A Decentralised Iterative Learning Control Framework for Collaborative Tracking

Shangcheng Chen[a,*], Christopher T. Freeman[a]

[a] *Faculty of Engineering and Physical Sciences, Electronics and Computer Science, University of Southampton, SO17 1BJ, United Kingdom*

Abstract

Collaborative tracking control involves two or more subsystems working together to perform a global objective, and is increasingly used within a diverse range of applications. Decentralised iterative learning control schemes have demonstrated highly accurate collaborative tracking by using past experience gained over repeated attempts at the task. However they impose highly restrictive constraints on the system dynamics, and their reliance on inverse dynamics has degraded their robustness to model uncertainty.

This paper proposes the first general decentralised iterative learning framework to address this problem, thereby enabling a wide range of existing iterative learning control methodologies to be applied in a decentralised manner to collaborative subsystems. This framework is illustrated through the derivation of a variety of new decentralised iterative learning control algorithms which balance collaborative tracking performance with optimisation of a general objective function. The framework is illustrated by application to wearable stroke rehabilitation technology in which each subsystem is a muscle artificially activated by electrical stimulation. These verify the framework's simplified design and reduced hardware and communication overheads.

Keywords: iterative learning control, collaborative control, stroke rehabilitation

1. Introduction

A multi-agent system (MAS) consists of two or more autonomous agents interacting in a networked environment using distributed sensors, wireless communication protocols and a control architecture in order to achieve a global goal. A variety of control strategies exist that govern the cooperative behaviour of groups of agents, with the most common being consensus and formation control. These primarily focus on the agent state/output tracking a specified reference trajectory in a finite time, subject to a wide variety of communication topologies which govern allowable data exchange between agents.

Collaborative control is an alternative strategy in which the global objective is for the output/state of each agent (typically termed a 'subsystem') to combine together to track a specified reference trajectory in a finite time. This methodology is required for applications such as (i) active orthoses to assist elderly or impaired people to perform day-to-day tasks such as eating or walking, e.g. [1], (ii) human-robot teams for search and rescue operations, construction, and space exploration, e.g., [2], (iii) dual-stage positioners which combine multiple actuators which each operate over a different frequency band, e.g. [3], and (iv) groups of robotic manipulators grasping a common object on a production line, e.g. [4]. Collaborative control typically assumes each subsystem has access to the overall collaborative output signal, rather than individual outputs, which thereby minimises communication and sensor overheads.

As the task is typically repeated, there is a strong motivation to employ learning to maximise collaborative tracking precision. This has led to the recent development of iterative learning control (ILC) algorithms for collaborative tracking, inspired by their successful application to both formation and consensus tracking (see [5] for an overview). ILC is a method that uses experimental input and output data from previous attempts at the task to update the control input on the current attempt, with the objective of successively improving the tracking performance [6, 7, 8]. In [9] an 'inverse-based' ILC algorithm was applied to solve the collaborative tracking problem, in which the input update for each subsystem was computed by inverting the model of that subsystem. This achieved rapid convergence, but is highly sensitive to modelling uncertainty [10, 11, 12], i.e. robustness margins tend to zero at high frequencies. Similarly, [13] developed a ILC update for human-robot collaborative output tracking with two subsystems. It proposed a frequency-domain ILC inverse update for each subsystem and applied it to human-robot tracking task. In [14] a further inverse-based ILC update was considered, establishing asymptotic convergence to zero error.

The above ILC updates are decentralised in the sense that the lLC update for the i^{th} subsystem uses only the dynamic model of that subsystem, thereby simplifying design. While these approaches can all provide convergence to zero error, they have limitations: (i) their choice of objective constrains the system to have zero relative degree, (ii) they consider only single-input, single-output linear subsystem dynamics, (iii) they assume the task is achievable, (iv) the solutions are not optimal (i.e. no criteria is used to select inputs from the infinite number that solve the collaboration problem), (v) they all use inverse-based ILC updates, so are not robust to modelling uncertainty.

Existing ILC methods are not able to solve these limitations,

*Corresponding author

Email addresses: shangcheng.chen@soton.ac.uk (Shangcheng Chen), ctf1@soton.ac.uk (Christopher T. Freeman)

e.g. neither existing formation/consensus ILC updates, nor ILC approaches that decouple general multiple-input, multiple-output (MIMO) systems (see e.g. [15] and references therein) are compatible with the collaborative structure which (1) amalgamates all subsystem dynamics to generate the output, and (2) requires a decentralised structure in which the update for each subsystem only uses local model information. Moreover, the vast majority of ILC algorithms consider only zero error tracking, rather than error minimisation. Exceptions, such as the multi-sensor system approach of [16] do minimise a cost function, but fail to satisfy the above structural requirements and, like almost all formation/concensus ILC methods, provide only asymptotic convergence. They also consider purely SISO subsystem dynamics. The aim of this paper is to address these limitations, and in particular derive a decentralised ILC framework that:

- establishes a broad class of ILC algorithms that optimise control performance and energy costs simultaneously,
- allows subsystems to be added or removed arbitrarily,
- can be used for general classes of MIMO subsystems without restrictive constraints, e.g. non-fully functional, rank deficient, input-output delay or non-communicable dynamics, and
- demonstrates robustness which can be transparently traded for convergence by the designer.

Initial results were given in [17] which proposed a single ILC update and did not contain experimental results.

The control structure is demonstrated through experimental application to rehabilitation engineering. Here each subsystem corresponds to an electrically stimulated muscle with the global goal of assisting movement. By employing the decentralised ILC framework, three different types of ILC algorithm are successfully applied to overcome long-standing problems in the field of rehabilitation [18]. Results confirm accurate assistance of human movement and illustrate the utility of decentralised control to substantially reduce hardware and communication overheads, as well as simplifying design.

This paper is organised as follows: Section 2 defines the MAS collaborative problem and Section 3 sets out the general framework for decentralised collaborative ILC. Specific example designs are proposed in Section 4, and Section 5 provides robustness properties. In Section 6 experimental results are presented, with conclusions and future work in Section 7.

Denote the set of real numbers as $\mathbb{R}$, and the set of integers as $\mathbb{N}$. The symbol k denotes the trial number and $k \in \mathbb{N}_{\geq 0}$. For any vector $x \in \mathbb{R}^n$, $\|x\| = \sqrt{x^\top x}$. For any matrix $A \in \mathbb{R}^{n\times n}$, $\|A\|$ is the induced norm of the vector norm, $\lambda_i(A)$ denotes the i^{th} eigenvalue of A, $\underline{\lambda}(A)$ is the minimal eigenvalue of A, $\overline{\lambda}(A)$ is the maximum eigenvalue of A, and $\rho(A) = \max_i |\lambda_i(A)|$ is the spectral radius of A, and $\underline{\sigma}(A)$, $\overline{\sigma}(A)$ are respectively the minimal and maximum singular value of A. The notation $A^\dagger$ denotes the Moore-Penrose inverse of any matrix A, with $A^\dagger = (A^\top A)^{-1}A^\top$ in the special case when A has linearly independent columns, and $A^\dagger = A^\top(AA^\top)^{-1}$ when A has linearly independent rows. Also P_A denotes the orthogonal projection onto the range of A, and $P_A^\perp = I - P_A$ is the orthogonal projection onto the null space of A. Where necessary for clarification, the $n \times n$ identity and zero matrices are denoted by I_n and 0_n, respectively. The null-space and column space of any matrix A are denoted $\mathscr{N}(A)$ and $\mathscr{C}(A)$ respectively.

2. PROBLEM FORMULATION

This section first defines the standard collaborative tracking control problem, and the corresponding ILC objective employed in [9, 14, 13]. We then show this generally leads to a centralised ILC algorithm, and also places unnecessary constraints on the system dynamics.

2.1. Standard ILC Statement

Consider a MAS system, consisting of n MIMO linear time invariant (LTI) subsystems, collaborating to perform a repeated tracking task of duration N samples. Subsystem i has p_i inputs and q outputs, and on the k^{th} trial is modelled by the system

$$\begin{aligned} x_{i,k}(t+1) &= A_i x_{i,k}(t) + B_i u_{i,k}(t), \quad x_{i,k}(0) = x_{i,0} \\ y_{i,k}(t) &= C_i x_{i,k}(t) + D_i u_{i,k}(t), \quad i = 1, 2, \cdots, n. \end{aligned} \tag{1}$$

The time step is $t = 0, 1, \cdots, N-1$, $x_{i,k}(t) \in \mathbb{R}^{n_i}$ is the state variable, $y_{i,k}(t) \in \mathbb{R}^q$ the output variable, $u_{i,k}(t) \in \mathbb{R}^{p_i}$ the control input variable, $A_i \in \mathbb{R}^{n_i\times n_i}$, $B_i \in \mathbb{R}^{n_i\times p_i}$ and $C_i \in \mathbb{R}^{q\times n_i}$ and $D_i \in \mathbb{R}^{q\times p_i}$.

At the end of every trial, subsystem i is reset to initial condition $x_{i,0}$ which is fixed for all trials. The system output over the k^{th} trial is the combined sum of all subsystem contributions, i.e.

$$y_k(t) = \sum_{i=1}^{n} Q_i y_{i,k}(t), \tag{2}$$

where $Q_i \in (0, 1]$ is a scalar weight and $y_k(t)$ is assumed to be measured directly and accessible to all subsystems. The standard collaborative tracking aim (see [9, 13, 14]) is for $y_k(t)$ to follow a reference $y_d(t) \in \mathbb{R}^q$ with minimal tracking error $e_k(t)$, i.e.

$$\lim_{k\to\infty} e_k(t) = y_d(t) - \lim_{k\to\infty} y_k(t) = 0. \tag{3}$$

While this is the standard task description in ILC, it places constraints on the system properties. To demonstrate this, let each trial be represented as a single sample of a high dimensional 'lifted' system. Accordingly, the signals in (1)-(3) become the super-vectors (for $i = 1, \ldots, n$)

$$\begin{aligned} u_{i,k} &= [u_{i,k}(0)^\top \; u_{i,k}(1)^\top \; \cdots \; u_{i,k}(N-1)^\top]^\top \in \mathbb{R}^{p_i N}, \\ y_{i,k} &= [y_{i,k}(0)^\top \; y_{i,k}(1)^\top \; \cdots \; y_{i,k}(N-1)^\top]^\top \in \mathbb{R}^{qN}, \\ y_k &= [y_k(0)^\top \; y_k(1)^\top \; \cdots \; y_k(N-1)^\top]^\top \in \mathbb{R}^{qN}, \\ y_d &= [y_d(0)^\top \; y_d(1)^\top \; \cdots \; y_d(N-1)^\top]^\top \in \mathbb{R}^{qN}. \end{aligned}$$

Each subsystem (1) can then be written in the equivalent form

$$y_{i,k} = G_i u_{i,k} + d_i, \quad i = 1, 2, \cdots, n, \tag{4}$$

where $G_i \in \mathbb{R}^{qN \times p_i N}$ is the Toeplitz matrix

$$G_i = \begin{bmatrix} g_i(0) & 0 & \cdots & 0 & 0 \\ g_i(1) & g_i(0) & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & g_i(0) & 0 \\ g_i(N-1) & \dots & \dots & g_i(1) & g_i(0) \end{bmatrix}$$

in which the impulse response function

$$g_i(j) = \begin{cases} D_i & , \ j = 0 \\ C_i A_i^{j-1} B_i & , \ j > 0 \end{cases},$$

and d_i is the response to initial conditions, i.e.

$$d_i = [C_i x_{i,0} \ C_i A_i x_{i,0} \ \cdots \ C_i A_i^{N-1} x_{i,0}]^\top \in \mathbb{R}^{qN}.$$

Without loss of generality, d_i can be omitted from (4) and the reference redefined as $y_d - \sum_{i=1}^n d_i$. The collaborative tracking output (2) is then

$$y_k = \sum_{i=1}^{n} Q_i G_i u_{i,k} = QGu_k, \tag{5}$$

where $G = \text{diag}(G_1, \cdots, G_n)$, $u_k = [u_{1,k}^\top \ u_{2,k}^\top \ \cdots \ u_{n,k}^\top]^\top$, $Q = [Q_1 \ Q_2 \ \cdots \ Q_n] \otimes I_{qN}$ with $\otimes$ the Kronecker product.

The standard ILC implementation to solve (3) has form

$$u_{k+1} = u_k + \gamma L(y_d - y_k),$$

which corresponds to the individual update for subsystem i

$$u_{i,k+1} = u_{i,k} + \gamma L_i(y_d - y_k)$$

with ILC operator $L = [L_1^\top \ L_2^\top \ \cdots \ L_n^\top]^\top$ and scalar γ selected by the user. It is well-known in ILC that L must satisfy

$$\|I - \gamma QGL\| < 1. \tag{6}$$

This shows the limitation of general ILC updates: (1) for a solution L to exist, G must have full rank which means at least one subsystem must have relative degree 0, and (2) a decentralised structure also requires that each L_i is selected with only knowledge of G_i. However, few ILC approaches have such a structure: the existing ILC application of [9] solved this using $L_i = G_i^{-1}$. Another approach is 'gradient ILC' (see [19, 20]) where $L_i = G_i^\top$, which was applied to collaborative control in our preliminary work [17]. To satisfy (6), both updates require every subsystem to have zero relative degree, the former is extremely sensitive to model uncertainty and disturbance [10, 11, 12], and the latter can be slow to converge. Nevertheless, these approaches will be used to inspire new ILC updates within the extended collaborative framework derived next.

2.2. Decentralised Collaborative ILC

To solve the limitations in ILC design for MAS, the new framework must first embed the required decentralised ILC update structure, i.e. subsystem i can only access the overall system output, y_k, and system model, G_i. Secondly, it must expand the collaborative tracking requirement so that it can be achieved by more general classes of MAS, e.g. non-fully functional, rank deficient or non-communicable dynamics. This generalisation must satisfy the original tracking requirement, while also ensuring other practical objectives are met, e.g. energy minimisation. This is achieved by introducing the following comprehensive decentralised ILC framework:

Definition 1. *A decentralised ILC update for the i^{th} subsystem must have the form*

$$u_{i,k+1} = f_i(u_{i,k}, e_k), \quad i = 1, 2, \cdots, n, \tag{7}$$

where function $f_i(\cdot)$ depends only on the subsystem dynamics $\{A_i, B_i, C_i, D_i\}$. When applied to each subsystem (1), output (2) must satisfy

$$\lim_{k\to\infty} y_k = y^* \tag{8}$$

and the input $u_k = [u_{1,k}^\top, u_{2,k}^\top, \cdots, u_{n,k}^\top]^\top$ must converge as

$$\lim_{k\to\infty} u_k = u^* \tag{9}$$

with the optimal limiting signals given by

$$y^* = \arg\min\{J(y_k) | J(y_k) = \|y_d - y_k\|^2\}, \tag{10}$$

$$u^* = \arg\min\{J(u_k) | J(u_k) = \sum_{i=1}^{n} \|u_{i,k}\|_{W_i}^2, y_k = y^*\}, \tag{11}$$

where W_i is a symmetric positive definite weighting matrix.

Definition 1 states that the output and input must both converge to optimal solutions. This is more general than the standard ILC formulation (3), which stipulates only zero error. Moreover, weight W_i can be used to address additional practical requirements.

3. DESIGN FRAMEWORK

3.1. Decentralised Algorithm

We introduce a general class of decentralised ILC algorithms to solve the problem of Definition 1 as follows.

Algorithm 1. *Define the decentralised ILC update algorithm*

$$u_{i,k+1} = u_{i,k} + \gamma L_i e_k, \quad i = 1, 2, \cdots, n \tag{12}$$

where L_i is any matrix such that $G_i L_i$ is symmetric positive semi-definite with $r_i := rank(Q_i G_i) = rank(Q_i G_i L_i)$, and $\gamma > 0$ is a scalar learning gain.

The next theorem establishes conditions for Algorithm 1 to solve the decentralised control problem.

Theorem 1. *Application of Algorithm 1 to subsystem (1) and (2) with an initial input for each subsystem given by $u_{i,0} = 0$, $i = 1, 2, \cdots, n$ and a gain γ satisfying*

$$0 < \gamma < \frac{2}{\rho\left(\sum_{i=1}^n Q_i G_i L_i\right)} \tag{13}$$

or more conservatively

$$0 < \gamma < \frac{2}{\sum_{i=1}^{n} Q_i \rho\,(G_i L_i)} \tag{14}$$

achieves monotonic reduction in tracking error norm, i.e.

$$\|e_{k+1}\|^2 \leq \|e_k\|^2 \tag{15}$$

and satisfies output convergence condition (8), (10) with

$$y^* = P_{QG} y_d. \tag{16}$$

It also satisfies input convergence condition (9), (11) with

$$u_i^* = L_i O^\top (OO^\top)^{-1} (U^\top U)^{-1} U^\top y_d, \tag{17}$$

where the $qN \times r$ matrix U, $r \times p_i N$ matrix O are the full rank decomposition of $\sum_{i=1}^{n}(Q_i G_i L_i)$ and $r := rank(QG)$. The corresponding optimal weight in (11) satisfies $L_i = Q_i W_i^{-1} G_i^\top$.

Proof. The proof is given in Appendix A. ■

Converged output (16) is the orthogonal projection onto the range of composite system QG. Therefore convergence to zero error is only possible if either the reference is chosen to belong to the range of QG, or if $P_{QG} = I$. The latter is equivalent to the requirement that $r = r^f = min\{qN, \sum_i p_i N\}$ (e.g. the composite system QG is full rank). Neither requirement can be guaranteed in practice since both y_d and G_i are likely to be pre-specified. Convergence condition (14) confirms that there always exists a suitably small $\gamma > 0$ which can be selected without knowledge of other subsystems.

The relation between the co-domains of individual subsystems $Q_i G_i$ and that of the overall system QG, is illustrated in Figure 1. It can be seen that the restrictive full rank condition

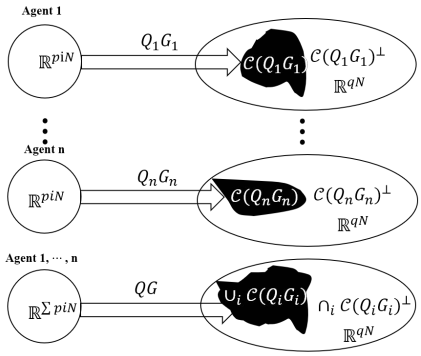

Figure 1: Individual and combined subsystem mapping relations.

on $\sum_{i=1}^{n}(Q_i G_i L_i)$ can be expressed purely in terms of the union of the achievable range of each subsystem.

3.2. Convergence properties

Theorem 1 defines a whole class of ILC updates, each converging to an optimal solution u^*, y^*. Each ILC update differs, however, in its convergence properties, as shown in the following result.

Theorem 2. *Application of Algorithm 1 under the conditions of Theorem 1, results in the error convergence relation*

$$e_{k+1} = (V_1(I - \gamma\Delta_r)V_1^\top + P_{QG}^\perp)e_k, \quad e_0 = y_d, \tag{18}$$

where Δ_r is the non-zero singular value matrix of $\sum_i Q_i G_i L_i$, and V_1 is the right singular vector corresponding to the non-zero singular values. It follows that the error on trial k satisfies

$$e_k = (V_1(I - \gamma\Delta_r)^k V_1^\top) y_d + P_{QG}^\perp y_d, \tag{19}$$

whose trial to trial evolution can also be expressed as

$$V_1^\top e_k = (I - \gamma\Delta_r)^k V_1^\top y_d. \tag{20}$$

Proof. This follows from the proof of Theorem 1. ■

Equation (19) shows that no convergence occurs in directions corresponding to zero singular values of $\sum_i Q_i G_i L_i$, reflecting the fact that final output (16) is not equal to y_d if $r \neq r^f$. The designer has no control over the number of zero singular values as the column space of QGL is equal to the column space of QG, which is dictated by individual subsystems via Figure 1. However, the designer can choose L_i to result in fast convergence by reducing $(I - \gamma\Delta_r)$ in (20). In principle, this term could be reduced to zero for all non-zero singular values resulting in convergence in one ILC trial. However, this will affect robustness properties, as examined in Section VI.

Theorem 3. *Application of Algorithm 1 under the conditions of Theorem 1, yields an error norm sequence in the interval*

$$(1 - \gamma\overline{\sigma}(\Delta_r))^k \leq \frac{\|V_1^\top e_k\|}{\|V_1^\top e_0\|} \leq (1 - \gamma\underline{\sigma}(\Delta_r))^k \tag{21}$$

where $\overline{\sigma}(\Delta_r)$ and $\underline{\sigma}(\Delta_r)$ are respectively the upper and lower non-zero singular values of QGL. If $\Delta_r = \overline{\sigma} I_r$ and $\Delta_r = \underline{\sigma} I_r$, the equality on the left and right are obtained respectively.

Proof. Follows from bounding of the equality (20). ■

4. APPLICATION OF FRAMEWORK

Theorem 1 can be used to design a wide variety of new decentralised ILC algorithms to solve the collaborative control problem. In addition, it can also be used:

1. to successfully implement standard ILC algorithms designed with only knowledge of each individual subsystem, and

2. to modify a standard ILC algorithm designed for the whole n-subsystem so that it assumes the required decentralised form, and can be applied to the case $r < r^f$.

Its utility is illustrated in the following examples, which each derive a new ILC update for MAS.

4.1. Gradient ILC

Gradient, or adjoint, ILC has received significant attention in the non-collaborative setting [19, 20], although it has only previously been considered in the full rank case, i.e. both the total system and all subsystems must be full rank ($r = r^f, r_i = r_i^f = min\{qN, p_iN\}$).The most general form of gradient algorithm uses operator

$$L_i = Q_i G_i^\top.$$

This satisfies the conditions of Algorithm 1 and therefore the update can be applied to the non-full rank case. Theorem 1 then holds with a convergence condition (13) of

$$0 < \gamma < \frac{2}{\rho\left(\sum_{i=1}^n Q_i^2 G_i G_i^\top\right)} \tag{22}$$

or a sufficient condition from (14) of

$$0 < \gamma < \frac{2}{\sum_{i=1}^n Q_i^2 \rho\left(G_i G_i^\top\right)}. \tag{23}$$

In particular, the resulting optimal weight is $W_i = I$.

A gain γ can be estimated in a transparent way as follows:

Remark 1. *Condition (23) is equivalent to*

$$0 < \gamma \leq \frac{2}{Q_1^2\|G_1(z)\|_\infty^2 + Q_2^2\|G_2(z)\|_\infty^2 + \cdots + Q_n^2\|G_n(z)\|_\infty^2}, \tag{24}$$

where the (along-the-trial) transfer-function of subsystem i is $G_i(z) = C_i(zI - A_i)^{-1}B_i + D_i$. Each $\|G_i(z)\|_\infty^2$ can readily be obtained from frequency response data as the peak magnitude. Therefore γ can be chosen as the reciprocal of the summed subsystem squared peak magnitudes.

The simple update structure of $L_i = Q_i G_i^\top$ means that the lifted update form of (12) can be further simplified in an along-the-trial form as follows:

Theorem 4. *Update (12) with $L_i = Q_i G_i^\top$ can be implemented for samples $j = 0, \cdots, N-1$ as:*

$$u_{i,k+1}(j) = u_{i,k}(j) + \gamma Q_i \sum_{j=0}^{N-1-i} g_i(j) e(i+j), \tag{25}$$

where for causality we require $C_i A_i^{j-1} B_i = 0$, for $j \geq N$.

4.2. Generalised Inverse ILC

First recall the inverse ILC algorithm used by the existing collaborative approach of [9]. This is given by

$$L_i = Q_i G_i^{-1}, \tag{26}$$

where $Q_i = 1$ was assumed in [9]. This fits in the framework of Algorithm 1, but is only defined for the full rank case ($r = r^f$). Even more restrictive, it requires every subsystem to be full rank ($r_i = r_i^f$). However, we can transparently extend this algorithm so that it satisfies the conditions of Algorithm 1. Accordingly, introduce

$$L_i = Q_i^{-1} G_i^\dagger = (U_i O_i)^\dagger = O_i^\top (O_i O_i^\top)^{-1} (U_i^\top U_i)^{-1} U_i^\top, \tag{27}$$

where U_i, O_i comprises a full rank decomposition of $Q_i G_i$. To see this satisfies the conditions of Algorithm 1, we note that

$$G_i L_i = (U_i O_i)(O_i^\top (O_i O_i^\top)^{-1} (U_i^\top U_i)^{-1} U_i^\top) = U_i (U_i^\top U_i)^{-1} U_i^\top$$

is symmetric positive semi-definite.

Moreover, $\text{rank}(Q_i G_i L_i) = r_i$ which also satisfies the rank condition $\text{rank}(Q_i G_i L_i) = \text{rank}(Q_i G_i) = r_i$. It can be shown that the optimal weight in (11) is $W_i = Q_i^2 G_i^\top G_i$ for $r_i = r_i^f$, and $W_i = ((Q_i G_i)^\dagger ((Q_i G_i)^\dagger)^\top + V_{i,0} x_i V_{i,0}^\top)^{-1}$, $x_i \in \mathbb{R}^{r_i^f - r_i}$, $x_i \neq 0$ for $r_i < r_i^f$, where $V_{i,0}$ is the right singular vector of $Q_i P_i G_i$ corresponding to zero singular vectors.

Note that if G_i is full rank, equation (27) collapses to (26), since we can select $U_i = I, O_i = Q_i G_i$. Moreover, in all cases $\rho(Q_i G_i L_i) = 1$, and therefore convergence condition (14) is

$$0 < \gamma < \frac{2}{\sum_{i=1}^n Q_i \rho(G_i L_i)} = \frac{2}{\sum_{i=1}^n 1} = \frac{2}{n}, \tag{28}$$

which also satisfies (13). To examine convergence, we employ Theorem 2, in which the singular value matrix $D = \text{diag}(\Delta_r, 0_{r^f - r})$ is equal to

$$D = n \begin{bmatrix} I_r & 0 \\ 0 & 0_{r^f - r} \end{bmatrix}. \tag{29}$$

It follows that error evolution expression (20) gives

$$V_1^\top e_k = (I - \gamma n I_r)^k V_1^\top y_d = ((1 - \gamma n)^k I_r) V_1^\top y_d. \tag{30}$$

Therefore choosing $\gamma = \frac{1}{n}$ gives the fastest convergence speed with the error converging to its minimal level in one trial.

Remark 2. *In practice, the designer may not accurately know the number of subsystems, n, (e.g. in large or varying subsystem groups). In this situation, the designer must estimate an upper bound, n^*, which satisfies $\frac{n}{2} < n^*$. Then setting $\gamma = \frac{1}{n^*}$ guarantees convergence. In particular, error evolution (20) gives*

$$V_1^\top e_k = (I - \frac{n}{n^*} I_r)^k V_1^\top y_d = (1 - \frac{n}{n^*})^k I_r V_1^\top y_d, \tag{31}$$

so that convergence in one trial is achieved when $n = n^*$, and an over estimation n^* decreases the convergence rate.

4.3. Norm Optimal ILC

The NOILC algorithm of [21, 22] has received significant attention due to its attractive performance properties. It is centralised, and until now it has not been formulated for the case of rank deficient systems. In this section, we first recall the standard centralised NOILC algorithm. We extend the existing analysis to the case of non-full rank systems, to show the resulting convergence properties. Then a new decentralised NOILC algorithm is derived using the framework of Algorithm 1 and Theorem 1.

4.3.1. Centralised NOILC

Between every trial, the standard centralised NOILC update is computed as the solution to the optimisation

$$\arg\min_{u_{k+1}} \left\{ J(u_{k+1}) = \|e_{k+1}\| + \sum_{i=1}^{n} \|u_{i,k+1} - u_{i,k}\|^2_{R_i} \right\}. \tag{32}$$

The resulting update for each subsystem

$$u_{i,k+1} = u_{i,k} + R_i^{-1} G_i^\top (I + G R^{-1} G^\top)^{-1} e_k \tag{33}$$

is clearly of a centralised structure and its properties have previously been analysed assuming G_i is full rank and $Q_i = 1$. In this case, it was shown that the tracking task is achieved since $\lim_{k\to\infty} e_k = 0$.

Theorem 5. *Let ILC update (33) be applied to system (1) and (2), where no rank condition and arbitrary Q_i are assumed. Then the total system input and output converge as*

$$\lim_{k\to\infty} u_k = R^{-1} G^\top Q^\top (Q G R^{-1} G^\top Q^\top)^\dagger y^*,$$
$$\lim_{k\to\infty} y_k = P_{QG} y_d,$$

where $R = \mathrm{diag}(R_1, \cdots, R_n)$. This hence satisfies (10), (11) with $u_\infty = u^*$, $y_\infty = y^*$ and $W_i = Q_i^{-1} R_i$.

Proof. Let $X = QGR^{-1}G^\top Q^\top$, we note that

$$G_i L_i = Q_i R_i^{-1} G_i G_i^\top (I + X)^{-1}$$

is symmetric positive semi-definite. Hence we can apply singular value decomposition to $Q_i G_i L_i$ and, following a similar proof to that of Appendix A, the convergence condition is

$$|\rho((I + X)^{-1})| < 1 \Leftrightarrow \rho(X) > 0,$$

which holds for all rank conditions. Therefore, the system output converges to $y_\infty = P_{QG} y_d$ and the input converges to $u_{i,\infty} = Q_i R_i^{-1} G_i^\top (I + X)^{-1} (X(I + X)^{-1})^\dagger y_d$. As $(I + X)^{-1}$ is full rank square matrix, we have

$$u_{i,\infty} = Q_i R_i^{-1} G_i^\top X^\dagger y_d = Q_i R_i^{-1} G_i^\top X^\dagger y^*$$

and it follows that $W_i = Q_i^{-1} R_i$. ■

Theorem 5 shows for the first time that the standard centralised NOILC update can be applied to non-full rank systems, in which case it will still achieve minimum error (i.e. solve (10)) and will converge to an optimal input (i.e. solve (11)).

4.3.2. Sequential decentralised NOILC

A naive way to employ the existing NOILC update in a decentralised manner is to apply it to each subsystem in turn, while keeping the input of the other subsystems constant. It can be shown that such a strategy converges to the minimum error norm solution given by condition (10). However, this sequential form does not satisfy (7) in Definition 1, since not all subsystems are updated in the same trial. This makes it slow, and it also fails to achieve an optimal input energy norm since it does not fit the framework of Algorithm 1. It does, however, motivate a form of L_i that satisfies the decentralised framework, as defined next.

4.3.3. Decentralised NOILC

Inspired by the centralised form (33), we introduce the new decentralised ILC update form

$$\begin{aligned} L_i &= Q_i R_i^{-1} G_i^\top (I + Q_i^2 G_i R_i^{-1} G_i^\top)^{-1} \\ &= Q_i (Q_i^2 G_i^\top G_i + R_i)^{-1} G_i^\top. \end{aligned} \tag{34}$$

It then follows that

$$G_i L_i = Q_i G_i (Q_i^2 G_i^\top G_i + R_i)^{-1} G_i^\top \tag{35}$$

is symmetric positive semi-definite with rank r, so that the conditions of Algorithm 1 are satisfied.

Theorem 1 then holds, and in particular the optimal weight in (11) is $W_i = (R_i + Q_i^2 G_i^\top G_i)$. Therefore like its centralised counterpart, the new decentralised NOILC form (34) solves optimisations (10), (11) but using a different weight.

From Theorem 1 the convergence condition of decentralised NOILC is given by

$$0 < \gamma < 2 \Big/ \rho\left(\sum_{i=1}^{n} Q_i^2 G_i (Q_i^2 G_i^\top G_i + R_i)^{-1} G_i^\top \right) \tag{36}$$

with a sufficient condition from (14) of

$$0 < \gamma < 2 \Big/ \sum_{i=1}^{n} Q_i^2 \rho\left(G_i (Q_i^2 G_i^\top G_i + R_i)^{-1} G_i^\top \right). \tag{37}$$

5. Robustness Analysis

Theorem 2 provides conditions for rapid convergence, however there always exists a trade-off with robustness. In practice, there exists model uncertainty, non-repeated noise/disturbances and imprecise initial resetting. In this section robustness of ILC for MAS with model uncertainties and subsystem break down will be analysed. This facilitates comparison between the various algorithms proposed.

5.1. Robustness for Model Uncertainties

Previous ILC robustness has focused on SISO systems [23], but cannot be related transparently to MAS. Moreover, those assumptions limit the system dynamics, e.g. full-rank and/or no input-output delay, do not consider practical setup, and are also conservative. In this section, the generalised robustness analysis for MIMO MAS is given:

Theorem 6. *Let the true subsystem dynamics $\hat{G}_i$ be described by the multiplicative uncertainty model*

$$\hat{G}_i = U_i G_i, \tag{38}$$

where positive definite uncertainty U_i is a matrix of consistent dimension and G_i is the nominal model of the subsystem.

Application of Algorithm 1 to subsystem dynamics (38) achieves monotonic reduction in tracking error norm, i.e. relation (15), if the learning gain γ satisfies

$$\rho\left(I - \gamma \sum_{i=1}^{n} Q_i U_i G_i L_i \right) < 1. \tag{39}$$

Proof. From (4), (5), (12) and (38) it follows that

$$e_{k+1} = \left(I - \gamma \sum_{i=1}^{n} Q_i \hat{G}_i L_i\right) e_k. \tag{40}$$

Let $\sum_{i=1}^{n} Q_i \hat{G}_i L_i = P$, then we have

$$\begin{aligned}\|e_{k+1}\|^2 &= ((I - \gamma P)e_k)^\top (I - \gamma P)e_k \\ &= \|e_k\|^2 - \gamma e_k^\top (P^\top + P)e_k + \gamma^2 e_k^\top (P^\top P)e_k.\end{aligned}$$

This yields the difference $\|e_{k+1}\|^2 - \|e_k\|^2$ given by

$$\|e_{k+1}\|^2 - \|e_k\|^2 = -\gamma e_k^\top (P^\top + P)e_k + \gamma^2 e_k^\top (P^\top P)e_k.$$

Since U_i is positive-definite and $\gamma > 0$, so are the terms $\gamma e_k^\top (P^\top + P)e_k$ and $\gamma e_k^\top (P^\top + P)e_k$. For any non-zero e_k, the necessary and sufficient condition for $\|e_{k+1}\|^2 < \|e_k\|^2$ is

$$-\gamma e_k^\top (P^\top + P)e_k + \gamma^2 e_k^\top (P^\top P)e_k < 0. \tag{41}$$

Note that this inequality always holds for all sufficiently small $\gamma > 0$, since γ has a slower convergence rate to zero than term γ^2 when γ is sufficiently small. This proves the existence of $\gamma^* > 0$ and independence of e_k that ensures monotonic convergence. It then follows from (40), that

$$\rho\left(I - \gamma \sum_{i=1}^{n} Q_i \hat{G}_i L_i\right) < 1 \Leftrightarrow \rho\left(I - \gamma \sum_{i=1}^{n} Q_i U_i G_i L_i\right) < 1, \tag{42}$$

so that (42) yields the proposed inequality (39). ∎

Theorem 6 shows that Algorithm 1 is robust to the model uncertainty. Note that (38) is a common uncertainty form [21], but similar analysis can be performed using an additive form. Although U_i is undeterminable in general, it can often be bounded and these bounds used to ensure convergence.

Theorem 7. *Application of Algorithm 1 under the conditions of Theorem 6, results in error norm convergence to* $P^{\perp}_{QG} y_d$.

Proof. Introduce the singular value decomposition $\sum_i (Q_i \hat{G}_i L_i) = \hat{U}\hat{D}\hat{V}^\top$ where the columns of $\hat{U}$ and $\hat{V}$ are the (orthogonal) unitary left and right singular vectors of $\sum_i (Q_i \hat{G}_i L_i)$ and $\hat{D}$ contains its singular values, given by

$$\begin{aligned}\sum_i (Q_i \hat{G}_i L_i) &= \hat{U}\hat{D}\hat{V}^\top \\ &= \begin{bmatrix}\hat{U}_1 & \hat{U}_0\end{bmatrix} \begin{bmatrix}\hat{\Delta}_r & 0 \\ 0 & 0_{rf-r}\end{bmatrix} \begin{bmatrix}\hat{V}_1^\top \\ \hat{V}_0^\top\end{bmatrix} \\ &= \hat{U}_1 \hat{\Delta}_r \hat{V}_1^\top. \end{aligned} \tag{43}$$

Hence, we have

$$\begin{aligned}\left(I - \gamma \sum_{i=1}^{n} Q_i \hat{G}_i L_i\right) &= I - \gamma \hat{U}_1 \hat{\Delta}_r \hat{V}_1^\top = \hat{V}_1 \hat{V}_1^\top - \gamma \hat{U}_1 \hat{\Delta}_r \hat{V}_1^\top + \hat{V}_0 \hat{V}_0^\top \\ &= (\hat{V}_1 - \gamma \hat{U}_1 \hat{\Delta}_r)\hat{V}_1^\top + \hat{V}_0 \hat{V}_0^\top \end{aligned} \tag{44}$$

and based on (40), it follows that

$$e_k = \left((\hat{V}_1 - \gamma \hat{U}_1 \hat{\Delta}_r)\hat{V}_1^\top\right)^k y_d + \hat{V}_0 \hat{V}_0^\top y_d. \tag{45}$$

The first part of (45) converges to zero with increasing trials under the conditions of Theorem 6. But the second part, $\hat{V}_0 \hat{V}_0^\top y_d$, which appears while $\sum_{i=1}^{n} Q_i \hat{G}_i L_i$ is not full rank, will not change with increasing iterations. Hence, the tracking error will converge to $\hat{V}_0 \hat{V}_0^\top y_d$, which equals $P^{\perp}_{Q\hat{G}} y_d$. Based on the analysis of Theorem 1, this also equals $P^{\perp}_{QG} y_d$ as $\text{rank}(Q\hat{G}) = \text{rank}(QG)$. ∎

Theorem 7 shows that Algorithm 1 maintains the desired convergence despite model uncertainty.

5.2. Robustness to subsystem break down

In practice, any subsystem in the system may fail. The sudden removal of its contribution to collaborative output then affects the ILC updates of all other subsystems, potentially destabilising the system. In this section, the robustness for the total system after any subsystem break down is examined.

Theorem 8. *Let Algorithm 1 be applied to system (1) and (2) with an initial input for each subsystem given by* $u_{i,0} = 0$, $i = 1, 2, \cdots, n$. *If a subset,* $\mathcal{S} \subset \{1, 2, \cdots, n\}$, *of subsystems break down, then the learning gain must satisfy*

$$0 < \gamma < \frac{2}{\rho\left(\sum_{i=1, i\neq\mathcal{S}}^{n} Q_i G_i L_i\right)} \tag{46}$$

or more conservatively

$$0 < \gamma < \frac{2}{\sum_{i=1, i\neq\mathcal{S}}^{n} Q_i \rho\left(G_i L_i\right)} \tag{47}$$

in order to achieve monotonic reduction in tracking error norm for the remaining system.

Proof. Follows the proof of Theorem 1. ∎

Theorem 9. *Let* γ *be designed to satisfy (14) and therefore achieve monotonic reduction in tracking error norm within Algorithm 1. If any subset,* $\mathcal{S}$, *of subsystems subsequently breaks down, this* γ *will still ensure the remaining system achieves monotonic reduction in tracking error norm after the point of failure.*

Proof. Note that

$$\begin{aligned}&\sum_{i=1, i\neq\mathcal{S}}^{n} Q_i \rho\left(G_i L_i\right) < \sum_{i=1}^{n} Q_i \rho\left(G_i L_i\right) \\ \Leftrightarrow \quad &\frac{2}{\sum_{i=1}^{n} Q_i \rho\left(G_i L_i\right)} < \frac{2}{\sum_{i=1, i\neq\mathcal{S}}^{n} Q_i \rho\left(G_i L_i\right)}. \end{aligned} \tag{48}$$

Recall from Theorem 1 and Theorem 8, the γ satisfies

$$0 < \gamma < \frac{2}{\sum_{i=1}^{n} Q_i \rho\left(G_i L_i\right)} < \frac{2}{\sum_{i=1, i\neq\mathcal{S}}^{n} Q_i \rho\left(G_i L_i\right)} \tag{49}$$

and achieves monotonic reduction in tracking error norm. ∎

Theorem (9) shows that a system using the proposed decentralised ILC has a high robustness to subsystem break down. With a pre-defined learning gain γ satisfying Theorem 1, when an subsystem breaks down, the remaining system still converges

thereby avoiding any danger. Whether the system can obtain zero tracking error, depends on the rank condition of the remaining system $\sum_{i=1,i\neq\mathcal{S}}^{n} Q_i G_i L_i$. Following the proof of Theorem 1, if $\sum_{i=1,i\neq\mathcal{S}}^{n} Q_i G_i L_i$ is full rank, zero error tracking can be achieved. If not, the tracking error converges to $P_{\sum_{i=1,i\neq\mathcal{S}}^{n} Q_i G_i L_i} y_d$.

6. EXPERIMENTAL EVALUATION

The decentralised control framework is now applied to a key problem in upper limb stroke rehabilitation. Functional electrical stimulation (FES) involves artificially activating muscles and is often used to assist people with stroke in performing arm movements. FES involves placing electrodes over the muscle and applying a train of electrical pulses which innervate the underling nerves. When used to control the pulsewidth of the FES signal, ILC has successfully achieved precise motion during repeated attempts at movement tasks required for rehabilitation. This accuracy has translated into significantly improved outcome measures in several clinical trials [24, 18, 25].

To improve rehabilitation outcomes, FES must be applied to multiple muscles to enable natural, coordinated gestures. Comfort and safety are also increased by applying FES to more than one muscle actuating a given joint, since it reduces fatigue. To achieve this goal, there has been a recent drive in creating wearable FES devices that are incorporated in clothing to maximise convenient and usability. However, hardware constraints mean that an effective, affordable wearable system is only feasible if wiring and communication overheads between stimulation sites are minimised. This scenario exactly fits the collaborative set-up in this paper: each FES device is an subsystem, their inputs are the applied FES, and the collaborative output comprises the joint angle actuated by the muscles as they work together.

Wrist extension is a key problem in stroke rehabilitation and can be assisted by stimulating many of the 20 muscles in the forearm. To evaluate the system, four sites on the wrist will be stimulated to collaboratively extend the wrist joint. This corresponds to $n = 4$ subsystems. For ease of set-up, these sites will be selected from a 24-element electrode array, with each site taking the form of 2 adjoining electrodes in the array. The hardware used is described in [26], and consists of a tracking sensor, user interface software running on a laptop, a control unit, a 24 channel FES electrode array sleeve and FES electronics. The components are shown in Figure 2. The sensor (Kinect V2, Microsoft) is a wide-angle time-of-flight camera, which collects the positional data of the wrist, which is then processed by the user interface to generate angle data. This is sent to the control unit (Raspberry Pi 3 Model B+) via wireless transmission, which runs the real-time controller (at 40 Hz). The controller computes the voltage pulse train applied to each element of the 24 channel electrode array. Here, the frequency and amplitude of each pulse train are fixed, and the pulse width of each pulse train is the controlled variable (0 - 300 μs).

A wide variety of models exist of the moment generated by stimulated muscle dynamics, with the most popular structure

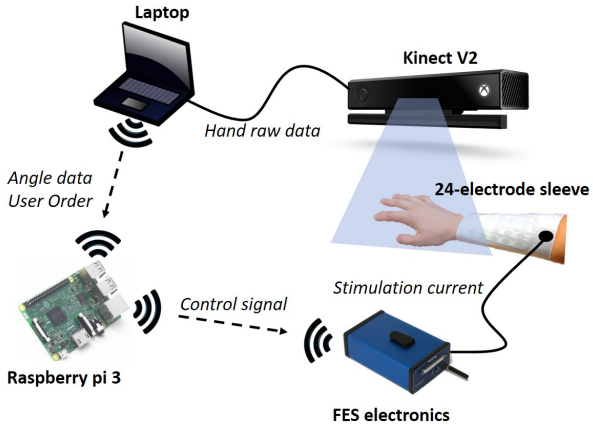

Figure 2: Upper limb stroke rehabilitation hardware

comprising a critically damped form [27, 28],

$$G_i = \frac{w_i^2}{s^2 + 2\zeta_i w_i s + w_i^2}. \tag{50}$$

These moments then sum up to generate movement about the wrist, so that the arm dynamics takes the overall form of (1), (2), as verified in [29]. Each subsystem therefore takes the form (50), but with differing w_i, ζ_i. Elbow angle reference y_d is selected to mimic a natural wrist extension motion including extension (0-3s) and 'holding' (3-6s) phases.

Tests were conducted in a study with 4 unimpaired participants (Ethics No. 23824.A7). The electrode array was first positioned on their forearm, and the voltage amplitude was set by applying a 300 μs FES signal to the two adjoining elements and increasing the amplitude until the comfortable limit was reached. Ramp profiles were then applied to each pair of elements in turn, and the resulting joint angles recorded. The 4 subsystems were selected as those elements which produced maximum wrist extension. The model of each subsystem was fitted using input-output data.

The design framework of Section 3 was evaluated by applying Algorithm 1 for the three ILC update types introduced in Section 4. First the gradient ILC update was applied, with a value of $\gamma = 0.000025$ chosen to satisfy (23). The tracking results for all participants (P1,$\cdots$,P4) are shown in Table 1 including error norm and input energy values. Representative tracking outputs are shown in Figure 3. The figure shows that the system output converges to the reference output gradually. The gap between system output and the reference is quite large for the first iteration, but reduces to a small level after several trials.

Then inverse ILC was applied with $\gamma = 0.1$ chosen to satisfy (28). The tracking results for all participants are shown in Table 1. Representative tracking outputs are shown in Figure 4. The figure shows that the convergence speed of the system output is very fast, however, some oscillations appear in tracking output, especially in the extension phase.

Then decentralised NOILC was applied, as discussed in Section 4.3. A weight of $R_i = 50,\ i = 1,\cdots,4$ was chosen to sat-

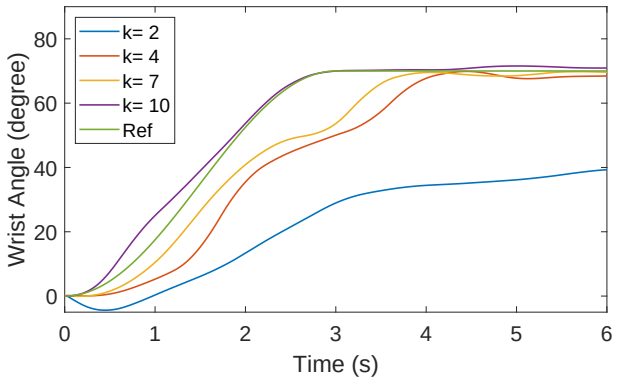

Figure 3: Tracking output using gradient ILC framework for P1

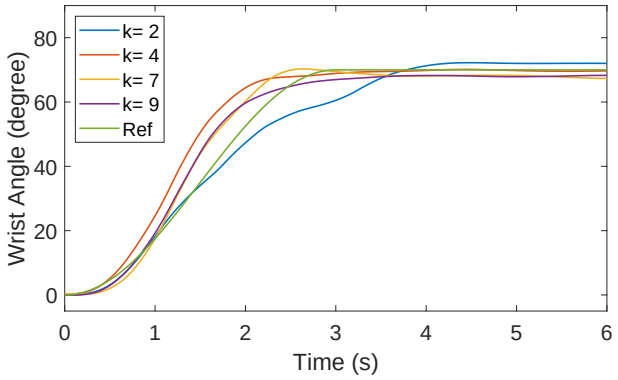

Figure 4: Tracking output using inverse ILC framework for P4

isfy (37) with γ values of between 0.34 and 0.42. The tracking error results for all participants are shown in Table 1. Representative tracking outputs with $\gamma = 0.39$ are shown in Figure 5. The figure shows that system output tracks the reference smoothly and converges rapidly, providing natural wrist extension.

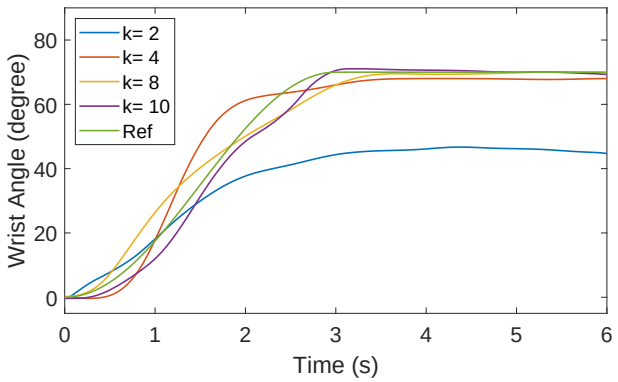

Figure 5: Tracking output using decentralised NOILC framework for P2

The convergence results and system total input energy of these three different decentralised control frameworks are shown in Figure 6 and 7 for P3. The results for other participants are shown in Table 1. Figure 6 shows that the three decentralised ILC updates each achieve a low error for wrist extension. They also show that the fastest convergence rate is obtained by the decentralised inverse ILC method and the slowest rate is obtained by the decentralised gradient ILC method. This verifies the results of Theorem 3. The algorithm converges to similar converged input signals and error values, thereby verifying Theorem 1 and Theorem 7. Zero error is not possible due to sensor error, output and input restriction, system nonlinearity, muscle fatigue and spasticity, modeling error, etc. Figure 7 shows the total input energy of systems using the three different decentralised control frameworks. The input norm closely match theoretical values given by (11). Note that, since each update has a different weight W_i, these are not the same, however they only vary by a small range (0.5%) from one another. The similarity in weighted input norm is due to the dynamics G_i which are relatively flat over the system bandwidth.

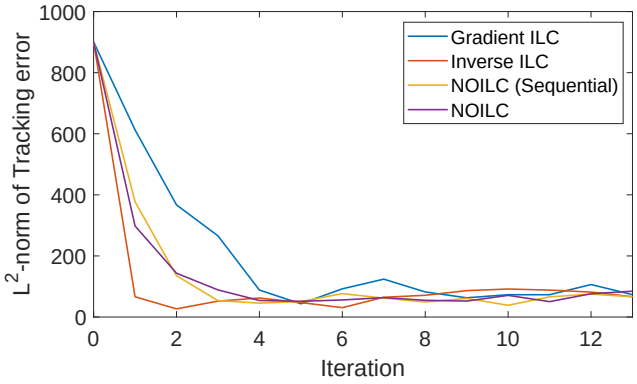

Figure 6: Convergence results using three different ILC methods for P3

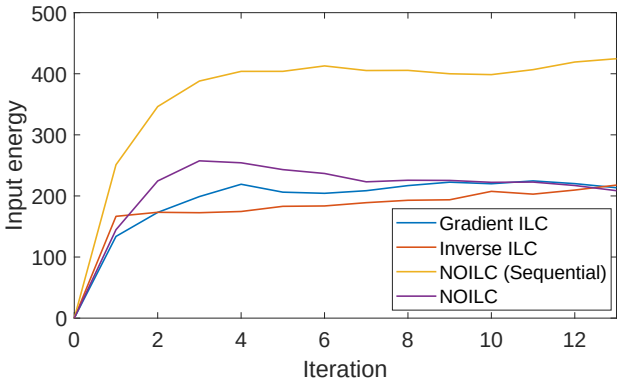

Figure 7: Convergence results using three different ILC methods for P3

To compare the framework to traditional ILC, we have implemented sequential NOILC which was proposed in Section C.2. Results for P3 with $R_i = 50$ are shown in Figure 6 and 7. It is clear that the sequential framework consumes more energy than the optimal decentralised frameworks. In practice this manifests in discomfort for the participant.

As the first application of ILC to FES wrist extension dynamics, these results confirm the high accuracy that is possible. They also confirm the properties of the decentralised framework. The merits of each algorithm enable the designer to select the most appropriate for an application, while benefitting from the shared properties of the decentralised framework.

Table 1: Testing results for 4 participants

No.	Method	Parameters	Results			
			$\|\|e_1\|\|^1$	$\|\|e\|\|_m{}^2$	$\|\|e\|\|_H{}^3$	$\|\|u\|\|_T{}^4$
P1	Gradient	γ =2.5e-05	507.3	47.2	11.9	141.3
	Inverse	γ =0.1	314.1	56.8	9.6	224.4
	NOILC	γ =0.34 R_i = 50	117.9	20.4	3.5	183.7
P2	Gradient	γ =2.5e-05	636.6	66.4	30.5	251.2
	Inverse	γ =0.1	125.0	33.2	2.8	227.5
	NOILC	γ =0.39 R_i = 50	92.5	32.6	9.4	241.7
P3	Gradient	γ =2.5e-05	612.6	43.3	15.4	213.4
	Inverse	γ =0.1	66.3	26.6	9.3	217.8
	NOILC	γ =0.36 R_i = 50	297.9	50.4	7.7	208.6
P4	Gradient	γ =2.5e-05	301.0	65.4	13.4	261.3
	Inverse	γ =0.1	70.2	41.3	4.5	274.1
	NOILC	γ =0.42 R_i = 50	271.9	40.6	11.8	280.3

[1] $\|\|e_1\|\|$: Tracking error norm for trial 1.
[2] $\|\|e\|\|_m$: Minimum tracking error norm of task.
[3] $\|\|e\|\|_H$: Minimum error norm in holding phase of task.
[4] $\|\|u\|\|_T$: Total input energy on the final iteration.

7. CONCLUSIONS AND FUTURE WORK

This paper developed a general decentralised ILC design framework for the collaborative tracking problem in which an arbitrary number of subsystems work together to achieve a global objective. The proposed design framework enables a wide range of new ILC algorithms to be formulated for MAS. These permit convergence speed to be traded with robustness to model uncertainty for the first time. To exemplify the proposed design framework, three novel decentralised ILC algorithm are developed. The proposed algorithm achieves optimal convergence of both input and output signals in a transparent manner.

The performance of the proposed algorithm is verified via application to stroke rehabilitation. Results illustrate decentralised control design in the first application to the wrist. This also constitutes the first application of ILC to a non-full rank MAS. The use of subsystems is particularly valuable in this application since they allow muscles to be stimulated without sharing information. It is therefore an important step for development of distributed wearable rehabilitation technology.

As well as providing a new framework for decentralised ILC, the results also extend ILC research for use with rank deficient dynamics. Future work will embed system constraints, e.g. input saturation, which exist widely in practice.

Appendix A. Proof of Theorem 1

To prove Theorem 1 we require the following results:

Proposition 1. *For any set of symmetric matrices $\{A_i\}$ of equal dimension*

$$\overline{\lambda}(\sum A_i) = \|\sum A_i\|_2 \leq \sum_i \|A_i\|_2 = \sum_i \overline{\lambda}(A_i)\ . \tag{A.1}$$

Proof. For a symmetric A we have $A^\top = A$, and $\overline{\lambda}(A^\top A) = \overline{\lambda}(A^2) = (\overline{\lambda}(A))^2$. Hence $\|A\|_2 = \sqrt{\overline{\lambda}(A^\top A)} = \sqrt{\overline{\lambda}(A^H A)} = \overline{\lambda}(A)$. For symmetrical matrices A, C, $\overline{\lambda}(A+C) = \|A+C\|_2 \leq \|A\|_2 + \|C\|_2 = \overline{\lambda}(A) + \overline{\lambda}(C)$ which extends to (A.1). ■

Proposition 2. *For positive semi-definite matrix A, B, X, have:*

$$1).\mathscr{C}(AB) \subseteq \mathscr{C}(A), \quad 2).\mathscr{C}(XX^\top) = \mathscr{C}(X).$$

Proof. 1). Since the ith column of AB, denoted as $(AB)_i$, is equal to $A(B)_i$. Thus, each column of AB is just a linear combination of columns of A. So, $\mathscr{C}(AB) \subseteq \mathscr{C}(A)$ holds.

2). First, we prove that $\mathscr{C}(XX^\top) \subseteq \mathscr{C}(X)$. Since the ith column of $XX^\top$, denoted as $(XX^\top)_i$ equals $X(X^\top)_i$. Thus, each column of $XX^\top$ is just a linear combination of columns of X. So, $\mathscr{C}(XX^\top) \subseteq \mathscr{C}(X)$ holds.

Now, we prove that $\mathscr{C}(X) \subseteq \mathscr{C}(XX^\top)$. Let $X = UDV'$ be the singular value decomposition of X. It follows that

$$XX^\top = UDV'V'^\top DU^\top = UD^2U^\top, \tag{A.2}$$

note that

$$XX^\top UD^{-1}V' = UD^2U^\top UD^{-1}V' = UDV' = X. \tag{A.3}$$

Therefore, the ith column of X, denoted as $(X)_i$, satisfies:

$$(X)_i = XX^\top (UD^{-1}V')_i. \tag{A.4}$$

We know each column of X is still a linear combination of columns of $XX^\top$, which shows $\mathscr{C}(X) \subseteq \mathscr{C}(XX^\top)$.

Both $\mathscr{C}(XX^\top) \subseteq \mathscr{C}(X)$ and $\mathscr{C}(X) \subseteq \mathscr{C}(XX^\top)$ hold, and hence $\mathscr{C}(XX^\top) = \mathscr{C}(X)$ holds. Equivalently, we have $\mathscr{N}(XX^\top) = \mathscr{N}(X^\top)$. ■

Proposition 3. *For $i = 1, \cdots n$, let G_i be a positive semi-definite square matrix, and let C_i be any symmetric positive definite matrix of consistent dimension. Then every $x \in \mathscr{N}(GCG^\top)$ satisfies $C_iG_i^\top x_i = 0$, where $G = [G_1, G_2, \cdots, G_n]$ and $C = \mathrm{diag}(C_1, C_2, \cdots, C_n)$.*

Proof. Since C is symmetric positive semi-definite, then it has a spectral decomposition:

$$C = \sum_i \lambda_i x_i x_i^\top. \tag{A.5}$$

Define $y_i = \sqrt{\lambda_i}x_i$. This definition is possible because λ_i are non-negative. Then,

$$C = \sum_i \lambda_i y_i y_i^\top. \tag{A.6}$$

Define L to be the matrix whose columns are y_i. It follows that there exists a matrix L satisfying $C = LL^\top$. From this construction, columns of L are orthogonal. Then based on Proposition 2, we have

$$\mathscr{C}(GCG^\top) = \mathscr{C}(GLL^\top G^\top) \subseteq \mathscr{C}(GLL^\top) \tag{A.7}$$

and we know $\mathrm{rank}(GL) = \mathrm{rank}(GL(GL)^\top)$, it follows that

$$\mathscr{C}(GCG^\top) = \mathscr{C}(GL(GL)^\top) = \mathscr{C}(GL) \supseteq \mathscr{C}(GLL^\top). \tag{A.8}$$

Hence, with (A.7) and (A.8) we have

$$\mathscr{C}(GCG^\top) = \mathscr{C}(GLL^\top) = \mathscr{C}(GC) \Leftrightarrow \mathscr{N}(GCG^\top) = \mathscr{N}(CG^\top). \tag{A.9}$$

We know for any matrices of form $CG^\top$ that

$$\mathcal{N}(CG^\top) = \mathcal{N}(C_1G_1^\top) \cap \cdots \cap \mathcal{N}(C_nG_n^\top).$$

So we have

$$\mathcal{N}(GCG^\top) \subseteq \mathcal{N}(C_iG_i^\top),\ i = 1, \cdots n.$$

■

Proposition 4. *Let C be any symmetric positive definite matrix of consistent dimension. When rank$(GCG^\top)$ = rank(G), then for matrix $GCG^\top$ we have $\mathscr{C}(GCG^\top) = \mathscr{C}(G)$.*

Proof. Based on Proposition 2, we have

$$\mathscr{C}(GCG^\top) \subseteq \mathscr{C}(G).$$

Let rank$(GCG^\top) = r$ and $\{a_1, a_2, \cdots, a_r\}$ be a basis for $\mathscr{C}(GCG^\top)$, so for some b_i we have

$$a_i = GCG^\top b_i = G(CG^\top b_i),\ i = 1, 2, \cdots, n. \tag{A.10}$$

So $\{a_1, a_2, \cdots, a_r\} \subseteq$ rank(G). As we have

$$\text{rank}(G) = \text{rank}(GCG^\top) = r,$$

this means these vectors also must be a basis for $\mathscr{C}(G)$. Hence, this shows $\mathscr{C}(GCG^\top) = \mathscr{C}(G)$. ■

Using Propositions 1 - 4, Theorem 1 can now be proved.

Proof. From (5), (4), (12) we have

$$e_{k+1} = \left(I - \gamma \sum_{i=1}^{n} q_i G_i L_i\right) e_k = (I - \gamma QGL)\, e_k \tag{A.11}$$

and since each G_iL_i is symmetric positive semi-definite, so is QGL. Introduce the singular value decomposition $QGL = UDV^\top$, where the columns of U and V are the (orthogonal) unitary left and right singular vectors of QGL and D contains its singular value $\lambda_1, \cdots, \lambda_N$. When no rank condition is assumed, we can expand QGL as

$$QGL = \begin{bmatrix} U_1 & U_0 \end{bmatrix} \begin{bmatrix} \Delta_r & 0 \\ 0 & 0_{r^f - r} \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_0^\top \end{bmatrix} = U_1 \Delta_r V_1^\top, \tag{A.12}$$

where $\Delta_r = \text{diag}(\lambda_1, \cdots, \lambda_r)$. As QGL is symmetric positive semi-definite, we have $U_1 = V_1$, so (A.12) can be rewritten as

$$QGL = V_1 \Delta_r V_1^\top. \tag{A.13}$$

Therefore

$$\left(I - \gamma QGL\right) = V_1(I - \gamma\Delta_r)V_1^\top + V_0V_0^\top. \tag{A.14}$$

Then (A.11) can be expressed as

$$e_{k+1} = (V_1(I - \gamma\Delta_r)V_1^\top + V_0V_0^\top)e_k. \tag{A.15}$$

We note that

$$\begin{aligned}&(V_1(I - \gamma\Delta_r)V_1^\top + V_0V_0^\top)^2 = \\ &V_1(I - \gamma\Delta_r)^2V_1^\top + V_0V_0^\top + V_1(I - \gamma\Delta_r)^\top V_1^\top V_0V_0^\top \\ &+ V_0V_0^\top V_1(I - \gamma\Delta_r)V_1^\top.\end{aligned} \tag{A.16}$$

As V are a set of orthonormal eigenvectors, so $V_0^\top V_1 = 0_{(r^f - r)\times r}$ and $V_1^\top V_0 = 0_{r\times(r^f - r)}$, then we can write

$$(V_1(I - \gamma\Delta_r)V_1^\top + V_0V_0^\top)^2 = V_1(I - \gamma\Delta_r)^2V_1^\top + V_0V_0^\top.$$

Similarly,

$$(V_1(I - \gamma\Delta_r)V_1^\top + V_0V_0^\top)^k = V_1(I - \gamma\Delta_r)^kV_1^\top + V_0V_0^\top.$$

Hence the error converges to

$$e_\infty = \lim_{k\to\infty} e_k = \lim_{k\to\infty} V_1(I - \gamma\Delta_r)^kV_1^\top y_d + V_0V_0^\top y_d. \tag{A.17}$$

We also note that, as V are a set of orthonormal eigenvectors, hence we have $V_1^\top V_1 = I_{r\times r}$. Left multiply $V_1^\top$ on both sides of (A.17), to produce

$$V_1^\top e_\infty = \lim_{k\to\infty}(I - \gamma\Delta_r)^kV_1^\top y_d. \tag{A.18}$$

Hence a necessary and sufficient condition for convergence to the limit $V_0V_0^\top y_d$ is

$$|\rho(I - \gamma\Delta_r)| < 1 \Leftrightarrow 0 < \overline{\lambda}(\gamma\Delta_r) < 2 \Leftrightarrow 0 < \gamma < \frac{2}{\overline{\lambda}(\Delta_r)}. \tag{A.19}$$

Note that $\overline{\lambda}(\Delta_r) = \overline{\sigma}(\Delta_r) = \rho\left(\sum_{i=1}^n Q_iG_iL_i\right)$. Applying Proposition 1 means $\rho\left(\sum_{i=1}^n Q_iG_iL_i\right) \le \sum_{i=1}^n \rho\left(Q_iG_iL_i\right)$, and a sufficient condition for (A.19) is

$$0 < \gamma < \frac{2}{\sum_{i=1}^n Q_i\rho\left(G_iL_i\right)}. \tag{A.20}$$

With γ satisfying (A.20), the final error is

$$e_\infty = V_0V_0^\top y_d = (I - P_{QGL})y_d. \tag{A.21}$$

Since $\sum_i(Q_iG_iL_i)$ is symmetric semi-positive definite with rank(QGL) = rank(QG), then it has form $QGCG^\top Q^\top$ where C is positive definite. Hence by Proposition 4, the eigenvectors of $QGCG^\top Q^\top$ corresponding to zero diagonal entries of D also form an orthogonal basis for $\mathcal{N}((QG)^\top) = \mathscr{C}(QG)^\perp$. Therefore, the column space of QGL is equal to that of QG, i.e. $P_{QGL} = P_{QG}$. Then (A.21) can be simplified as

$$e_\infty = (I - P_{QG})y_d = P_{QG}^\perp y_d \tag{A.22}$$

and therefore the system output converges to

$$y_\infty = y_d - e_\infty = y_d - P_{QG}^\perp y_d = P_{QG}y_d. \tag{A.23}$$

This means that the system output y_∞ is always a unique minimiser, which is independent of the learning algorithm under the conditions of Theorem 1. Different learning algorithms, however, affect the output convergence speed. Hence $y^* = y_\infty$, which satisfies (10).

Now consider the system input, starting from the expression (A.15). Repeated application of (12) and (14) gives

$$\begin{aligned}u_{i,k+1} &= u_{i,k} + \gamma L_i((V_1(I - \gamma\Delta_r)^kV_1^\top) + V_0V_0^\top)y_d \\ &= \gamma L_i(V_1\sum_{p=0}^{k}(I - \gamma\Delta_r)^kV_1^\top)y_d + \gamma kL_i(I - P_{QGL})y_d.\end{aligned}$$

So that applying Proposition 3 we obtain $L_i(I-P_{QGL}) = L_i P^{\perp}_{QGL}$ $L_i P^{\perp}_{Q_i L_i} = 0$, then

$$\begin{aligned} u_{i,\infty} &= \lim_{k\to\infty} \gamma L_i (V_1 \sum_{p=0}^{k} (I-\gamma\Delta_r)^k V_1^{\top}) y_d \\ &= \gamma L_i V_1 (I-(I-\gamma\Delta_r))^{-1} V_1^{\top} y_d \\ &= L_i V_1 \Delta_r^{-1} V_1^{\top} y_d = L_i V_1 \Delta_r^{-1} U_1^{\top} y_d. \end{aligned} \tag{A.24}$$

Then, we can obtain relation

$$u_{i,\infty} = L_i (\sum_i (Q_i G_i L_i))^{\dagger} y_d \tag{A.25}$$

or, equivalently,

$$u_{\infty} = L(QGL)^{\dagger} y_d. \tag{A.26}$$

We next show this is the solution to (11). From Theorem 1 conditions, with no loss of generality, introduce L_i with form $L_i = Q_i W_i^{-1} G_i^{\top}$, so that

$$L = W^{-1} G^{\top} Q^{\top} \tag{A.27}$$

where $W = \mathrm{diag}(W_1, \cdots, W_n)$. Then (A.26) can be written as

$$u_{\infty} = W^{-1} G^{\top} Q^{\top} (QGW^{-1} G^{\top} Q^{\top})^{\dagger} y_d. \tag{A.28}$$

The term $W^{-1} G^{\top} Q^{\top} (QGW^{-1} G^{\top} Q^{\top})^{\dagger}$ is a generalised inverse $(QG)^{\dagger}$ of QG, with the property $(QG)^{\dagger} P_{QG} = (QG)^{\dagger}$. It follows that

$$\begin{aligned} u_{\infty} &= W^{-1} G^{\top} Q^{\top} (QGW^{-1} G^{\top} Q^{\top})^{\dagger} y_d \\ &= W^{-1} G^{\top} Q^{\top} (QGW^{-1} G^{\top} Q^{\top})^{\dagger} y^*. \end{aligned} \tag{A.29}$$

This is the solution u^* to

$$\min_u \|u\|_W, \quad s.t \;\; QGu = y^* \tag{A.30}$$

which hence solves (11). ■

References

[1] J. Eriksson, M. J. Mataric, C. Winstein, Hands-off assistive robotics for post-stroke arm rehabilitation, in: IEEE International Conference on Rehabilitation Robotics (ICORR'05), 2005, pp. 21–24.

[2] M. A. Goodrich, A. C. Schultz, Human-robot interaction: a survey, Foundations and trends in human-computer interaction 1 (3) (2007) 203–275.

[3] S. J. Schroeck, W. C.Messner, R. J. McNab, On compensatordesign for linear time-invariant dual-input single-output systems, IEEE/ASME Transactions on Mechatronics 6 (1) (2001) 50–57.

[4] H. Kawasaki, S. Ueki, S. Ito, Decentralized adaptive coordinated control of multiple robot arms without using a force sensor, Automatica 42 (3) (2006) 481–488.

[5] D. Meng, K. L. Moore, Learning to cooperate: Networks of formation agents with switching topologies, Automatica 64 (2016) 278–293.

[6] K. L. Moore, Iterative Learning Control for Deterministic Systems. Advances in Industrial Control., Springer-Verlag, 1993.

[7] D. A. Bristow, M. Tharayil, A. G. Alleyne, A survey of iterative learning control: A learning-based method for high-performance tracking control, IEEE Control Systems Magazine 26 (3) (2006) 96–114.

[8] D. H. Owens, Iterative Learning Control: An Optimization Paradigm, Springer, 2015.

[9] S. Devasia, Iterative learning control with time-partitioned update for collaborative output tracking, Automatica 69 (2016) 258–264.

[10] T. J. Harte, J. Hätönen, D. H. Owens, Discrete-time inverse model-based iterative learning control: stability, monotonicity and robustness, International Journal of Control 78 (8) (2005) 577–586.

[11] H. Ahn, K. L. Moore, Y. Chen, Iterative learning control: robustness and monotonic convergence for interval systems, Springer Science & Business Media, 2007.

[12] T. D. Son, G. Pipeleers, J. Swevers, Robust monotonic convergent iterative learning control, IEEE Transactions on Automatic Control 61 (4) (2016) 1063–1068.

[13] J. Realmuto, R. B. Warrier, S. Devasia, Iterative learning control for human-robot collaborative output tracking, in: 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), 2016, pp. 1–6.

[14] J. Realmuto, R. B. Warrier, S. Devasia, Data-inferred personalized human-robot models for iterative collaborative output tracking, J Intell Robot Syst 91 (2018) 137–153.

[15] T. Dinh Van, C. Freeman, P. Lewin, Assessment of gradient-based iterative learning controllers using a multivariable test facility with varying interaction, Control Engineering Practice 29 (2014) 158–173.

[16] D. Shen, C. Liu, L. Wang, X. Yu, Iterative learning tracking for multisensorsystems: A weighted optimization approach, IEEE Transactions on Cybernetics (2019) 1–14.

[17] S. Chen, C. T. Freeman, B. Chu, Gradient-based iterative learning control for decentralised collaborative tracking, in: IEEE European Control Conference, 2018, pp. 721–726.

[18] C. T. Freeman, E. Rogers, A. Hughes, J. H. Burridge, K. L. Meadmore, Iterative learning control in health care: Electrical stimulation and robotic-assisted upper-limb stroke rehabilitation, IEEE Control Systems Magazine 32 (1) (2012) 18–43.

[19] J. J. Hatonen, C. T. Freeman, D. H. Owens, P. L. Lewin, E. Rogers, A gradient-based repetitive control algorithm combining ILC and pole placement, European Journal of Control 12 (3) (2006) 278–292.

[20] H. S. Ahn, Y. Chen, K. L. Moore, Iterative learning control: Brief survey and categorization, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37 (6) (2007) 1099–1121.

[21] N. Amann, D. H. Owens, E. Rogers, Iterative learning control for discrete-time systems with exponential rate of convergence, IEE Proceedings-Control Theory and Applications 143 (2) (1996) 217–224.

[22] K. L. Barton, A. G. Alleyne, A norm optimal approach to time-varying ILC with application to a multi-axis robotic testbed, IEEE Transactions on Control Systems Technology 19 (1) (2011) 166–180.

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

[24] C. T. Freeman, A. Hughes, J. H. Burridge, P. H. Chappell, P. L. Lewin, A robotic workstation for stroke rehabilitation of the upper extremity using fes, Medical engineering & physics 31 (3) (2009) 364–373.

[25] C. T. Freeman, E. Rogers, J. H. Burridge, A. Hughes, K. L. Meadmore, Iterative learning control for electrical stimulation and stroke rehabilitation, Springer, 2015.

[26] K. Yang, K. Meadmore, C. T. Freeman, N. Grabham, A. Hughes, Y. Wei, R. Torah, M. Glanc-Gostkiewicz, S. Beeby, J. Tudor, Development of user-friendly wearable electronic textiles for healthcare applications, Sensors 18 (8) (2018) 2410.

[27] W. K. Durfee, K. E. MACLean, Methods for estimating isometric recruitment curves of electrically stimulated muscle, IEEE Transactions on Biomedical Engineering 36 (7) (1989) 654–667.

[28] P. Veltink, H. Chizeck, P. Crago, A. El-Bialy, Nonlinear joint angle control for artificially stimulated muscle, IEEE Transactions on biomedical engineering 39 (4) (1992) 368–380.

[29] E. Copur, C. Freeman, B. Chu, D. Laila, Repetitive control of electrical stimulation for tremor suppression, IEEE Transactions on Control Systems Technology 27 (2) (2017) 540–552.