2025 23rd European Control Conference (ECC)
June 24-27, 2025. Thessaloniki, Greece

A Biologically Inspired Modular-Based Iterative Learning Control Design with Provable Convergence

Daniel Hobson, Bing Chu and Xiaohao Cai

***Abstract*— For control problems that repeat with resets, such as batch processing and robotic trajectory tracking, iterative learning control is an established high-performance control design method. Fast learning is achievable with model-based schemes when the dynamics of the system are known to be described accurately by a prior model, but such accurate models are often difficult or expensive to obtain. The field of sensorimotor control studies the motion control systems of humans and other animals, which appear to quickly achieve accurate trajectory tracking without detailed prior knowledge. In this paper, we present a novel modular-based design inspired by the learning behaviour of these sensorimotor control systems. The 'modules' used are a generalisation of pre-defined orthonormal basis functions, and the parameters of these modules are learnt using an alternating direction method of multipliers. We analyse the convergence properties of the proposed design rigorously, and also discuss how this approach may be successfully applied when the reference changes over the trials. Generalising learnt skill in this way is a current challenge in iterative learning control design, and is a key benefit of the modular structure of sensorimotor-inspired schemes.**

I. Introduction

Iterative learning control (ILC) is a powerful feedforward control design based on the principle of incrementally improving repeated attempts at a problem [1]–[3]. ILC designs are practical and useful in a wide range of applications [4]–[6], in part because they may be constructed to be model-based or model free [1], [7]. Model-free schemes emphasise the flexibility of learning-based approaches to derive information from measurements of the performance of the system on previous trials, whereas model-based schemes also exploit the available model information to accelerate the learning. In this paper, we seek to achieve this fast learning using a method inspired by the biological control systems responsible for coordinated motion in humans and other animals [8], [9]. These sensorimotor control systems exhibit high performance – in particular, the fast learning of control signals that achieve accurate tracking of a reference trajectory. This high performance is maintained when faced with large disturbances and even generalises to unseen tasks, and the mechanisms underlying these capabilities have been studied. The literature establishes *modularity* as a key property; the central nervous system of vertebrates is separated into functional blocks or 'modules' at the neural level [10] and the response elicited by artificial stimulation of multiple such modules is the *linear superposition* of the individual responses.

This motivates the structure of our proposed ILC algorithm – the ILC controller is parameterised as a linear combination of modules. These modules may be a pre-specified finite set as in [11] or orthogonal basis functions such as polynomials [12], [13] or rational functions [14], [15]. Furthermore, our approach generalises to any set of linearly independent modules, which presents opportunities for the online learning of efficient basis representations in the future, as suggested in [16], [17]. We then iteratively learn the parameters of the controller using a framework similar to that proposed in our earlier work [18] which does not have a convergence guarantee. In this paper, we formulate an optimisation problem with solution corresponding to the desired controller parameters. We solve the optimisation problem with a method based on the alternating direction method of multipliers (ADMM) algorithm, which inherits strong convergence properties under mild assumptions [19].

We also discuss the implications of learning the controller parameters, rather than directly learning the control sequence as is typical for traditional ILC schemes. Since the system model is invariant to reference changes, the learning achieved while tracking one reference *generalises* to achieve accurate tracking on subsequent trials performed with a *different* reference [12], [14], [20]. This is understood in sensorimotor control to be a major benefit of a modular control scheme, and there is interest in studying how the contributions of each module depend on the underlying module behaviour.

II. Problem Formulation

We seek to design a controller for a discrete time single-input-single-output (SISO) linear time invariant (LTI) system, which is taken to be stable and minimum phase. The system operates iteratively, with each iteration termed a trial (denoted by k) and has length N. On each trial, we begin from the initial state x_0 and seek to track the reference $r(t)$, subject to the dynamics described by the state space model

$$x_k(t+1) = Ax_k(t) + Bu_k(t), \qquad x_k(0) = x_0, \tag{1}$$

$$y_k(t) = Cx_k(t), \qquad t \in [0, N]. \tag{2}$$

The time index is denoted t and the input, output and state on trial k at time t are respectively denoted as $u_k(t)$, $y_k(t) \in \mathbb{R}$ and $x_k(t) \in \mathbb{R}^n$ (where n is the system order). Matrices $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times 1}, C \in \mathbb{R}^{1\times n}$ define the dynamics in each trial. When the trial ends, the system is reset to x_0 and the task is repeated.

Daniel Hobson, Bing Chu and Xiaohao Cai are with the Department of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom. Email: dsh1g18@soton.ac.uk; b.chu@soton.ac.uk; x.cai@soton.ac.uk

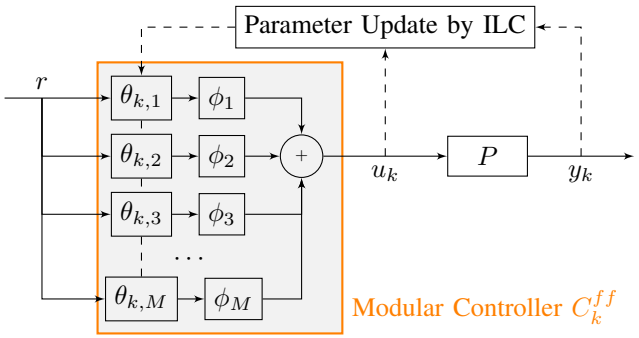

Fig. 1. Modular controller with parameters learnt to reconstruct the inverse of the system dynamics from a linear combination of M feedforward blocks.

We write the input and output on trial k into the vectors $u_k, y_k \in \mathbb{R}^N$ as

$$u_k = \begin{bmatrix} u_k(0) & u_k(1) & \cdots & u_k(N-1) \end{bmatrix}^\top, \tag{3}$$

$$y_k = \begin{bmatrix} y_k(1) & y_k(2) & \cdots & y_k(N) \end{bmatrix}^\top. \tag{4}$$

Then we write the dynamics in lifted form [21] as $y_k = Pu_k + d$, where P is the system matrix, i.e.,

$$P = \begin{bmatrix} CB & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & \cdots & CB \end{bmatrix}, \tag{5}$$

and the initial conditions are contained in d as defined below

$$d = \begin{bmatrix} CAx_0 & CA^2x_0 & \cdots & CA^Nx_0 \end{bmatrix}^\top. \tag{6}$$

The above assumes the system has relative degree one, i.e., $CB \neq 0$. For systems of higher relative degrees, the problem can be reformulated to accommodate. Without loss of generality, we take $x_0 = 0$ and then we have $d = 0$.

The problem is to use information from previous trials to generate a sequence of control inputs u_{k+1} for the upcoming trial $k+1$, to achieve small tracking error

$$e_{k+1} = r - y_{k+1} \tag{7}$$

between the system output y_{k+1} and a specified reference r (i.e., the vector form representation of the reference signal $r(t)$). We seek to achieve small e_{k+1} in as few iterations as possible, ideally, $e_k \to 0$ as $k \to \infty$.

III. A Modular-Based ILC Design

In this section, we first introduce our modular-based ILC design, then present the solution of the design using ADMM.

A. Controller Structure

We begin by declaring u_{k+1} to be generated from a parameterised controller, which is the linear combination of M modules and is illustrated in Figure 1, i.e.,

$$u_{k+1} = \sum_{i=1}^{M} \theta_{k+1,i}\phi_i r = C_{k+1}^{ff} r. \tag{8}$$

This replaces the problem of selecting u_{k+1} with the problem of selecting the parameters $\{\theta_{k+1,i}\}_{i=1}^M$, and we collect these parameters into the vector $\theta_{k+1} = \begin{bmatrix} \theta_{k+1,1} & \cdots & \theta_{k+1,M} \end{bmatrix}^\top$. The modules $\phi_i \in \mathbb{R}^{N\times N}$ are expressed as lifted form representations (of certain dynamical systems). We seek θ_{k+1} such that C_{k+1}^{ff} approaches the inverse of the system dynamics, yielding the perfect tracking $y_{k+1} \to r, \forall\, r \in \mathbb{R}^N$, using information from previous trials. We aim to find the best parameters by iteratively minimising the cost function

$$f(\theta) = \lambda_1 f_1 + \lambda_2 f_2, \tag{9}$$

comprised of the two terms

$$f_1(\theta) = \|r - P\phi_r\theta\|_2^2, \tag{10}$$

$$f_2(\theta) = \|\phi_r\theta - \begin{bmatrix} \phi_1 P\phi_r\theta & \cdots & \phi_M P\phi_r\theta \end{bmatrix} \theta\|_2^2, \tag{11}$$

where $\lambda_1, \lambda_2 > 0$ are weighting parameters and the $N \times M$ matrix ϕ_r is defined by $\phi_r = \begin{bmatrix} \phi_1 r & \cdots & \phi_M r \end{bmatrix}$.

Note that the first term f_1 in (9) is the tracking error, since $\phi_r\theta$ is the input signal defined in (8) and therefore $y_k = P\phi_r\theta_k$. Also the second term f_2 can be expressed as $f_2 = \|u - \phi_y\theta\|_2^2$ (with ϕ_y defined analogously to ϕ_r) – which is the error in reconstructing the input signal from the output, i.e. a metric of model mismatch. As such, the ratio λ_1/λ_2 has the interpretation of balancing the tracking error and the approximation error of the system's inverse representation. Without loss of generality, we let $\lambda_1 \in (0, 1]$ and $\lambda_2 = 1 - \lambda_1$. In this way, they directly represent the proportion of the information from each source; generally λ_1/λ_2 should be larger if the prior model is trusted and the measurement noise is large, and smaller if the prior model is poor and the measurements are accurate.

B. ADMM-Based Solution

The problem (9) is in general nonconvex. To see this, note that in the limit of $\lambda_1 \to 0$, $f(\theta)$ becomes f_2 which is minimised at both the true solution $\theta = \theta^*$ (defined by $P\phi_r\theta^* = r$, i.e. θ^* yields perfect tracking) *and* a degenerate solution of $\theta = 0$. To efficiently minimise this problem, we introduce an auxiliary variable ζ in f_2 and define

$$f_2(\theta, \zeta) = \|\phi_r\theta - \begin{bmatrix} \phi_1 P\phi_r\theta & \cdots & \phi_M P\phi_r\theta \end{bmatrix} \zeta\|_2^2. \tag{12}$$

Then minimising (9) is equivalent to finding the solution of the following optimisation problem, i.e.,

$$\min_{\theta,\zeta} f(\theta, \zeta) = \lambda_1 f_1(\theta) + \lambda_2 f_2(\theta, \zeta), \tag{13}$$

$$\text{s.t.} \quad \zeta = \theta. \tag{14}$$

Remark 1: Note that $f_2(\theta, \zeta)$ in (12) is a *biconvex* function [22], meaning that for each of θ and ζ held fixed, the problem is convex over the other.

To solve problem (13), we apply the ADMM algorithm [19] by first constructing the augmented Lagrangian of the problem, using σ to denote dual variables, i.e.,

$$\mathcal{L}(\theta, \zeta, \sigma) = f(\theta, \zeta) + \sigma^\top(\theta - \zeta) + \frac{\rho}{2}\|\theta - \zeta\|_2^2 \tag{15}$$

where $\rho > 0$ is a parameter to choose. The ADMM algorithm proceeds by alternately minimising with respect to decision variables ζ, θ, σ over $\mathcal{L}(\theta, \zeta, \sigma)$. The $(k+1)$-th

iterate is generated by solving a sub-minimisation problem with respect to each of ζ and θ, as

$$\zeta_{k+1} = \text{argmin}_{\zeta}\mathcal{L}(\theta_k, \zeta, \sigma_k), \tag{16}$$

$$\theta_{k+1} = \text{argmin}_{\theta}\mathcal{L}(\theta, \zeta_{k+1}, \sigma_k). \tag{17}$$

Then, the dual variable σ is updated by setting

$$\sigma_{k+1} = \sigma_k + \rho(\theta_{k+1} - \zeta_{k+1}). \tag{18}$$

Note that the sub-optimisation problems (16) and (17) have analytic solutions, as given by the following proposition.

Proposition 1: The sub-minimisations (16) and (17) admit analytic solutions

$$\zeta_{k+1} = (2\lambda_2\phi_{y_k}^\top\phi_{y_k} + \rho I)^{-1} (2\lambda_2\phi_{y_k}^\top u_k + \rho\theta_k + \sigma_k), \tag{19}$$

$$\theta_{k+1} = (2\lambda_1\phi_r^\top P^\top P\phi_r + 2\lambda_2\Phi_{\zeta_{k+1}}^\top\Phi_{\zeta_{k+1}} + \rho I)^{-1} (2\lambda_1\phi_r^\top P^\top r + \rho\zeta_{k+1} - \sigma_k), \tag{20}$$

where the elements of $\zeta_{k+1} = \begin{bmatrix}\zeta_{k+1,1} & \cdots & \zeta_{k+1,M}\end{bmatrix}^\top$ are used to construct the intermediate result

$$\Phi_{\zeta_{k+1}} \triangleq (I - \zeta_{k+1,1}\phi_1 P \cdots - \zeta_{k+1,M}\phi_M P)\,\phi_r. \tag{21}$$

Proof: The solutions to the sub-minimisation problems (16) and (17) are respectively obtained by solving

$$\frac{\partial}{\partial\zeta}\mathcal{L}(\theta_k, \zeta, \sigma_k) = 0 \quad \text{and} \quad \frac{\partial}{\partial\theta}\mathcal{L}(\theta, \zeta_{k+1}, \sigma_k) = 0. \tag{22}$$

The former can be done directly and produces (19) as required. To solve the latter, observe that

$$\phi_r\theta - \begin{bmatrix}\phi_1 P\phi_r\theta & \cdots & \phi_M P\phi_r\theta\end{bmatrix}\zeta = (I - \zeta_{k+1,1}\phi_1 P \cdots - \zeta_{k+1,M}\phi_M P)\,\phi_r\theta. \tag{23}$$

With the definition (21), then we have the solution given by (20), the details of which are omitted here for brevity. ■

The full algorithm is summarised in Algorithm 1.

Algorithm 1: Modular ILC Algorithm by ADMM

Input : $\lambda_1, \lambda_2, \rho, e_{\text{stop}}$

Output: θ for minimising $f(\theta)$ in (9)

Initialise θ_0, σ_0, $k \leftarrow 0$;
while $\|e_k\|_2 > e_{\text{stop}}$ **do**
  Construct ϕ_r and ϕ_{y_k};
  Update ζ as in (19);
  Construct $\Phi_{\zeta_{k+1}}$ as in (21);
  Update θ as in (20);
  Update σ as in (18);
  Generate the control signal u_{k+1} as in (8);
  Apply u_{k+1} to the system (1);
  $k \leftarrow k + 1$.
end

Remark 2: On each trial k, many ADMM update steps could be taken before the result is applied to the system on the next trial $k+1$ as in [28], which would be appropriate if the learning rate was low. In this work we take one ADMM step of the optimisation (13) per trial, so that the iteration index for the optimisation corresponds with the trial number and we may reuse k in our notation for readability.

IV. Convergence Analysis

Algorithm 1 has well-defined convergence, given below.

Theorem 1: Algorithm 1 generates a sequence of parameters $\theta_0, \ldots, \theta_k$ converging to a stationary point, say θ_∞, minimising the cost function (9) (or a compact set of such points) when the matrix ϕ_r has full column rank and ρ is chosen to be sufficiently large.

Proof: The proof is presented in Appendix I. ■

Remark 3: The full column rank condition on ϕ_r corresponds with linear independence in the modules when applied to the reference. It generalises the typical choice of modules, which is a set of orthogonal basis functions.

Theorem 1 guarantees that Algorithm 1 converges to a stationary point of problem (9) with the following properties.

The Fully-Parameterised Case: When our chosen modules form a complete basis, or the true system dynamics are accurately approximated in their span, our algorithm approaches the point that parameterises the true dynamics. This is because problem (9) has a single stationary point corresponding with the parameterisation of P^{-1}, which leads to perfect tracking $e_k \to 0$. The number of modules M required to span a space containing the true dynamics depends on the complexity of the system and the dynamics of each module. In Section VI, we discuss an example system and illustrate how the modules may be *well adapted* to represent that system, by which we mean they exactly parameterise the true dynamics with a small number of simple modules.

The Under-Parameterised Case: When the true dynamics lie outside the span of the modules, the converged controller is a weighted balance between the tracking accuracy and a simplified system model that lies in the module span. In this case, the designer chooses the ratio λ_1/λ_2 to reflect their design intent. Increasing this ratio prioritises the reduction of tracking error, leading to converged θ_k with low e_k as $k \to \infty$. A smaller λ_1/λ_2 prioritises the reduction of approximation error in the system inverse reconstruction, which implies improved generalisation to track a different reference trajectory on future trials with reduced error.

V. Discussion of Generalisation Properties

So far, Algorithm 1 has been presented in the traditional ILC setting; with an *identical* reference on every trial, directly learning the control sequence would be appropriate. However in practice, the system may be required to track *multiple* trajectories (which we denote r_k) – then the optimal control sequence changes and it is not suitable to use an incremented copy of the previous control for the new task. This results in ILC designs 'forgetting' learnt information when the reference changes, rather than *continuing* to refine a reference-agnostic representation. Modular schemes parameterise the system inverse, which is invariant to reference changes and when the true system inverse P^{-1} lies in the span of the chosen modules, the converged parameterisation

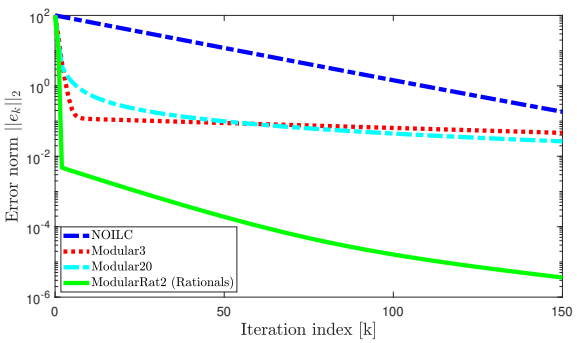

Fig. 2. Tracking error evolution for a selection of module choices, compared against NOILC. Parameters: $\lambda_1 = \lambda_2 = 0.5, \rho = 100$, $\lambda_{\text{NOILC}} = 0.01$.

is a good approximation of it. This yields accurate tracking irrespective of how the reference changes, since

$$y_k = Pu_k = PC_k^{ff} r_k = PP^{-1} r_k = r_k. \tag{24}$$

As such, the converged controller is applicable to the iteration-varying reference case. While the trajectory through parameter space varies for different references, problem (13) is solved as $C_k^{ff} \to P^{-1}$ regardless of the current reference. This makes it appropriate to apply Algorithm 1 even when the reference changes during the learning process. In this way, these modular schemes achieve the dual objectives of accurate tracking in the next trial with respect to a prior model and improvement of that model in order to achieve low error on future trials, even when tracking *dissimilar* tasks.

Additionally, ILC designs assume fixed-length trials and require specific extensions to handle early termination [25]. Varying trial lengths may be incorporated naturally when the control input is generated by a modular model. The bases ϕ_i, as lifted representations, can be generated for whatever trial length N_k is required.

VI. Numerical Example

In this section we present two numerical examples to demonstrate the proposed design.

A. Fixed-Reference Case

Consider the test system with the state space dynamics

$$A = \begin{bmatrix} 0.2 & 0 \\ 0.1 & 0.3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}, \tag{25}$$

which we seek to control with a selection of modular-based controllers to track the reference

$$r(t) = 2\sin(0.5t). \tag{26}$$

As shown in Section IV, the convergence of the parameters generated by Algorithm 1 is assured for any linearly independent modules. Therefore, we present and discuss the performance of the algorithm with a selection of bases as follows: 3 terms of the form z^j, i.e., $\{z, 1, z^{-1}\}$ (labelled 'Modular3'), and 20 such terms, i.e., $\{z, 1, z^{-1} \ldots, z^{-18}\}$ ('Modular20'), and two rational functions corresponding with the partial fraction expansion of the plant dynamics, i.e., $\{10z - 4, -10/(z - 0.1)\}$ ('ModularRat2'). We include also the performance of the well-known norm-optimal ILC (NOILC) scheme [7] to emphasise the improved performance of modular parameterised schemes when the reference changes. The NOILC scheme solves the problem

$$u_{k+1} = \arg\min_u \{\lambda_{\text{NOILC}} ||e_{k+1}||_2^2 + ||u - u_k||_2^2\}, \tag{27}$$

and the learning rate λ_{NOILC} is set at $1/\rho$ for fair results – since doing so ensures the term $||u - u_k||_2^2$ is assigned comparable weight to the term $|\theta - \zeta\|_2^2$ in (15).

Convergence of the Tracking Error: The tracking performance of each scheme for the fixed-reference case is shown in Figure 2. The NOILC scheme produces gradual improvement, which eventually tends to $e_k = 0$. The schemes 'Modular3' and 'Modular20' are initialised with coefficients at zero. Since the true dynamics are described by a rational transfer function, a finite number of these terms are insufficient to completely model the system [14]. The 3-term controller has favourable convergence in early trials, but beyond trial $k \approx 10$, the model parameterisation has reached convergence and the tracking error cannot improve further, i.e., under-modelling the true dynamics imposes a performance limit. The conventional approach to reduce the model mismatch is to include more basis functions, as seen in 'Modular20'. The larger model will eventually reach a more accurate representation and therefore improve the tracking accuracy, but to reach the optimal point requires searching over a much larger parameter space. Doing so takes considerably more iterations and as such the larger polynomial model is not substantially more accurate than the simplified model in the first 150 trials.

In contrast, the fast learning of an accurate representation is achieved by the rational model 'ModularRat2', with only two bases. This is because the bases chosen for this scheme are *well adapted* for the true dynamics. For the example system (25), the transfer function $P(z)$ may be written as

$$P(z) = \frac{z - 0.1}{(z - 0.2)(z - 0.3)}, \tag{28}$$

and its inverse may therefore be expressed as

$$P^{-1}(z) = z - 0.4 + \frac{0.02}{z - 0.1}. \tag{29}$$

This describes the true dynamics as a linear combination of three modules, i.e., the lifted form P in (5) includes the terms corresponding to the two leading Markov parameters and a term from the first order system corresponding with the zero of $P(z)$ (for systems with n_z zeros, this becomes n_z modules where each corresponds with the dynamics of a zero). As such, in the ideal case we can choose appropriate modules such that $P^{-1}(z)$ lies in their span. For ease of presentation, we remove one degree of freedom by including both the z^1 and z^0 coefficients in a single module, i.e., let ϕ_1 correspond with $10z - 4$ and ϕ_2 with $-10/(z - 0.1)$, then

$$P^{-1}(z) = 0.1\phi_1 - 0.002\phi_2, \tag{30}$$

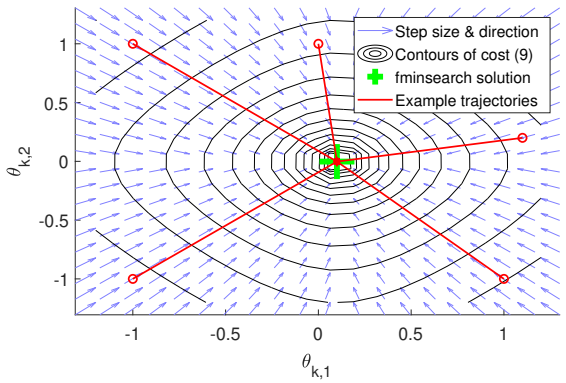

Fig. 3. Parameter convergence for 'ModularRat2', with example trajectories converging to θ^*. Contour lines indicate equal-cost manifolds of (9) and background arrows indicate update step size and direction over the grid.

and the optimal solution is $\theta^* = \begin{bmatrix} 0.1 & -0.002 \end{bmatrix}^\top$. By selecting our modules to match the true dynamics, we learn an accurate representation of these parameters efficiently (i.e., with few modules and in few iterations) and obtain accurate models suitable for high performance tracking. Importantly, the rational model *does not* need to be constructed with the true system zeros to exceed the performance of the other representations presented here. The model mismatch is bounded by the differences between the zeros of the true system and the model [17], so any chosen modules with zeros closer to the true values than $z = 0$ more accurately reconstruct the true dynamics (with equal number of terms) when compared to controllers of the form z^j.

Convergence of the Model Parameters: By our choice to set $\phi_1 = 10z - 4$, the rational scheme is presented with only two modules. Therefore, θ_k can be displayed as the phase portrait in Figure 3. We illustrate the trajectories from a number of example starting points, all of which converge to the stationary point $\theta_\infty = \begin{bmatrix} 0.1 & -0.002 \end{bmatrix}^\top$. In particular, we observe in this example that the first step from each starting point θ_0 results in an estimate θ_1 that is close to the true solution ($\|\theta_1 - \theta^*\|_2$ order of 10^{-4}), which explains the improvement in tracking error to $e_1 \approx 10^{-2}$ in the first trial seen in Figure 2. This behaviour is also illustrated by the direction of the background arrows. These point in the direction of the update step from each point on the grid and have length corresponding to the step size. They give the impression of the parameter flow across the entire space and attract to the global optimum, which is computed with `fminsearch` and marked. The gradient of the cost function (9) is indicated by the contour lines, showing the converged point as the minimiser of this function.

B. Reference-Varying Case

We now present the same problem (25) in the challenging case where the reference is randomly selected on each trial from one of the options

$$r_\alpha(t) = 2\sin(0.5t), \qquad r_\gamma(t) = 4\sin(0.5t), \tag{31}$$

$$r_\beta(t) = 3\sin(0.5t), \qquad r_\delta(t) = 2\sin(t). \tag{32}$$

The error norms for each algorithm are presented in Figure 4. In these conditions the tracking error achieved by NOILC remains at $e_k \approx 10^2$ for all trials, since it applies an incremented copy of the control sequence used for the *previous* reference. With the reference varying, the desired control sequence must vary on each trial and NOILC is unable to converge. For more dissimilar references, this effect is exaggerated. The modular schemes are qualitatively different; the learnt representation C_k^{ff} is still appropriate and the new control sequence is similarly accurate to the previous task. That is, the modular schemes achieve a level of generalisation to potentially unseen tasks – mimicking how biological systems model complex, often nonlinear, dynamics. This underscores the value of a provably convergent ILC design that accepts any set of modules with the mild condition of full-rank ϕ_r, as we may utilise a wide selection of modules with desirable behaviour.

VII. Conclusions and Future Work

We propose a novel modular-based ILC design algorithm inspired by biological systems. We show that the convergence of the parameters of such a controller is guaranteed under mild assumptions by using an ADMM-based design. We present numerical examples highlighting the improved tracking performance of parameterised ILC over NOILC and that such schemes can be successfully applied with iteration-varying references too. We discuss the need for modules that share properties of the true system dynamics and the links to biological controllers which have motivated this work.

In these biological systems, the basis representations themselves change – adding, removing and modifying modules – over time. The processes that drive the module selection are not well understood, but since it is clear that appropriate modules are crucial to efficient learning, considering how to alter the basis to achieve a specified level of model accuracy and interpretability *during* the data-collection procedure offers additional opportunities for research. By viewing our algorithm as an iterative method to recover a separable and potentially low-rank approximation of the unknown matrix P^{-1}, we may inform the future selection of modules using existing tools from linear algebra such as a truncated singular value decomposition [26]. Additionally, if we consider nonminimum phase systems, the problem (9) becomes that of stable inversion.

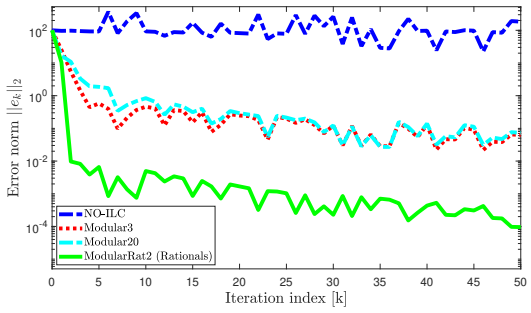

Fig. 4. Tracking error evolution with trial-varying references r_k.

Considering sparsity-promoting optimisation methods (as in e.g. [27]) is also an interesting direction to extend this work. In the context of iteration-varying basis representations, the magnitudes of the parameters forms a natural metric for choosing which modules to adjust or remove.

APPENDIX I
PROOF OF THEOREM 1

Proof: We first show that problem (13) has four properties: A1 (*coercivity*), A2 (*feasibility*), A3 (*Lipschitz sub-minimisation paths*), and A4 (*Lipschitz cost*).

*a) A1 (*coercivity*):* The problem is coercive if feasible decision variables of infinitely large size yield infinitely large costs. Denoting the Frobenius norm $\|\cdot\|_F$, that is

$$f(\theta,\zeta)\to\infty,\quad \forall\,\|(\theta,\zeta)\|_F\to\infty,\quad \theta=\zeta. \tag{33}$$

Coercivity is ensured by the model-based term f_1, provided that the matrix $P\phi_r$ has trivial null space. Additionally, we formulate the problem with a full-rank representation of P in (5). As such, A1 is satisfied if ϕ_r has full column rank.

*b) A2 (*feasibility*):* The feasibility of the problem holds trivially as the only constraint is the artificial one $\zeta=\theta$.

*c) A3 (*Lipschitz sub-minimisation paths*):* Lipschitz sub-minimisation paths are assured when the sub-minimisation problems have unique solutions and the change in one decision variable causes a Lipschitz-bounded change in the corresponding optimal value of the other decision variable. These conditions are satisfied by observing that (13) is biconvex with sufficiently large ρ. For a detailed discussion of the bound on ρ see [19, Lemma 9].

*d) A4 (*Lipschitz cost*):* The cost function is the sum of squared L2-norms, which are smooth functions. As such, the cost is smooth with Lipschitz bound L_f. In practice, this bound L_f can be found over a suitable domain $|\theta|\leq d_\theta$ and then used to choose ρ sufficiently large to satisfy A3.

These properties then establish problem (13) as a special case of the general result in [19, Theorem 2], which proves the convergence from any initial point of ADMM to a feasible stationary point of the cost function (or a compact set of such points). As such, Theorem 1 holds. Since the converged solution is feasible, it also solves the original problem of minimising the cost function (9). ■

REFERENCES

[1] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," IEEE Control Systems Magazine, vol. 26, no. 3, pp. 96-114, Jun. 2006.

[2] D. Owens and J. Hätönen, "Iterative learning control - An optimization paradigm," Annual Reviews in Control, vol. 29, no. 1, pp. 57-70, Jan. 2005.

[3] E. Rogers, B. Chu, C. Freeman, and P. Lewin, Iterative learning control algorithms and experimental benchmarking. Wiley, Jan. 2023.

[4] S. Mandra, K. Galkowski, E. Rogers, A. Rauh, and H. Aschemann, "Performance-enhanced robust iterative learning control with experimental application to PMSM position tracking," IEEE Transactions on Control Systems Technology, vol. 27, no. 4, pp. 1813-1819, Jul. 2019.

[5] E. Rogers, D. Owens, H. Werner, C. Freeman, P. Lewin, S. Kichhoff, C. Schmidt, and G. Lichtenberg, "Norm-optimal iterative learning control with application to problems in accelerator-based free electron lasers and rehabilitation robotics," European Journal of Control, vol. 16, no. 5, pp. 497-522, Jan. 2010.

[6] C. Xu et al, "Application of a modified iterative learning control algorithm for superconducting radio-frequency cavities," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 1026, Mar. 2022.

[7] N. Amann, "Optimal algorithms for iterative learning control." Ph.D., University of Exeter, 1996.

[8] D. Wolpert, J. Diedrichsen, and J. R. Flanagan, "Principles of sensorimotor learning," Nature Reviews Neuroscience, vol. 12, no. 12, pp. 739-751, Dec. 2011.

[9] J. Tresilian, Sensorimotor Control and Learning: An introduction to the behavioral neuroscience of action, 2012th ed. Basingstoke, Hampshire: Bloomsbury Academic, Jul. 2012.

[10] F. Mussa-Ivaldi and E. Bizzi, "The modular organization of motor control: what frogs can teach us about adaptive learning," IFAC Proceedings Volumes, vol. 28, no. 15, pp. 413-418, Jun. 1995.

[11] D. Hoelzle, A. Alleyne, and A. Wagoner Johnson, "Basis task approach to iterative learning control with applications to micro-robotic deposition," IEEE Transactions on Control Systems Technology, vol. 19, no. 5, pp. 1138-1148, Sep. 2011.

[12] J. van de Wijdeven and O. Bosgra, "Using basis functions in iterative learning control: analysis and design theory," International Journal of Control, vol. 83, no. 4, pp. 661-675, Apr. 2010.

[13] J. Bolder, T. Oomen, S. Koekebakker, and M. Steinbuch, "Using iterative learning control with basis functions to compensate medium deformation in a wide-format inkjet printer," Mechatronics, vol. 24, no. 8, pp. 944-953, Dec. 2014.

[14] J. Bolder and T. Oomen, "Rational basis functions in iterative learning control - with experimental verification on a motion system," IEEE Transactions on Control Systems Technology, vol. 23, no. 2, pp. 722-729, Mar. 2015.

[15] J. van Zundert, J. Bolder, and T. Oomen, "Optimality and flexibility in iterative learning control for varying tasks," Automatica, vol. 67, pp. 295-302, May 2016.

[16] P. M. J. Van Den Hof, System identification data-driven modelling of dynamic systems, Feb. 2020.

[17] P. Heuberger, P. Van den Hof, and O. Bosgra, "A generalized orthonormal basis for linear dynamical systems," IEEE Transactions on Automatic Control, vol. 40, no. 3, pp. 451-465, Mar. 1995.

[18] D. Hobson, B. Chu, and X. Cai, "Biologically-inspired iterative learning control design: a modular-based approach," in 2024 UKACC 14th International Conference on Control (CONTROL), Apr. 2024, pp. 175-176.

[19] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," Journal of Scientific Computing, vol. 78, no. 1, pp. 29-63, Jan. 2019.

[20] L. Blanken, G. Isil, S. Koekebakker, and T. Oomen, "Flexible ILC: towards a convex approach for non-causal rational basis functions," IFAC-PapersOnLine, vol. 50, no. 1, pp. 12107-12112, Jul. 2017.

[21] J. J. Hätönen, D. H. Owens, and K. L. Moore, "An algebraic approach to iterative learning control," International Journal of Control, vol. 77, no. 1, pp. 45–54, Jan. 2004.

[22] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: a survey and extensions," Mathematical Methods of Operations Research, vol. 66, no. 3, pp. 373-407, Dec. 2007.

[23] R. A. Schmidt and T. D. Lee, Motor control and learning: a behavioral emphasis, 4th ed. Champaign, IL: Human Kinetics, 2005.

[24] M. Jeannerod, Attention and performance XIII: motor representation and control, ser. Attention and performance. Hillsdale, N.J.: L. Erlbaum, 1990.

[25] J.-X. Xu, "A survey on iterative learning control for nonlinear systems," International Journal of Control, vol. 84, no. 7, pp. 1275-1294, Jul. 2011.

[26] S. Brunton and N. Kutz, Data-driven science and engineering: machine learning, dynamical systems, and control. Cambridge University Press, Feb. 2019.

[27] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," Proceedings of the National Academy of Sciences, vol. 113, no. 15, pp. 3932-3937, Apr. 2016.

[28] X. Cai, L. Pratley, and J. D. McEwen, "Online radio interferometric imaging: assimilating and discarding visibilities on arrival," Monthly Notices of the Royal Astronomical Society, vol. 485, no. 4, pp.4559–4572, Jun. 2019.