transformation

To prove duality of model and control learning, we first introduce a lifting operator. For this purpose, let \( T_N \) denote the space of all lower triangular, toeplitz matrices of dimensions \( N \times N \), i.e.,

\[ \forall T \in T_N, \quad T = \begin{bmatrix} t_1 & 0 & \ldots & 0 \\ t_2 & t_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ t_N & t_{N-1} & \ldots & t_1 \end{bmatrix} \]

with \( \forall n \in [1, N], t_n \in \mathbb{R} \). Now, for any given vector \( v \in \mathbb{R}^N \)

\[ v = [v_1 \ v_2 \ldots \ v_N]^T \]

let \( L : \mathbb{R}^N \mapsto T_N \) denote a lifting operator defined by

\[ L(v) = \begin{bmatrix} v_1 & 0 & \ldots & 0 \\ v_2 & v_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ v_N & v_{N-1} & \ldots & v_1 \end{bmatrix} \]

Note that the inverse of the lifting operator \( L^{-1} : T_N \mapsto \mathbb{R}^N \) exists. The lifting operator and its inverse motivate the following result regarding the multiplication of a lower-triangular, toeplitz matrix, and a vector.

Lemma 1: Given a matrix \( T \in T_N \) and a vector \( v \in \mathbb{R}^N \),

\[ Tv = L(v)L^{-1}(T) . \]

Proof: Let \( w = Tv \),

\[ w_n = \sum_{i=1}^{n} t_{n+1-i} v_i = \sum_{i=1}^{n} v_{n+1-i} t_i \]

and, hence,

\[ Tv = w = L(v)L^{-1}(T) . \]

Lemma 1 enables the transformation of the model learning problem, which leads to the formal definition of an Iterative Model Learning System, see Figure 1. For this purpose, let \( P := L^{-1}(P) \) denote the plant vector, let \( U_j := L(u_j) \) denote the input matrix, and let \( m_j := L^{-1}(M_j) \) denote the model vector.

Applying Lemma 1 to (12) yields the model equation

\[ \hat{y}_j = U_j m_j . \]

Combining the latter with (13) yields the prediction error

\[ \hat{e}(u_j) = y_j - U_j m_j . \]

And introducing the model update law

\[ \forall j \in \mathbb{N}_{\geq 0}, \quad m_{j+1} = m_j + \hat{L}_j e_j(u_j) , \]

where \( \hat{L}_j \in \mathbb{R}^{N \times N} \) is the trial-varying model learning matrix that stems from the model design function \( D : \mathbb{R}^{N \times N} \mapsto \mathbb{R}^{N \times N} \) such that

\[ \hat{L}_j = D(U_j) . \]
leads to the formal definition of an IML system.

Definition 5 (Iterative Model Learning System): The combination of model dynamics (22), prediction error (23), model update law (24) and design function (25) is called an Iterative Model Learning System.

The core property of an IML system is that it is dual to an ILC system as specified by the following theorem.

Theorem 1 (Duality of ILC and IML): An IML system is equivalent to an ILC system with trial-varying reference and trial-varying, but known dynamics.

Proof: The proof follows from comparing the definitions of an ILC and IML system, as summarized by Table I. The dynamics of an ILC system are given by (3) and the dynamics of an IML system are given by (22), where the predicted output trajectory \( \hat{y}_j \) is the IML’s equivalent to the ILC’s trajectory \( y_j \), the input matrix \( U_j \) is the IML’s trial-variant, known equivalent to the ILC’s plant matrix \( P \), and the model vector \( m_j \) is the IML’s equivalent to an ILC’s input trajectory \( u_j \).

The error of an ILC system is given by (5) and the error of an IML system is given by (23), where the predicted error \( \hat{e}_j \) is the IML’s equivalent to the ILC’s tracking error \( e_j \), and the output trajectory \( y_j \) is the IML’s trial-varying equivalent to the ILC’s reference trajectory \( r \). The update law of an ILC system is given by (6) and the update law of an IML system is given by (24), where the trial-varying model learning matrix \( L_j \) is the ILM’s equivalent to the ILC’s learning gain matrix \( L \). The design function of an ILC system is given by (7) and the design function is given by (25), which are of identical form and, in fact, interchangeable.

Given duality of ILC and IML, i.e., an IML system is equivalent to an ILC system with trial-varying reference and trial-varying, but known dynamics, we are interested in deriving formal conditions to guarantee convergence of the model learning. A quantity of particular interest is the difference between the unknown plant vector \( p \) and the model vector \( m_j \), which motivates the concept of monotonic model convergence.

Definition 6 (Monotonic Model Convergence): An IML system is called monotonically model convergent if and only if there exists an \( \alpha \in (0, 1) \) such that

\[
\forall j \in \mathbb{N}_{\geq 0}, \quad ||p - m_{j+1}|| \leq \alpha ||p - m_j|| .
\]  (26)

Theorem 2 (Monotonic Model Convergence): An IML system is monotonically model convergent if and only if

\[
\forall j \in \mathbb{N}_{\geq 0}, \quad ||\hat{L}_j U_j|| < 1 .
\]  (27)

Proof: Combining (22)-(24) leads to, \( \forall j \in \mathbb{N}_{\geq 0}, \)

\[
p - m_{j+1} = (I - \hat{L}_j U_j) (p - m_j) ,
\]  (28)

which means that the model error is determined by linear, trial-varying dynamics. Hence, monotonic model convergence requires the linear, trial-varying operator \( I - \hat{L}_j U_j \) to be a contraction, which is the case if and only if

\[
\forall j \in \mathbb{N}_{\geq 0}, \quad ||I - \hat{L}_j U_j|| < 1 .
\]  (29)

A second quantity of interest is the prediction error \( \hat{e} \), which motivates the concept of monotonic prediction convergence.

Definition 7 (Monotonic Prediction Convergence): An IML system is called monotonically prediction convergent if and only if there exists a \( \beta \in (0, 1) \) such that

\[
\forall j \in \mathbb{N}_{\geq 0}, \quad ||\hat{e}_{j+1}|| \leq \beta ||\hat{e}_j|| .
\]  (30)

Theorem 3 (Monotonic Prediction Convergence): An IML system is monotonically prediction convergent if and only if

\[
\forall j \in \mathbb{N}_{\geq 0}, \quad ||I - U_j \hat{L}_j|| < 1 .
\]  (31)

Proof: Combining (22)-(24) yields

\[
\hat{e}_{j+1} = (I - U_j \hat{L}_j) \hat{e}_j (u_j) ,
\]  (32)

which means that the prediction error also follows linear, trial-varying dynamics which relate to the model dynamics (28) via a trial-varying similarity transformation using the model matrix \( U_j \). Hence, monotonic prediction convergence requires
the linear, trial-varying operator $I - U_j \tilde{L}_j$ to be a contraction, which is the case if and only if
\[ \forall j, \quad \| I - U_j \tilde{L}_j \| < 1. \quad (33) \]

According to Theorems 2 and 3, monotonic convergence of the model and prediction errors requires trial-varying conditions consisting of the input matrix $U_j$ and learning gain matrix $L_j$ to be satisfied. To find design functions $\tilde{D}$ that compute $\tilde{L}_j$ depending on $U_j$ and meet said conditions, duality of IML and ILC can be exploited. In fact, any model-based ILC design function can be applied in an IML system to design $L_j$ based on $U_j$, and the design function leads to monotonic prediction convergence of the IML system if the design function leads to monotonic tracking convergence in an ILC system as stated by the following theorem.

**Theorem 4 (Duality of Convergence):** Given a design function $\tilde{D}$, an IML system which employs $\tilde{D}$ as design function is monotonically prediction convergent if an ILC system with design function $\tilde{D}$ is monotonically tracking convergent.

**Proof:** Analogous to the proof of Theorem 3 the definition of an ILC system yields
\[ e_{j+1} = (I - PL) e_j, \quad (34) \]
which implies that monotonic tracking convergence is equivalent to
\[ \| I - PL \| < 1. \quad (35) \]
Hence, because the design function $\tilde{D}$ leads to monotonic tracking convergence, the following holds:
\[ L = \tilde{D}(P) \iff \| I - PL \| < 1 \quad (36) \]
Applying $\tilde{D}$ to an IML system and combining the latter with the duality result of Theorem 1 yields
\[ \tilde{L}_j = \tilde{D}(U_j) \iff \| I - \tilde{L}_j U_j \| < 1, \quad (37) \]
which according to Theorem 3 guarantees monotonic prediction convergence of the IML system. □

Based on this result, any model-based ILC design function that is monotonically tracking convergent leads to monotonic prediction convergence of an IML system. In this work, we consider, without loss of generality, the two examples of Gradient-ILC (G-ILC) and Norm-Optimal ILC (NO-ILC) being applied to IML. If applied to IML, G-ILC becomes Gradient-IML (G-IML), where the design function is given by
\[ \tilde{L}_j = \tilde{D}_G \left( U_j, \beta_j \right) = \beta_j U_j^T. \quad (38) \]
And NO-ILC becomes Norm-Optimal-IML (NO-IML), where the design function is given by
\[ \forall j \in \mathbb{N}_{\geq 0}, \quad \tilde{L}_j = \tilde{D}_{NO}(U_j, R, S) \]
\[ = (U_j^T RU_j + S)^{-1} U_j^T R. \quad (39) \]
Note that for both G-ILC and NO-ILC conditions for monotonic conditions have been derived [30], [31], which, according to Theorem 4 also guarantee monotonic prediction convergence of their respective IML counterparts.

**V. Dual Iterative Learning Control**

In this section, we consider the control learning problem that consists in learning a feedforward input trajectory to track the desired reference despite no a priori model of the plant dynamics being known. To solve this problem, we propose the framework of so called Dual Iterative Learning Control, in which the concepts of Iterative Model Learning (IML) and model-based ILC are combined.

Each iteration of the DILC scheme consists of three steps, see Figure 2. First, the current trial input trajectory is applied to the plant yielding an input/output trajectory pair. Second, there is the so called model learning step, in which the input/output trajectory pair is employed to update a model of the unknown plant dynamics using the IML framework. Third, there is the so called control learning step, in which the updated model is used to perform a model-based ILC update to learn the desired input trajectory. The key strength of the DILC framework is that model-based ILC design methods can be employed in the model learning and the control learning step. Hence, the DILC framework enables the application of well-established, model-based ILC methods in settings where a model of the dynamics is not a priori known.

In the proposed DILC framework, two design functions are employed. One to iteratively compute the IML’s learning gain matrix $\tilde{L}_j$ based on the input matrix $U_j$, and one to iteratively compute the ILC’s learning gain matrix $L_j$ based on the model $M_j$. Here, a major question is which sort of properties the design functions have to possess such that the model and the control learning converge monotonically. This question is addressed by the following theorem, which proves separability, i.e., if the design functions lead to monotonic model convergence in an IML setting and monotonic tracking convergence in an ILC setting, both of these properties are also going to be guaranteed if the design functions are combined in a DILC setting.

**Theorem 5 (Separation Principle):** A DILC system with model learning design function $\tilde{D}$ and control learning design function $D$ is monotonically model convergent and there exists a threshold trial $J \in \mathbb{N}_{\geq 0}$ from which onwards the control learning is monotonically tracking convergent, i.e., there exists $\alpha \in [0, 1)$ such that
\[ \forall j \geq J, \quad ||e_{j+1}|| \leq \alpha ||e_j|| \quad (40) \]
if the design function $\tilde{D}$ is continuous and leads to monotonic model convergence in an IML system and $D$ is continuous and leads to monotonic tracking convergence in an ILC system.

**Proof:** See Appendix A.

**VI. Validation by Simulation**

In this section, the two proposed frameworks of IML and DILC are investigated in simulation.

**A. Iterative Model Learning**

In this section, the proposed IML framework’s capability of learning a model for arbitrary, linear dynamics using different design methods is investigated. For this purpose, the IML framework is applied to 25 linear, asymptotically stable, 3rd
Fig. 2. Overview of the proposed Dual Iterative Learning Control (DILC) scheme which combines IML and model-based ILC to relieve the latter of a priori model requirements while also preserving the modularity and theoretical guarantees of model-based ILC. Each of the DILC’s iterations consists of three steps. First, the current input trajectory is applied to the plant with unknown dynamics. Second, the resulting input/output trajectory pair is used in an IML iteration to update the model of the unknown plant dynamics. Third, the updated model is employed in a model-based ILC iteration to update the input trajectory.

Fig. 3. Simulation of the proposed IML scheme using Gradient IML (G-IML) and Norm-Optimal IML (NO-IML) design approaches for 25 randomly generated linear systems. For both design methods and each of the 25 systems, the proposed IML framework achieves monotonic convergence of the model error and the prediction error.

sequences \( \{u_j, y_j\} \) have been produced by drawing 25 input trajectories from a zero-mean normal distribution with standard deviation \( \sigma = 1 \) and applying the input trajectories to the plant dynamics. Each trial consists of \( N = 100 \) samples.

To demonstrate that the IML framework is capable of solving the model learning task using different design functions, both G-IML and NO-IML are applied to the benchmark. The G-IML approach uses the self-parametrization of

\[
\forall j \in \mathbb{N}_{\geq 0}, \quad \beta_j = \frac{2}{\sigma^2(U_j)}.
\]

The NO-IML approach uses the self-parametrization of

\[
\forall j \in \mathbb{N}_{\geq 0}, \quad Q_j = I, \quad S_j = ||U_j||^2 I.
\]

Note that it can be shown that these update laws lead to monotonic tracking convergence in an ILC system if a precise model was known, which - by the duality of ILC and IML convergence - implies monotonic prediction convergence of the IML system.

Learning performance is judged based on the normalized model error norm

\[
\bar{\epsilon}_j := \frac{||p - m_j||}{||p||},
\]

and the normalized prediction error norm

\[
\bar{e}_j := \frac{||y^* - M_j u^*||}{||y^*||},
\]

where \( u^* \) is a randomly generated input trajectory and \( y^* \) the corresponding output trajectory which serve as trial-invariant prediction benchmark.

The progression of the normalized model error’s median and maximum depicted in Figure 3 demonstrate that the proposed IML framework rapidly learns a model of the unknown dynamics using either the G-IML or NO-IML design. Independent of the design function, the median of the model error drops below 50% of the initial value within three trials, drops below 10% of the initial value within ten trials, and approaches a value close to zero roughly at the 20th trial. Similarly, the prediction error drops below 20% of the initial value within order systems that have been randomly generated using the MATLAB command \textit{drss}. For each system, the input/output
five trials and less than 10% of the initial value within ten trials. Furthermore, the model and prediction error decline monotonically for both design functions and all trials over all 25 randomly generated systems. In summary, the simulations demonstrate that the proposed DILC framework is capable of employing different model-based ILC design functions to autonomously learn a model of unknown dynamics while also preserving the monotonic convergence property of model-based ILC methods.

B. Dual Iterative Learning Control

This section investigates the proposed DILC framework’s capability to solve a variety of reference tracking tasks in different systems by employing various model-based ILC design techniques despite no a priori model information being available. For this purpose, 25 linear, asymptotically stable, 3rd order systems are randomly generated using the MATLAB command rssd. For each of these systems, a random reference trajectory is generated, by drawing a trajectory from a zero-mean normal distribution with variance of one and applying a zero-phase lowpass filter with a cut-off frequency of 5 Hz to the latter. The initial input trajectory is generated by drawing a trajectory from a zero-mean normal distribution with variance of one.

Second, there is the so called NONO-DILC configuration, which employs the norm-optimal design in the model learning and the control learning step, and uses the self-parametrization of

\[ \hat{Q}_j = Q_j = I, \quad \hat{S}_j = \|U_j\|^2 I, \quad S_j = \|M_j\|^2 I. \]  (46)

And third, there is the so called GNO-DILC configuration, which employs the gradient design in the model learning step and the norm-optimal design in the control learning step and uses the self-parametrization of

\[ \hat{\beta}_j = \frac{2}{\|U_j\|^2}, \quad Q_j = I, \quad S_j = \|M_j\|^2 I. \]  (47)

Learning performance is judged based on the mean and maximum progress of the normalized tracking error norm

\[ \forall j \in \mathbb{N}_{\geq 0}, \quad e_j := \frac{\|r - y_j\|}{\|r\|}. \]  (48)

The results depicted in Figure 4 show that for all three configurations the mean of the normalized tracking error norm rapidly declines in a rapid fashion, reaches 50% of the initial value after just two trials, and drops below 20% of the initial value after roughly ten trials. In the worst case, the maximum error norm increases on the first trial because the learned model deviates too strongly from the actual dynamics. However, from the second trial onward the tracking error norm is strictly monotonically declining. Note that all three configurations achieve comparable performance, i.e., the results demonstrate that the proposed DILC framework can be utilized with various, well-established ILC design methods to learn reference tracking without any a priori model information.

VII. Validation of DILC by Experiment

To demonstrate the practical application of the proposed DILC framework, we consider the problem of learning to swing up an inverted pendulum without any prior model knowledge. The target system consists of the pendulum, which is driven by a motor, whose torque \( u \in \mathbb{R} \) serves as input variable. The pendulum’s angle serves as output variable \( y \in \mathbb{R} \) and is measured by an encoder. The system is controlled by a microcomputer, and the discrete-time system is sampled with a frequency of 50 Hz. The motion task consists in tracking the reference trajectory \( r \in \mathbb{R}^{175} \), which transitions the pendulum from the bottom position, \( y = 0^\circ \), to the upright equilibrium, \( y = 180^\circ \).

To solve the learning problem, the proposed DILC framework is employed, whereby norm-optimal design is employed in both the model learning and the control learning step. On each trial, the weights are determined autonomously according to (46). The initial input trajectory is generated by drawing a trajectory from a zero-mean normal distribution with variance 0.1 and applying a zero-phase lowpass filter with a cut-off frequency of 5 Hz to the latter.

Results depicted in Figure 5 show that the initial input trajectory \( u_0 \) hardly excites the system, and, hence, the corresponding trajectory \( y_0 \) significantly deviates from the desired reference trajectory. However, from here onward the error norm rapidly declines, drops below 30% of its initial
value on the fifth trial, and reaches roughly 10% of its initial value on the 20th trial. Note that the tracking error norm is monotonically decreasing on most trials and only slightly increases on two trials, which is likely due to trial-varying disturbances. Note that the output trajectories significantly deviate from the starting position, and, hence, the pendulum’s nonlinear dynamics clearly violate the assumption of linear dynamics. Nonetheless, the proposed DILC framework solves the learning task without any a priori model knowledge, which indicates that the proposed DILC framework also provides some degree of robustness with respect to nonlinear dynamics.

VIII. DISCUSSION AND CONCLUSION

In this work, we have proposed two learning frameworks, namely Iterative Model Learning (IML) and Dual Iterative Learning Control (DILC). The IML framework solves the problem of iteratively learning a model of unknown dynamics by transforming the model learning problem into an ILC problem with trial-varying reference and trial-varying, but known dynamics. This equivalence, called duality of IML and ILC, is formally proven, and, hence, the IML framework enables the modular employment of established, model-based ILC methods for learning models of unknown dynamics. To guarantee monotonic convergence of the model learning, generic conditions were formally derived and duality of the IML’s and ILC’s convergence was proven, i.e., IML preserves the monotonic convergence property of model-based ILC methods. The proposed IML’s ability to modularly employ established, model-based ILC methods and preserve their monotonic convergence properties was validated by simulations of 25 randomly generated systems. To each of the 25 systems, both gradient ILC and norm-optimal ILC were applied to learn models of the dynamics using the IML framework, and in each of the total of 50 learning tasks rapid monotonic convergence was achieved.

The proposed DILC framework relieves established, model-based ILC approaches of their a priori model requirements by combining IML and model-based ILC. The DILC framework enables the modular combination of established, model-based ILC methods for both the model and the control learning. A separation principle for the model and control learning was formally proven, i.e., the monotonic convergence property of model-based ILC is preserved within the DILC framework if the model learning is also monotonically convergent. The DILC framework was also validated by simulations of 25 randomly generated systems using different combinations of gradient ILC and norm-optimal ILC for the model and control learning, which stresses the framework’s capability of modularly combining established, model-based ILC methods without a priori model information to solve reference tracking tasks. The DILC framework was further validated in real-world experiments on a pendulum with unknown, nonlinear dynamics which indicates the framework’s ability to also deal with nonlinear dynamics to some extent.

Unlike existing data-driven ILC approaches which only combine specific model learning schemes with specific model-based ILC methods and require novel guarantees for stability and convergence, the two frameworks proposed in this work pose a unified approach for data-driven ILC that preserves the modularity and theoretical properties of established, model-based ILC methods. As a consequence, the DILC framework
enables the seamless application of state-of-the-art, model-based ILC in applications where model information is unavailable or difficult to obtain.

**APPENDIX**

**A. Proof of Theorem 5**

First, note that monotonic model convergence of $\hat{D}$ implies monotonic model convergence of the DILC system. Second, to show monotonic tracking convergence, it follows from the definition of a DILC system that

$$e_{j+1} = (I - PL_j) e_j,$$  \hspace{1cm} (49)

where $L_j = D(M_j)$, and, hence, monotonic tracking convergence from some trial $J$ onwards is given if and only if

$$\forall j > J, \ |A(M_j)| < 1,$$  \hspace{1cm} (50)

where for the sake of notational simplicity $A(X) := I - PD(X)$. As shown in the proof of Theorem 4, monotonic tracking convergence of $D$ in an ILC system implies

$$|A(P)| < 1.$$  \hspace{1cm} (51)

Because $D$ is continuous, $A(X)$ also is continuous, and, hence, \eqref{eq:51} implies that there exists some neighbourhood $|P - X| < \delta$ such that $|A(X)| < 1$. Hence, monotonic tracking convergence of the DILC system from some trial $J$ onwards is given if

$$\forall j > J, \ |P - M_j| < \delta.$$  \hspace{1cm} (52)

To show the latter, note that monotonic model convergence implies that there exists some $\alpha \in [0, 1)$ such that

$$|p - m_{j+1}| < \alpha |p - m_j|$$  \hspace{1cm} (53)

and, hence,

$$|p - m_j| \leq \alpha^j |p - m_0|.$$  \hspace{1cm} (54)

Next, consider that the lifting operator $L$ is a map between two finite dimensional vector spaces, namely $\mathbb{R}^N$ and $\mathbb{R}^{N \times N}$, and that $L$ is linear because, for some $a \in \mathbb{R}$, $v$, $w \in \mathbb{R}^N$, $aL(v) = L(av)$ and $L(v) + L(w) = L(v + w)$. Hence, $L$ is bounded, i.e., there exists some $\beta > 0$ such that

$$\forall v \in \mathbb{R}^N, \ |L(v)| \leq \beta ||v||.$$  \hspace{1cm} (55)

Combining the latter with \eqref{eq:54} yields

$$\forall j, \ |P - M_j| \leq |L(p - m_j)| \leq \beta \alpha^j |p - m_0|,$$  \hspace{1cm} (56)

i.e., $|P - M_j|$ is upper bounded by a sequence that monotonically converges to zero. Hence, for any $\delta > 0$ there exists a trial $J \in \mathbb{N}_{\geq 0}$ such that

$$\forall j \geq J, \ |P - M_j| \leq \beta \alpha^j |p - m_0| < \delta.$$  \hspace{1cm} (57)

**REFERENCES**


Michael Meindl received the M.Sc. degree in mechatronic engineering from the University of Applied Sciences Karlsruhe, Germany, in 2020. He is currently pursuing the PhD degree jointly at Leibniz University Hannover and FAU Erlangen-Nürnberg, and his research focuses on motor learning in robotic systems with the prime interest of combining methods from the fields of control theory and machine learning.

Simon Bachhuber received the M.Sc. degree in physics from Universität Regensburg, Regensburg, Germany, in 2020. He is currently pursuing the Ph.D. degree with Friedrich-Alexander-Universität (FAU) Erlangen-Nürnberg, Erlangen, Germany. He completed an eight-month internship at the BMW Group as part of the Predictive Maintenance team until 2021. Since 2021, he has been employed by FAU Erlangen-Nürnberg. His research focuses on developing ML-based solutions for state estimation and control of highly dynamic systems, with a particular emphasis on IMU-based motion tracking.

Thomas Seel is the Director of the Institute of Mechatronic Systems at Leibniz Universität Hannover. He studied Engineering Cybernetics at OVGU Magdeburg and UC Santa Barbara and received the PhD in Control Engineering from TU Berlin in 2016. He has been a full professor at the Department AI in Biomedical Engineering of FAU Erlangen-Nürnberg since 2021 and has been appointed at LUH in 2023. His research interests include sensor fusion, dynamic learning, and sensorimotor AI in robotics and biomedical applications.