2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Cooperative Learning Control for Multi Input Multi Output Systems

Rangana N. Jayawardhana[1] and Bijoy K. Ghosh[1]

***Abstract*— A new *Cooperative Learning Control* problem is introduced where a dynamical system is controlled by the sum of two controllers. Each of the two controllers, we design, has the structure of an iterative learning controller, which learns to track a desired, a priori chosen, output sequence. Once learned, the strength of one of the controller is reduced while this loss of control is iteratively transfered to the other controller. There is no direct communication between the two controllers and each controller updates iteratively, using the error signal between the system and the desired output. The point of this paper is to show that 'controller participation' can be iteratively transferred until one controller has completely acquired full control of the closed loop system. An important application of the proposed cooperative control system is in Rehabilitation of stroke patients, wherein a loss of control in the arm movement is initially aided by additive control signals from a computer. Subsequently, with therapeutic recovery, dependence on the computer control is reduced while the patient learns to be self reliant on his/her own motor control capability. The proposed cooperative control is illustrated using a non-square linear dynamical system in discrete time and we also consider a 2-link arm as an additional example of a nonlinear system.**

I. INTRODUCTION

In this paper we consider a pair of iterative learning controllers (ILCs) and connect them additively, so that the learnt control input to a dynamical system is the sum of the signals from each of the two ILCs. We assume that the ILCs do not communicate between each other and each ILC has access to the output of the dynamical system produced by the combined sum of the control signals generated by the two ILCs (see Fig. 1). The iterations of the two ILCs separately converge, depending only on their respective initial conditions. The main point of this paper is to show that when the two ILCs share a suitably chosen gain matrix proportionately and if the initial condition of the first ILC is lowered from its earlier converged value, while keeping the initial condition of the other ILC unchanged, the two ILCs converge once again with the following property. The converged control from the first ILC is slightly lowered whereas from the second ILC is slightly elevated. In the limit as this process of lowering the initial condition is repeated, the control signal sequences generated from the first ILC approach zero and we say that the second ILC has acquired full control.

Iterative learning control has been used in a number of areas of research since its inception [1], [2], [3], [4], [5]. In particular, in the area of multi agent system coordination [6], [7]. Moreover, cooperation of multi agent systems have been studied separately [8], [9]. Whereas multi-agents are independently controlled to maintain, for example a formation or shape, the illustration in this paper is about one dynamical system being controlled by two controllers cooperating to fulfill an objective of tracking. Additionally, one controller is handing off the control to the other, iteratively, without an explicit communication between the two.

Recently, iterative learning controllers have been used in stroke rehabilitation of the human upper limb [10], [11]. A stroke patient undergoes an iterative learning process similar to that of a child learning to execute a task for the first time, where it is assumed that the patient will improve upon the task execution through repeated trials. In the framework of [10], it had been assumed that the patient's own motor controller cannot learn to perform the task on its own. An additive control signal, generated by a computer, is added on to what the motor signals of the patient had been generating. Therapeutically, it was shown that the patient made progress and dependence on the computer is gradually weaned off.

In the cooperative control problem introduced here, we propose that both the motor controller (MC) and the computerized controller (CC) are Iterative Learning Controllers (see Fig. 1). This assumption is only for the purpose of illustration of the main result in this paper. We do not claim that the human motor controller mimics an ILC. We emphasize that the CC is there to assist the learning process, so that the combined effort of both the MC and CC is adequate to generate the required tracking control to the arm. Initially CC will assist the patient to make the required arm movement. During the subsequent sessions, the CC input is reduced allowing the MC to catch up to the slack and improve upon what it had already learned in the previous session.

The ILC formulation, considered here, was introduced in [12]. In Section II, we summarize the main content of the observer based iterative learning control algorithm. In Section III we formulate the cooperative learning controller, and discuss the convergence results that guarantee transfer of control from one controller to the other. In Section IV we show simulation results by considering two examples where a 3-input 2-output discrete time system and a linearized 2-link arm are controlled in cooperation. We conclude our findings in Section V.

II. BACKGROUND

We start with a $m \times p$ linear time invariant, dynamical system of degree n in discrete time given by

$$x_{k+1} = Ax_k + Bw_k, \quad y_k = Cx_k, \tag{1}$$

*Partially supported by Dick & Martha professorship at Texas Tech University.

[1]Rangana N. Jayawardhana and Bijoy K. Ghosh are with Department of Mathematics and Statistics, Texas Tech University, Broadway and Boston, Lubbock, TX 79409-1042, USA. rangana.jayawardhana@ttu.edu; bijoy.ghosh@ttu.edu

where $x_0 = 0$ and $k \in \mathbb{N}$ is the discrete time index, $x_k \in \mathbb{R}^n$, $y_k \in \mathbb{R}^m$ and $w_k \in \mathbb{R}^p$. For an a priori chosen positive integer N, we define an input sequence vector

$$\mathcal{W} \ = \ (w_0^T, w_1^T, w_2^T, \cdots, w_{N-1}^T)^T,$$

and a corresponding output sequence vector

$$\mathcal{Y} \ = \ (y_1^T, y_2^T, y_3^T, \cdots, y_N^T)^T,$$

where, $\mathcal{W} \in \mathbb{R}^{pN}$ and $\mathcal{Y} \in \mathbb{R}^{mN}$. Using (1) we write

$$\mathcal{Y} \ = \ \mathbf{P}\,\mathcal{W},$$

where

$$\mathbf{P} = \begin{pmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CB \end{pmatrix}. \quad (2)$$

Note that the matrix $\mathbf{P}$ has N^2 number of blocks of $m \times p$ matrices. Therefore $\mathbf{P} \in \mathcal{M}^{mN \times pN}$ where, for all positive integers p, q; $\mathcal{M}^{p \times q}$ is defined to be the set of all p by q matrices with real elements. Throughout this paper we make the following assumption.

Assumption 2.1: The matrix $\mathbf{P}$ in (2) is of full-rank.

Let us now assume that we have an a priori given desired sequence of output vectors, that we would like to track

$$\mathcal{Y}^* \ = \ (y_1^{*T}, y_2^{*T}, y_3^{*T}, \cdots, y_N^{*T})^T, \qquad \mathcal{Y}^* \in \mathbb{R}^{mN}.$$

Our goal is to find a $\mathcal{W}^*$ where,

$$\mathcal{W}^* \ = \ (w_0^{*T}, w_1^{*T}, w_2^{*T}, \cdots, w_{N-1}^{*T})^T, \qquad \mathcal{W}^* \in \mathbb{R}^{pN};$$

such that $\mathcal{Y}^*$ is arbitrary close to $\mathbf{P}\mathcal{W}^*$ in the standard norm of an Euclidean space. We would assume that, only the first Markov parameter CB of the matrix $\mathbf{P}$ in (2) is known, and the later Markov parameters which are the off diagonal blocks of $\mathbf{P}$ are not known. Starting from an initial arbitrary guess, $\mathcal{W}(1)$ of $\mathcal{W}$, we consider the input update equation[1]

$$\mathcal{W}(j+1) \ = \ \mathcal{W}(j) \ + \ \mathcal{L}\ [\mathcal{Y}^* - \mathcal{Y}(j)] \qquad (3)$$

that recursively updates the choice of $\mathcal{W}$, based on the observed error vector $\mathcal{Y}^* - \mathcal{Y}(j)$. In (3), the 'gain matrix' $\mathcal{L} \in \mathcal{M}^{pN \times mN}$ and $j = 1, 2, \cdots$ are the iteration numbers.

Now we state (without proof) our main result from [12], which describes the design mechanism of the Luenberger observer based ILC.

Theorem 2.1: Consider the dynamical system (1) and the ILC input update equation (3), where $\mathcal{Y}^*$ is a desired sequence of output vectors in $\mathbb{R}^m$, in the image of $\mathbf{P}$. Let $\mathcal{L}$ be chosen such that

1) When $m \geq p$
   the poles of $\mathbf{I}_{pN} - \mathcal{L}\mathbf{P}$ are inside the unit disk.
2) When $m < p$
   the poles of $\mathbf{I}_{mN} - \mathbf{P}\mathcal{L}$ are inside the unit disk.

It would follow that for the input sequence $\mathcal{W}(j)$ of (3), we have

$$\lim_{j \to \infty} \mathcal{W}(j) \ = \ \mathcal{W}(\infty),$$

and for the corresponding output sequence $\mathcal{Y}(j)$, we have

$$\lim_{j \to \infty} \mathcal{Y}(j) \ = \ \mathcal{Y}^*,$$

where the convergences are exponential. When $m \geq p$, the input sequence $\mathcal{W}(\infty)$ is unique and does not depend on the initial condition $\mathcal{W}(1)$; and when $m < p$, the input sequence $\mathcal{W}(\infty)$ is not unique and depends on the initial condition $\mathcal{W}(1)$.

III. COOPERATIVE CONTROL

In this section, we introduce two iterative learning controllers in a cooperative setting. In our formulation, we assume that the 'motor controller' (MC) in the brain has a learning structure similar to an ILC. We also assume that the computer controller (CC) aiding the brain also has an ILC structure. Our main result is two-fold. In Section III-A, we prove the convergence of the cooperative learning controller. In Section III-B, we study the transfer of control from the computer controller (CC) to the motor controller (MC), as the patient recovers under therapy.

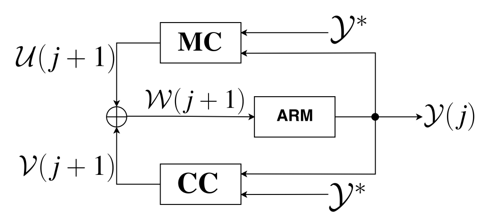

Fig. 1. Block diagram showing the cooperative 'arm dynamics' motor control learning behavior between motor controller (MC) in the brain and the computer controller (CC).

In Fig. 1, we show that $\mathcal{U}, \mathcal{V}$ are the input vectors to the arm dynamics both generated by MC and CC, viewed as iterative learning controllers. The output of the arm, that is being controlled, is $\mathcal{Y}$ and $\mathcal{Y}^*$ is the desired output (which is assumed to be in the column span of $\mathbf{P}$). The corresponding Markov parameter matrix (2) is $\mathbf{P}$, obtained from the arm dynamics[2]. The claim is that the sum of the MC and the CC device outputs would learn to control the arm dynamics to track a desired signal.

The input to the arm dynamical system is given by $\mathcal{W} = \mathcal{U} + \mathcal{V}$. The ILC update equations for the MC and CC are described as follows:

$$\mathcal{U}(j+1) = \mathcal{U}(j) + \mathcal{L}_1[\mathcal{Y}^* - \mathbf{P}\{\mathcal{U}(j) + \mathcal{V}(j)\}] \qquad (4)$$

$$\mathcal{V}(j+1) = \mathcal{V}(j) + \mathcal{L}_2[\mathcal{Y}^* - \mathbf{P}\{\mathcal{U}(j) + \mathcal{V}(j)\}] \qquad (5)$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the gain matrices of the MC and the CC devices respectively. Now we prove the convergence of the learning controllers in (4), (5).

[1] This iteration is standard in the ILC literature, see [13].

[2] See Section IV for its derivation.

Remark 3.1: In what follows, the theorems on cooperative learning control have been stated and proven for the case where $m \geq p$. A similar convergence result can be proven for the case where $m < p$. In Section IV, we illustrate this with an example.

A. Part I

Theorem 3.2: Let $m \geq p$ and let the MC and the CC input update laws be given as in (4), (5) respectively. If $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$ is chosen in accordance with Theorem 2.1, then $\mathcal{U}(j)$ and $\mathcal{V}(j)$ converge to $\mathcal{U}(\infty)$ and $\mathcal{V}(\infty)$ respectively such that $\mathcal{W}(\infty) = \mathcal{U}(\infty) + \mathcal{V}(\infty)$ where $\mathcal{Y}^* = \mathbf{P}\mathcal{W}(\infty)$.

Proof: It follows from adding the equations (4) and (5),

$$\mathcal{W}(j+1) = \mathcal{W}(j) + \mathcal{L}[\mathcal{Y}^* - \mathbf{P}\mathcal{W}(j)]. \tag{6}$$

If $\mathcal{L}$ is chosen in accordance with Theorem 2.1, then $\mathcal{W}(j)$ converges to $\mathcal{W}(\infty)$ exponentially so that $\mathcal{Y}^* = \mathbf{P}\mathcal{W}(\infty)$. We can re-write (4) as,

$$\begin{aligned} &\mathcal{U}(j+1) - \mathcal{U}(j) = \mathcal{L}_1\mathbf{P}[\mathcal{W}(\infty) - \mathcal{W}(j)] \\ &\implies \zeta_{\mathcal{U}}(j+1) = \mathcal{L}_1\mathbf{P}\mathcal{E}_{\mathcal{W}}(j), \end{aligned} \tag{7}$$

where we define, $\zeta_{\mathcal{U}}(j+1) = \mathcal{U}(j+1) - \mathcal{U}(j)$ and $\mathcal{E}_{\mathcal{W}}(j) = \mathcal{W}(\infty) - \mathcal{W}(j)$. By taking the limit as $j \to \infty$, it follows that $\mathcal{U}(j) \to \mathcal{U}(\infty)$[3]. Similarly from (5) it follows that $\mathcal{V}(j) \to \mathcal{V}(\infty)$. ∎

Let us write (4), (5) as a single dynamical system,

$$\begin{bmatrix} \mathcal{U} \\ \mathcal{V} \end{bmatrix}(j+1) = \Omega \begin{bmatrix} \mathcal{U} \\ \mathcal{V} \end{bmatrix}(j) + \begin{bmatrix} \mathcal{L}_1 \\ \mathcal{L}_2 \end{bmatrix}\mathcal{Y}^*, \tag{8}$$

$$\text{where,} \qquad \Omega = \left[I_{2pN} - \begin{bmatrix} \mathcal{L}_1 \\ \mathcal{L}_2 \end{bmatrix} \begin{bmatrix} \mathbf{P} & \mathbf{P} \end{bmatrix} \right]. \tag{9}$$

By subtracting two consecutive iterations of (8), we obtain,

$$\begin{bmatrix} \zeta_{\mathcal{U}} \\ \zeta_{\mathcal{V}} \end{bmatrix}(j+2) = \Omega \begin{bmatrix} \zeta_{\mathcal{U}} \\ \zeta_{\mathcal{V}} \end{bmatrix}(j+1). \tag{10}$$

We now state and prove the following lemma.

Lemma 3.3: Let $m \geq p$ and let the eigenvalues of $\mathbf{I}_{pN} - \mathcal{L}\mathbf{P}$ be chosen as $\lambda_1, \lambda_2, \cdots, \lambda_{pN}$ and let their corresponding eigenvectors be $\alpha_1, \alpha_2, \cdots, \alpha_{pN}$ respectively. Then the eigenvalues of the system matrix Ω of (10), are located at $\lambda_1, \lambda_2, \cdots, \lambda_{pN}$ and at 1 with algebraic multiplicity pN. The corresponding eigenvectors of the matrix Ω are respectively, $\begin{bmatrix} \mathcal{L}_1 \\ \mathcal{L}_2 \end{bmatrix} \circ \mathbf{P}(\alpha_1, \alpha_2, \cdots, \alpha_{pN})$ and $\begin{bmatrix} -I_{pN} \\ I_{pN} \end{bmatrix} \circ (\beta_1, \beta_2, \cdots, \beta_{pN})$, where $\beta_1, \beta_2, \cdots, \beta_{pN}$ are the eigenvectors of I_{pN}.

Remark 3.4: In the above theorem and hereafter, the notation $\circ$ is used to mean the following:

$$\begin{bmatrix} A \\ B \end{bmatrix} \circ C = \begin{bmatrix} AC \\ BC \end{bmatrix}, \tag{11}$$

where A and B are two matrices of order $n_1 \times n_2$ and where C is a matrix of order $n_2 \times n_3$.

[3]This is because the right hand side of (7) converges to zero exponentially.

Proof: For a given $i = 1, 2, \cdots, pN$,

$$\begin{aligned} \Omega \begin{bmatrix} \mathcal{L}_1 \\ \mathcal{L}_2 \end{bmatrix} \circ \mathbf{P}\alpha_i &= \begin{bmatrix} I - \mathcal{L}_1\mathbf{P} & -\mathcal{L}_1\mathbf{P} \\ -\mathcal{L}_2\mathbf{P} & I - \mathcal{L}_2\mathbf{P} \end{bmatrix} \begin{bmatrix} \mathcal{L}_1 \\ \mathcal{L}_2 \end{bmatrix} \circ \mathbf{P}\alpha_i \\ &= \begin{bmatrix} \mathcal{L}_1 - \mathcal{L}_1\mathbf{P}\mathcal{L}_1 - \mathcal{L}_1\mathbf{P}\mathcal{L}_2 \\ -\mathcal{L}_2\mathbf{P}\mathcal{L}_1 + \mathcal{L}_2 - \mathcal{L}_2\mathbf{P}\mathcal{L}_2 \end{bmatrix} \circ \mathbf{P}\alpha_i \\ &= \begin{bmatrix} \mathcal{L}_1\{I - \mathbf{P}(\mathcal{L}_1 + \mathcal{L}_2)\} \\ \mathcal{L}_2\{I - \mathbf{P}(\mathcal{L}_1 + \mathcal{L}_2)\} \end{bmatrix} \circ \mathbf{P}\alpha_i \\ &= \lambda_i \begin{bmatrix} \mathcal{L}_1 \\ \mathcal{L}_2 \end{bmatrix} \circ \mathbf{P}\alpha_i \end{aligned}$$

Therefore λ_i is an eigenvalue with the corresponding eigenvector $\begin{bmatrix} \mathcal{L}_1 \\ \mathcal{L}_2 \end{bmatrix} \circ \mathbf{P}\alpha_i$. Next for a given $i = 1, 2, \cdots, pN$;

$$\begin{aligned} \Omega \begin{bmatrix} -I_{pN} \\ I_{pN} \end{bmatrix} \circ \beta_i &= \begin{bmatrix} I - \mathcal{L}_1\mathbf{P} & -\mathcal{L}_1\mathbf{P} \\ -\mathcal{L}_2\mathbf{P} & I - \mathcal{L}_2\mathbf{P} \end{bmatrix} \begin{bmatrix} -I_{pN} \\ I_{pN} \end{bmatrix} \circ \beta_i \\ &= \begin{bmatrix} -\beta_i + \mathcal{L}_1\mathbf{P}\beta_i - \mathcal{L}_1\mathbf{P}\beta_i \\ \mathcal{L}_2\mathbf{P}\beta_i + \beta_i - \mathcal{L}_2\mathbf{P}\beta_i \end{bmatrix} \\ &= 1 \begin{bmatrix} -I_{pN} \\ I_{pN} \end{bmatrix} \circ \beta_i \end{aligned}$$

It follows that, 1 is an eigenvalue of Ω with the corresponding eigenvector $\begin{bmatrix} -I_{pN} \\ I_{pN} \end{bmatrix} \circ \beta_i$ and this completes the proof. ∎

Remark 3.5: We note that, when $m \geq p$, it is possible to choose an $\mathcal{L}$ that will arbitrarily place the eigenvalues of $\mathbf{I}_{pN} - \mathcal{L}\mathbf{P}$ as, $(\mathbf{P}, \mathbf{I}_{pN})$ is an observable pair (see [14]). However when $m < p$, this is not possible. In this case, we may still choose an $\mathcal{L}$ that will arbitrarily place the eigenvalues of $\mathbf{I}_{mN} - \mathbf{P}\mathcal{L}$ as, $(\mathbf{I}_{pN}, \mathbf{P})$ is a controllable pair. Then, the eigenvalues of $\mathbf{I}_{pN} - \mathcal{L}\mathbf{P}$ will include the same eigenvalues of $\mathbf{I}_{mN} - \mathbf{P}\mathcal{L}$ that we chose and the repeated eigenvalue 1 with algebraic multiplicity $(p-m)N$. Then it can be shown that the ILC process will not excite this unstable subspace (See [12] for details and IV-A for an example).

B. Part II

In this subsection, we look at the transfer of control from CC to MC. This is achieved by lowering the strength of the CC input $\mathcal{V}(j)$ by a factor η at the start of each session.

Theorem 3.6: Let $m \geq p$. Let $\mathcal{L}_1 = \tau\mathcal{L}$ and $\mathcal{L}_2 = (1-\tau)\mathcal{L}$ with $\tau \in (0,1)$ in (4) and (5) respectively. Let $\mathcal{L}$ be chosen in accordance with Theorem 2.1, and let the eigenvectors of $\mathbf{I}_{pN} - \mathcal{L}\mathbf{P}$ form a basis of $\mathbb{R}^{pN}$. Let us also assume that at the beginning of each session, the input vectors $\mathcal{U}^{(r)}$ and $\mathcal{V}^{(r)}$ are initialized as $\mathcal{U}^{(r)}(1) = \mathcal{U}^{(r-1)}(\infty)$ and $\mathcal{V}^{(r)}(1) = \eta\mathcal{V}^{(r-1)}(\infty)$ respectively where $\eta \in (0,1)$. Then for all sessions $r = 1, 2, \cdots$, and for all iterations $j = 1, 2, \cdots$, we have the following

1) $(1-\tau)\zeta_{\mathcal{U}}^{(r)}(j+2) = \tau\zeta_{\mathcal{V}}^{(r)}(j+2)$,

2) $\mathcal{V}^{(r)}(\infty) = \{1 - \tau(1-\eta)\}\mathcal{V}^{(r-1)}(\infty)$,

3) $\mathcal{U}^{(r)}(\infty) = \mathcal{U}^{(r-1)}(\infty) + \tau(1-\eta)\mathcal{V}^{(r-1)}(\infty)$.

Thus, $\lim_{r\to\infty} \mathcal{U}^{(r)}(\infty) = \mathcal{W}(\infty)$ and $\lim_{r\to\infty} \mathcal{V}^{(r)}(\infty) = 0$. (The superscript $^{(r)}$ indicates the respective control variables on the r^{th} session with $r \in \mathbb{Z}^+$.)

Proof: First let us observe that, at the beginning of the r^{th} session, with $j = 1$ and with the initializations as in the hypothesis, it follows from (4) (or (8)) that,

$$\begin{aligned}\zeta_{\mathcal{U}}^{(r)}(2) &= \mathcal{L}_1\mathbf{P}[\mathcal{W}(\infty) - \mathcal{U}^{(r-1)}(\infty) - \eta\mathcal{V}^{(r-1)}(\infty)] \\ &= \mathcal{L}_1\mathbf{P}[\mathcal{W}(\infty) - \mathcal{U}^{(r-1)}(\infty) - \mathcal{V}^{(r-1)}(\infty) \\ &\quad + \mathcal{V}^{(r-1)}(\infty) - \eta\mathcal{V}^{(r-1)}(\infty)]. \end{aligned} \tag{12}$$

Since $\mathcal{W}^* = \mathcal{U}^{(r-1)}(\infty) + \mathcal{V}^{(r-1)}(\infty)$, it follows from (12) that,

$$\zeta_{\mathcal{U}}^{(r)}(2) = \mathcal{L}_1\mathbf{P}(1-\eta)\mathcal{V}^{(r-1)}(\infty). \tag{13}$$

Similarly considering (5) (or (8)), we obtain

$$\zeta_{\mathcal{V}}^{(r)}(2) = \mathcal{L}_2\mathbf{P}(1-\eta)\mathcal{V}^{(r-1)}(\infty). \tag{14}$$

Therefore, $\begin{bmatrix}\zeta_{\mathcal{U}}^{(r)}\\ \zeta_{\mathcal{V}}^{(r)}\end{bmatrix}(2) = (1-\eta)\begin{bmatrix}\mathcal{L}_1\\ \mathcal{L}_2\end{bmatrix} \circ \mathbf{P}\mathcal{V}^{(r-1)}(\infty)$.
Since $\mathcal{V}^{(r-1)}(\infty) \in \mathbb{R}^{pN}$ let us denote,

$$\mathcal{V}^{(r-1)}(\infty) = \sum_{i=1}^{pN} c_i^{\mathcal{V}(\infty),(r-1)}\alpha_i \tag{15}$$

where, $c_i^{\mathcal{V}(\infty),(r-1)} \in \mathbb{R}$ and $\alpha_i \in \mathbb{R}^{pN}$ are the eigenvectors of $\mathbf{I}_{pN} - \mathcal{L}\mathbf{P}$. It can be shown that these eigenvectors construct a basis of $\mathbb{R}^{pN}$ (see [15]). It now follows from (10) that,

$$\begin{aligned}\begin{bmatrix}\zeta_{\mathcal{U}}^{(r)}\\ \zeta_{\mathcal{V}}^{(r)}\end{bmatrix}(j+2) &= \Omega^j\begin{bmatrix}\zeta_{\mathcal{U}}^{(r)}\\ \zeta_{\mathcal{V}}^{(r)}\end{bmatrix}(2) \\ &= \Omega^j(1-\eta)\begin{bmatrix}\mathcal{L}_1\\ \mathcal{L}_2\end{bmatrix} \circ \mathbf{P}\sum_{i=1}^{pN} c_i^{\mathcal{V}(\infty),(r-1)}\alpha_i \\ &= (1-\eta)\sum_{i=1}^{pN} c_i^{\mathcal{V}(\infty),(r-1)}\Omega^j\begin{bmatrix}\mathcal{L}_1\\ \mathcal{L}_2\end{bmatrix} \circ \mathbf{P}\alpha_i. \end{aligned} \tag{16}$$

Applying the results from Lemma 3.3 we have,

$$\begin{bmatrix}\zeta_{\mathcal{U}}^{(r)}\\ \zeta_{\mathcal{V}}^{(r)}\end{bmatrix}(j+2) = (1-\eta)\sum_{i=1}^{pN} c_i^{\mathcal{V}(\infty),(r-1)}\lambda_i^j\begin{bmatrix}\mathcal{L}_1\\ \mathcal{L}_2\end{bmatrix} \circ \mathbf{P}\alpha_i. \tag{17}$$

Now, for a given r, it follows from Theorem 3.2 that, $\mathcal{U}^{(r)}(j) \to \mathcal{U}^{(r)}(\infty)$ and $\mathcal{V}^{(r)}(j) \to \mathcal{V}^{(r)}(\infty)$. Therefore, by summing up over all j and taking the limit as $j \to \infty$, it follows from (17) that,

$$\begin{aligned}\begin{bmatrix}\mathcal{U}^{(r)}\\ \mathcal{V}^{(r)}\end{bmatrix}(\infty) - \begin{bmatrix}\mathcal{U}^{(r)}\\ \mathcal{V}^{(r)}\end{bmatrix}(1) &= (1-\eta)\sum_{i=1}^{pN} c_i^{\mathcal{V}(\infty),(r-1)}\lim_{j\to\infty}\sum_{l=1}^{j}\lambda_i^l\begin{bmatrix}\mathcal{L}_1\\ \mathcal{L}_2\end{bmatrix} \circ \mathbf{P}\alpha_i, \\ \begin{bmatrix}\mathcal{U}^{(r)}\\ \mathcal{V}^{(r)}\end{bmatrix}(\infty) - \begin{bmatrix}\mathcal{U}^{(r-1)}\\ \eta\mathcal{V}^{(r-1)}\end{bmatrix}(\infty) &= (1-\eta)\sum_{i=1}^{pN} c_i^{\mathcal{V}(\infty),(r-1)}\lim_{j\to\infty}\frac{1-\lambda_i^j}{1-\lambda_i}\begin{bmatrix}\mathcal{L}_1\\ \mathcal{L}_2\end{bmatrix} \circ \mathbf{P}\alpha_i \\ &= (1-\eta)\sum_{i=1}^{pN} c_i^{\mathcal{V}(\infty),(r-1)}\frac{1}{1-\lambda_i}\begin{bmatrix}\mathcal{L}_1\\ \mathcal{L}_2\end{bmatrix} \circ \mathbf{P}\alpha_i. \end{aligned} \tag{18}$$

Next, considering the decomposition of $\mathcal{V}^{(r-1)}(\infty)$ as in (15), it follows form (18) that,

$$\mathcal{V}^{(r)}(\infty) = \sum_{i=1}^{pN}\Big\{I\eta + \frac{1-\eta}{1-\lambda_i}\mathcal{L}_2\mathbf{P}\Big\}c_i^{\mathcal{V}(\infty),(r-1)}\alpha_i. \tag{19}$$

Now if we apply the conditions that, $\mathcal{L}_1 = \tau\mathcal{L}$ and $\mathcal{L}_2 = (1-\tau)\mathcal{L}$ in (13) and (14), we have,

$$(1-\tau)\zeta_{\mathcal{U}}^{(r)}(j+1) = \tau\zeta_{\mathcal{V}}^{(r)}(j+1),$$

and from (19) we have,

$$\mathcal{V}^{(r)}(\infty) = \sum_{i=1}^{pN}\Big\{I\eta + (1-\tau)\frac{1-\eta}{1-\lambda_i}\mathcal{L}\mathbf{P}\Big\}c_i^{\mathcal{V}(\infty),(r-1)}\alpha_i.$$

It is a simple exercise to see that, given the eigenvalue, eigenvector pairs (λ_i, α_i) of $\mathbf{I}_{pN} - \mathcal{L}\mathbf{P}$ where $i = 1, 2, \cdots, pN$; the eigenvalue eigenvector pairs of $\mathcal{L}\mathbf{P}$ are $(1-\lambda_i, \alpha_i)$. Therefore we have,

$$\begin{aligned}\mathcal{V}^{(r)}(\infty) &= \sum_{i=1}^{pN}\Big\{\eta + (1-\tau)\frac{1-\eta}{1-\lambda_i}(1-\lambda_i)\Big\}c_i^{\mathcal{V}(\infty),(r-1)}\alpha_i \\ &= \{\eta + (1-\tau)(1-\eta)\}\sum_{i=1}^{pN} c_i^{\mathcal{V}(\infty),(r-1)}\alpha_i \end{aligned} \tag{20}$$

Using the decomposition (15), (20) simplifies into,

$$\mathcal{V}^{(r)}(\infty) = \{1 - \tau(1-\eta)\}\mathcal{V}^{(r-1)}(\infty). \tag{21}$$

Similarly from (18) we have,

$$\mathcal{U}^{(r)}(\infty) = \mathcal{U}^{(r-1)}(\infty) + \tau(1-\eta)\mathcal{V}^{(r-1)}(\infty).$$

Let us denote $a = 1 - \tau(1-\eta)$. Then from the definition of τ and η it follows that $0 < a < 1$. Therefore it follows from (21) that,

$$\|\mathcal{V}^{(r)}(\infty)\| \leq a^{r-1}\|\mathcal{V}^{(1)}(\infty)\|. \tag{22}$$

This implies that $\lim_{r\to\infty} \mathcal{V}^{(r)}(\infty) = 0$ exponentially. But since $\mathcal{U}^{(r)}(\infty) + \mathcal{V}^{(r)}(\infty) = \mathcal{W}(\infty) = \mathcal{U}^{(r-1)}(\infty) + \mathcal{V}^{(r-1)}(\infty)$ it must follow that, $\lim_{r\to\infty} \mathcal{U}^{(r)}(\infty) = \mathcal{W}(\infty)$; and this completes the proof. ■

Remark 3.7: We would like to note that the hypothesis $\mathcal{L}_1 = \tau\mathcal{L}$ and $\mathcal{L}_2 = (1-\tau)\mathcal{L}$ was made as an easily implementable design method. If a suitable η can be found such that $\|I\eta + \frac{1-\eta}{1-\lambda_{max}}\mathcal{L}_2\mathbf{P}\| < 1$, where $\lambda_{max} = \max_{1\leq i\leq pN}\lambda_i$, the two input vectors will still converge as $\lim_{r\to\infty} \mathcal{V}^{(r)}(\infty) = 0$ and $\lim_{r\to\infty} \mathcal{U}^{(r)}(\infty) = \mathcal{W}^*$; with the only requirement for $\mathcal{L}_1$ and $\mathcal{L}_2$ being that, $\mathcal{L}_1 + \mathcal{L}_2 = \mathcal{L}$.

Remark 3.8: The cooperative control algorithm presented in this section, can also be implemented block-wise as discussed in [12]. As long as the eigenvalues of $\mathbf{I}_{pN} - \mathcal{L}\mathbf{P}$ (or $\mathbf{I}_{mN} - \mathbf{P}\mathcal{L}$) are chosen inside the unit disk, the convergence of the inputs $u_i^{(r)}(j+1)$ and $v_i^{(r)}(j+1)$ are guaranteed and the convergence of the vectors $\mathcal{U}^{(r)}(j+1)$ and $\mathcal{V}^{(r)}(j+1)$ will follow suit. We omit the details.

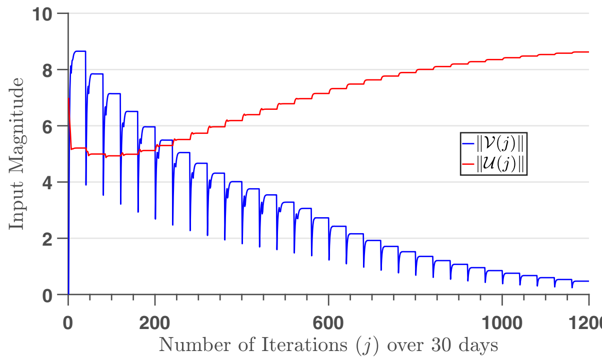

Fig. 2. Cooperative learning control of the 3-input 2-output system: Showing the transfer of control from one controller to the other.

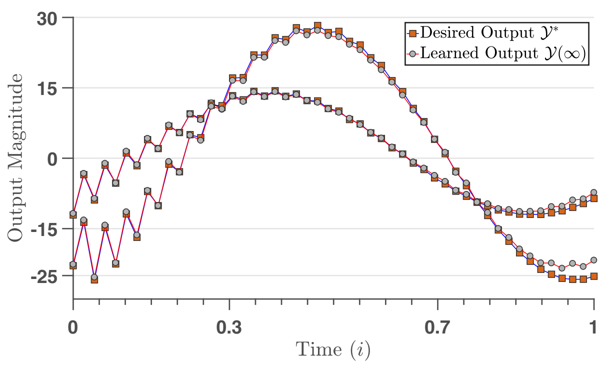

Fig. 3. Cooperative learning control of the 3-input 2-output system: Comparison of the desired vs. the learned trajectories at the end of 30 sessions.

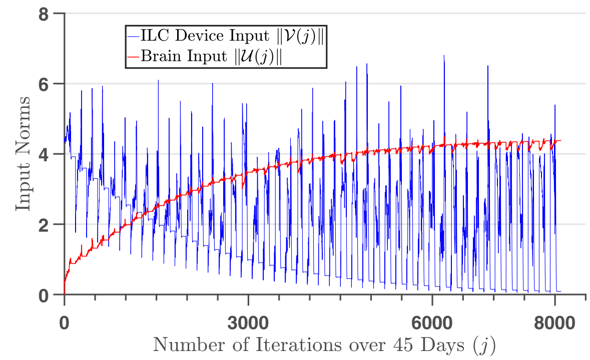

Fig. 4. Cooperative learning control of the 2-link arm with observer based iterative learning controllers. Control is transfered from Motor Controller to the Computer Controller in 45 sessions.

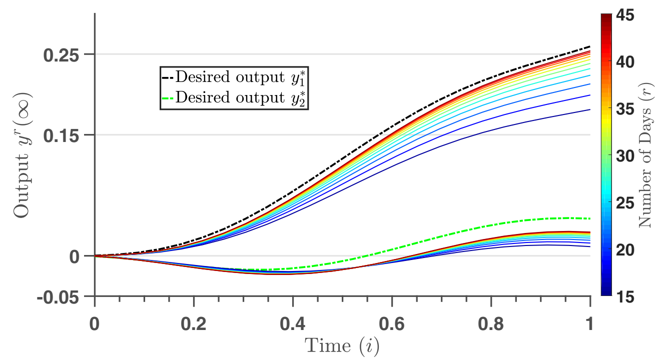

Fig. 5. Cooperative learning control of the 2-link arm with observer based iterative learning controllers: End-effector trajectory of (25) has been shown, with no assistance from the Computer Controller. The trajectory is approaching the desired trajectory with increasing number of sessions.

IV. SIMULATION

We demonstrate our simulation results in two parts. We use the cooperative learning control algorithm, in Section IV-A, to control a 3-input 2-output discrete time system and in Section IV-B, to control a linearized 2-link arm.

A. Part I

Here we consider a minimal 3-input, 2-output system with the with the following system parameters.

$$B = \begin{bmatrix} 1 & 3 & -1 \\ 2 & -1 & 1 \\ -2 & 0 & 2 \\ 0 & 1 & -2 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 4 & 2 & -3 \\ -4 & 1 & 2 & -1 \end{bmatrix}.$$

$$A = T \,\mathrm{diag}\,[-0.9 \quad -0.2 \quad 0.1 \quad 0.5]\; T^{-1}, \quad \text{where, } T = \begin{bmatrix} 1 & 2 & 0 & -2 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & 3 & -1 \\ 1 & -1 & -1 & 0 \end{bmatrix}.$$

The desired input is a priori chosen (for the purpose of simulation) as

$$u^*(t) = \left[\sin 2\pi t + 0.2\cos\tfrac{\pi t}{8}, \cos 2\pi t, 0.3\sin 2\pi t - 0.7\cos 2\pi t\right]^T.$$

With, $N = 50, \eta = 0.45, \tau = 0.8$ the Luenberger observer based ILC algorithm, with error saturation, is run 40 iterations per session for 30 sessions. The repeated poles of $\mathbf{I}_{mN} - \mathbf{P}\mathcal{L}$ are placed at $[0.5, 0.52]$. Fig. 2 shows the transfer of control from one controller to the other. At the end of 30 sessions, the input $\mathcal{U}$ is fed into the system and the outputs are compared in Fig. 3.

B. Part II

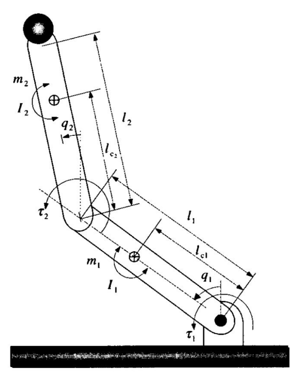

Fig. 6. The 2-link arm.

Next consider the 2-link arm depicted in Fig. 6. Let $\mathbf{q} = [q_1, q_2]^T$ be the 2 joint angles, $\tau = [\tau_1, \tau_2]^T$ be the joint input torques, $\mathbf{y}$ be the output of the system and I_1, I_2 be the moment of inertia of the two links. Then the perturbed two link arm can be represented as,

$$\begin{aligned}(\mathbf{D} + \mathbf{\Delta D})\,\ddot{\mathbf{q}} + (\mathbf{C} + \mathbf{\Delta C})\dot{\mathbf{q}} + \mathbf{g} + \mathbf{\Delta g} &= \tau \\ \mathbf{y} &= \mathbf{q},\end{aligned} \tag{23}$$

where, $\mathbf{D}(\mathbf{q})$ is the 2×2 arm inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ is a 2-vector of Coriolis torques and centrifugal forces and $\mathbf{g}(\mathbf{q})$ is a 2-vector of gravitational torques [16]. The Δ represents unknown perturbations to the system. Let the feedback controller τ be chosen as,

$$\tau = \mathbf{D}(\mathbf{q})\mathbf{w} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}), \tag{24}$$

where $\mathbf{w}$ is the input of the closed loop system [17]. Assuming that $(\mathbf{D} + \Delta\mathbf{D})$ is invertible, the closed loop system becomes,

$$\ddot{\mathbf{q}} = (\mathbf{D} + \boldsymbol{\Delta}\mathbf{D})^{-1}\mathbf{D}\mathbf{w} - (\mathbf{D} + \boldsymbol{\Delta}\mathbf{D})^{-1}\boldsymbol{\Delta}\mathbf{C}\dot{\mathbf{q}} - (\mathbf{D} + \boldsymbol{\Delta}\mathbf{D})^{-1}\boldsymbol{\Delta}\mathbf{g}, \quad (25)$$

where, $\mathbf{D} = \begin{bmatrix} H_{11} & H_{12}cos(q_1 - q_2) \\ H_{21}cos(q_1 - q_2) & H_{22} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 & h\,sin(q_1 - q_2)\dot{q}_2 \\ -h\,sin(q_1 - q_2)\dot{q}_1 & 0 \end{bmatrix}$, and $\mathbf{g} = \begin{bmatrix} g_{11}\,sin(q_1) & g_{22}\,sin(q_2) \end{bmatrix}^T$. The various parameters are defined as; $H_{11} = m_1 l_{c1}^2 + m_2 l_1^2 + I_1$, $H_{22} = m_2 l_{c2}^2 + I_2$, $H_{12} = m_2 l_1 l_{c2}$, $H_{21} = m_2 l_1 l_{c2}$, $g_{11} = -[m_1 l_{c1} + m_2 l_1]g$, $g_{22} = -m_2 l_{c2} g$, $h = m_2 l_1 l_{c2}$. For simulation purposes the physical parameters of the 2-link are assigned the following numerical values; $l_1 = 0.5, l_2 = 0.6, l_{c1} = 0.3, l_{c2} = 0.35, m_1 = 2.5, m_2 = 2, I_1 = 4.2, I_2 = 2.5, g = 9.8$. As for the uncertainty Δ, a 15% increment is added to each of these physical parameters. Using the state transformation $\begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^T = \begin{bmatrix} q_1 & q_2 & \dot{q}_1 & \dot{q}_2 \end{bmatrix}^T$, and a Taylor series approximation about the states $\mathbf{q}$ that correspond to the middle block $(N/2)$ of the desired output trajectory, (25) can be written in the following form,

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{w}; \quad \mathbf{y} = \psi\mathbf{z} \quad (26)$$

where, $\mathbf{A} \in \mathcal{M}^{4\times4}, \mathbf{B} \in \mathcal{M}^{4\times2}$ and $\psi \in \mathcal{M}^{2\times4}$. This system (26), can then be discretized with zero-order-hold and we shall denote it as,

$$\mathbf{z_{k+1}} = \mathbf{G}\mathbf{z_k} + \mathbf{H}\mathbf{w_k}; \quad \mathbf{y_k} = \psi\mathbf{z_k} \quad (27)$$

To illustrate the use of cooperative learning control of the 2-link arm described above, we apply the Luenberger observer based ILC algorithm, with error saturation [12], to the two learners, MC and CC depicted in Fig. 1. The desired output is generated by feeding the input,

$$\mathbf{w(t)} = \left[\sin(2\pi t) + 0.2\cos(\frac{\pi t}{8}),\ 0.3\sin(2\pi t) - 0.7\cos(2\pi t)\right]$$

(for the purpose of simulation) to the *perturbed* closed loop system (25). Each updated input $\mathcal{U}(j)$ and $\mathcal{U}(j)$ (according to (4) and (5)) is then fed into the linear approximation (26) to produce the output $\mathcal{Y}(j)$, which is used in (3) iteratively. In this simulation we choose the parameter values $N = 24, \tau = 0.85$ and $\eta = 0.45$. The poles of each block $\mathbf{I}_p - \mathcal{L}_b CB$ are placed at $[0.5, 0.52]$(see [12] for details). In each session the ILC algorithm is run 180 iterations and this is continued for 45 sessions.

At the end of each session, the overall output error $\|\mathcal{Y}^* - \mathcal{Y}(j)\|$ of the arm, will converge to zero. As the output $\mathcal{Y}(j)$ converge to the desired value $\mathcal{Y}^*$, the two inputs $\mathcal{U}(j)$ and $\mathcal{V}(j)$ will converge to fixed values. Then, as these inputs are initialized according to Theorem 3.6, the control will transfer from the ILC device CC (i.e. $\mathcal{V}$) to the MC (i.e. $\mathcal{U}$). This is shown in Fig. 4. If the control retained by the MC at the end of each session $\mathcal{U}^r(\infty)$ alone were injected to the arm (i.e. without the involvement of the ILC device), the resulting output will still be erroneous. However this output will approach the desired trajectory over sessions. This is shown in Fig. 5.

Remark 4.1: In the above simulations, all parameters of the system (1), including the desired inputs, are presented. It must be understood that this is done merely for the reproduction of results. In order to learn the desired input, ILC algorithm (3), requires only the knowledge of the first Markov parameter CB and the desired output $\mathcal{Y}^*$.

V. CONCLUSION

Cooperative learning between two iterative learning controllers, that characterizes the stroke rehabilitation of the upper limb [10] scenario, is studied. In particular, the Luenberger observer based ILC algorithm [12] is considered. Once the two controllers, in cooperation with each other, learn the desired input; one controller transfers it's control to the other iteratively. The two controllers have to be properly initialized at the beginning of each session. They are allowed to communicate with each other only at the beginning of the first session and further communication between the controllers is not necessary.

REFERENCES

[1] M. Uchiyama, "Formation of high speed motion pattern of mechanical arm by trial," *Trans. of the Society of Instrumentation and Control Engineers*, vol. 19, pp. 706–712, May 1978.

[2] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1(2), pp. 123–140, 1984.

[3] H.-S. Ahn, Y. Chen, and K. Moore, "Iterative learning control: Brief survey and categorization," vol. 37, pp. 1099 – 1121, 12 2007.

[4] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, June 2006.

[5] X. Zhao, Y. Chu, J. Han, and Z. Zhang, "Ssvep-based brain-computer interface controlled functional electrical stimulation system for upper extremity rehabilitation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 947–956, July 2016.

[6] X. L. Shiping Yang, Jian-Xin Xu and D. Shen, *Iterative Learning Control for Multi-agent Systems Coordination*. Wiley-IEEE Press, 2017.

[7] D. Meng and K. L. Moore, "Learning to cooperate: Networks of formation agents with switching topologies," *Automatica*, vol. 64, pp. 278 – 293, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000510981500477X

[8] N. E. Leonard and A. Olshevsky, "Cooperative learning in multiagent systems from intermittent measurements," *SIAM Journal on Control and Optimization*, vol. 53, no. 1, pp. 1–29, 2015. [Online]. Available: https://doi.org/10.1137/120891010

[9] K. H.-M. Frank L. Lewis, Hongwei Zhang and A. Das, *Cooperative Control of Multi-Agent Systems*. New Jersey, USA: Springer, 2014.

[10] C. T. Freeman, E. Rogers, A. M. Hughes, J. H. Burridge, and K. L. Meadmore, "Iterative learning control in health care: Electrical stimulation and robotic-assisted upper-limb stroke rehabilitation," *IEEE Control Systems*, vol. 32, no. 1, pp. 18–43, Feb 2012.

[11] C. Freeman, E. Rogers, J. H. Burridge, A.-M. Hughes, and K. L. Meadmore, *Iterative Learning Control for Electrical Stimulation and Stroke Rehabilitation*. London, UK: Springer-Verlag London, 2015.

[12] R. N. Jayawardhana and B. K. Ghosh, "Observer based iterative learning controller design for MIMO systems in discrete time," in *Proceedings of the American Control Conference (ACC)*, Milwaukee, USA, June 27-29, 2018.

[13] K. L. Moore, *Iterative Learning Control for Deterministic Systems*. Springer Verlag, 1993.

[14] K. Ogata, *Discrete-Time Control Systems*. Prentice-Hall, Inc., 1994.

[15] R. N. Jayawardhana, "Iterative learning control for discrete-time MIMO systems and its application in non-linear systems and cooperative learning control," Ph.D. dissertation, Texas Tech University, 2017.

[16] H. Asada and J.-J. E. Slotine, *Robot Analysis and Control*. Massachusetts Institute of Technology: John Wiley & Sons, Inc., 1986.

[17] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. John Wiley & Sons, Inc., 1989.