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

Gradient-Based Iterative Learning Control for Decentralised Collaborative Tracking

Shangcheng Chen, Christopher Freeman and Bing Chu

Abstract— **Collaborative tracking control of multi-agent systems involves two or more subsystems working together to perform a global objective, and is increasingly used within a diverse range of applications. However existing, predominately centralised, control structures are sensitive to communication delays and data drop-out leading to inaccurate tracking. Iterative learning control (ILC) has been applied to increase performance using past experience, but reliance on inverse dynamics has severely reduced robustness to model uncertainty.**

This paper proposes the first general decentralized iterative learning framework to address this problem, thereby enabling a wide range of existing ILC methodologies to be applied to this area. This framework is illustrated through the derivation of a decentralised gradient based ILC algorithm which ensures convergence to the required reference trajectory, while simultaneously optimising the control input energy. In addition, a novel balancing algorithm is also proposed to distribute the input energy of each agent and hence avoid sub agent overloading.

I. 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. Collaborative tracking for MAS is an important problem in many engineering applications [1] such as unmanned aerial vehicles, underwater systems, smart home technology [2] and multiple robot industrial manufacturing. Here the global goal is for the combined system to achieve a specified reference trajectory and typically the task is repeatedly performed, e.g. a group of manipulators performing operations on a payload at different stages on a production line must collaborate to accurately achieve the required force profile.

Existing control approaches for collaborative control comprise distributed learning [3], adaptive neural networks [4], feedback control using minimal positional estimates [5], and distributed adaptive control [6]. Research has primarily focused on communication protocols and asymptotic behaviour, as opposed to achieving high performance dynamic tracking. This is because most existing approaches are assumed to operate using a centralised control structure, which is often impractical for two main reasons: firstly, it requires full real-time input and output information from all subsystems which is generally not available due to the limited communication capacity and specific network topology; secondly, it has poor scaling properties, i.e. when the number of subsystems is large, traditional centralised control approaches require a complete redesign which is computationally expensive. This was addressed by the novel decentralized control structure of [6] which improved the robustness of the system to network delays and data drop-out by reducing the communication times between each agents.

In addition to the drawback of centralised control structures, tracking performance is also degraded due to the reliance of existing approaches on accurate models of agent dynamics. This was addressed in [7] by proposing iterative learning control (ILC) to solve the collaborative tracking problem. ILC uses experimental input and output error from previous attempts at the collaborative task to update the current system input, with the objective of successively improving the output tracking performance [8], [9], [10]. ILC can theoretically ensure convergence of the output to zero tracking error through this exploitation of past experience. In [7] an inverse ILC algorithm was proposed, which achieves rapid convergence, but is known to be extremely sensitive to modeling uncertainty [11], i.e. its robustness margin tends to zero at high frequencies. Furthermore, it cannot be used with non-minimum phase dynamics which excludes large application classes, such as many aerial and water vehicles.

This paper addresses the above limitations by introducing a general decentralized ILC framework for collaborative tracking. This enables a broad class of ILC algorithms to be used in a decentralized manner, thereby addressing the drawbacks of inverse based ILC while still ensuring high performance tracking. This framework is exemplified through development of a decentralized gradient-based ILC update, which is shown to provide attractive convergence properties, i.e. zero tracking error and minimal input energy. A novel balancing algorithm is then proposed to allow the input energy allocation of each agent to be prescribed. This is the first solution to the problem of energy distribution in collaborative MAS control.

This paper is organized as follows. In Section 2, the MAS collaborative problem is defined and the first general framework for decentralised collaborative ILC is introduced. In Section 3, a specific solution is proposed and MAS convergence properties are derived. Section 4 develops the input energy balance algorithm. In Section 5, simulation results are presented to demonstrate efficacy of the proposed framework, and Section 6 draws conclusions.

II. PROBLEM FORMULATION

A. Collaborative tracking control of repeated tasks

Consider a group of n single-input, single-output (SISO) agents collaborating to perform a repeated tracking task of

S.C. Chen, C.T. Freeman and B. Chu are with Electronics and Computer Science, University of Southampton, Southampton, UK, SO17 1BJ {Shangcheng.Chen, ctf1, B.Chu } @soton.ac.uk

duration N samples. On the k^{th} iteration of the task, agent i is modelled by the discrete linear time-invariant (LTI) system

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

where $t = 0, 1, \cdots, N$ is the time step, $x_{i,k}(t) \in \mathbb{R}^{n_i}$ is the state variable, $y_{i,k}(t) \in \mathbb{R}$ the output variable, $u_{i,k}(t) \in \mathbb{R}$ the control input variable, and A_i, B_i and C_i are matrices of appropriate dimensions. For simplicity it is assumed that $C_i B_i \neq 0$, $i = 1, \cdots, n$. At the end of every trial, agent i is reset to initial condition $x_{i,0}$ which is fixed for all trials of the task.

The total system output over the k^{th} trial is the combined sum of all agent contributions, i.e. it is given by

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

and the collaborative tracking problem is for $y_k(t)$ to follow a reference $y_d(t) \in \mathbb{R}$ with zero tracking error $e_k(t)$, i.e.

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

A common ILC design and analysis tool is to represent each trial as a single sample of a high dimensional 'lifted' system. Accordingly the signals in (1),(2),(3) become the supervectors (for $i = 1, \ldots, n$)

$$\begin{aligned} u_{i,k} &= [u_{i,k}(0), u_{i,k}(1), \cdots, u_{i,k}(N-1)]^T \in \mathbb{R}^N, \\ y_{i,k} &= [y_{i,k}(1), y_{i,k}(2), \cdots, y_{i,k}(N)]^T \in \mathbb{R}^N, \\ y_k &= [y_k(1), y_k(2), \cdots, y_k(N)]^T \in \mathbb{R}^N, \\ y_d &= [y_d(1), y_d(2), \cdots, y_d(N)]^T \in \mathbb{R}^N, \end{aligned}$$

then (1) can be written in the corresponding form

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

where $G_i \in \mathbb{R}^{N \times N}$ is the toeplitz matrix

$$G_i = \begin{bmatrix} C_i B_i & 0 & \cdots & 0 & 0 \\ C_i A_i B_i & C_i B_i & \ddots & 0 & 0 \\ C_i A_i^2 B_i & C_i A_i B_i & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & C_i B_i & 0 \\ C_i A_i^{N-1} B_i & \ldots & \ldots & C_i A_i B_i & C_i B_i \end{bmatrix}$$

which is the lifted implementation of the along-the-trial transfer-function $G_i(z) = C_i(zI - A_i)^{-1} B_i$, and d_i is the initial condition response

$$d_i = [C_i A_i x_{i,0}, C_i A_i^2 x_{i,0}, \cdots, C_i A_i^N x_{i,0}]^T \in \mathbb{R}^N.$$

The initial state can be consumed into y_d and hence we may always assume $d_i = 0$, or equivalently $x_{i,0} = 0$.

B. Optimal decentralized collaborative ILC

The ILC problem is to update the control input to each agent based on experimental data from previous trials of the collaborative tracking task. In this paper a decentralized ILC framework is proposed in which each agent must update its own input using the global error, but with no knowledge of the other agents. This overcomes the limitations associated with communications overhead, complexity and lack of flexibility inherent in centralised control approaches. The resulting constrained ILC structure has not previously been addressed in the ILC research community. The requirements of this structure are defined as follows.

Definition 1: The ILC update for the i^{th} agent has form

$$u_{i,k+1} = f_i(u_{i,k}, e_k), \quad i = 1, 2, \cdots, n \tag{5}$$

where f_i depends only on $\{A_i, B_i, C_i\}$. When iteratively applied to agent dynamics (1), system output (2) satisfies

$$\lim_{k \to \infty} y_k = y^* \tag{6}$$

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

$$\lim_{k \to \infty} u_k = u^* \tag{7}$$

with the optimal limiting signals given by

$$u^* = \arg\min\{J(u_k) \mid J(u_k) = \sum_{i=1}^{n} \|u_{i,k}\|_{R_i}^2, y_k = y^*\} \tag{8}$$

$$y^* = \arg\min\{J(y_k) \mid J(y_k) = \|y_d - y_k\|_Q^2\} \tag{9}$$

in which the weighted norms are defined by

$$\begin{aligned} ||u_i||_{R_i} &= \sqrt{\langle u_i, u_i \rangle_{R_i}} = \sqrt{u_i^T R_i u_i} \\ ||y||_Q &= \sqrt{\langle y, y \rangle_Q} = \sqrt{y^T Q y} \end{aligned}$$

with R_i,Q symmetric positive definite weighting matrices.

Remark 1: It is later illustrated how selection of R_i within (8) can be used to minimise the norm of a wide variety of signals independently for the i^{th} agent, e.g. control effort, output velocity, acceleration or jerk.

III. COLLABORATIVE GRADIENT-BASED ILC

An explicit ILC design is now formulated which solves the decentralized ILC problem of Definition 1. Convergence properties for the multi-agent system are also derived.

A. Algorithm description

Algorithm 1: Let each agent have initial input $u_{i,0}$, $i = 1, 2, \cdots, n$, with associated tracking error e_0. The following decentralized gradient ILC update law

$$u_{i,k+1} = u_{i,k} + \gamma R_i^{-1} G_i^T Q e_k, \quad i = 1, 2, \cdots, n \tag{10}$$

with $\gamma > 0$ a suitably chosen scalar learning gain solves the collaborative tracking problem.

B. Convergence Properties

System output convergence properties of decentralized gradient-based ILC are derived in the following theorem:

Theorem 1: Application of Algorithm 1 to system (1) and (2) with a gain γ satisfying

$$0 < \gamma < \frac{2}{\rho(G R^{-1} G^T Q)} \tag{11}$$

where $R = \text{diag}\{R_1, \cdots, R_n\}$ and $\rho(\cdot)$ is the spectral radius of its argument, achieves monotonic reduction in tracking error norm, i.e.

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

and satisfies output convergence conditions (6), (9).

Proof: Applying (10) to (1) and (2) yields the output

$$y_{k+1} = y_k + \gamma \sum_{j=1}^{n} G_j R_j^{-1} G_j^T Q e_k \tag{12}$$

and from (12) the following relation is obtained

$$e_{k+1} = (I - \gamma \sum_{j=1}^{n} G_j R_j^{-1} G_j^T Q) e_k \tag{13}$$

and we note that

$$\sum_{j=1}^{n} G_j R_j^{-1} G_j^T = \underbrace{\begin{bmatrix} G_1 & G_2 & \cdots & G_n \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} R_1^{-1} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & R_n^{-1} \end{bmatrix}}_{R^{-1}} \underbrace{\begin{bmatrix} G_1^T \\ G_2^T \\ \vdots \\ G_n^T \end{bmatrix}}_{G^T}$$

It follows that a sufficient condition for monotonic convergence to zero error (i.e. (6), (9) with $y^* = y_d$) is

$$\|I - \gamma G R^{-1} G^T Q\| < 1 \tag{14}$$

which is equivalent to requiring the eigenvalues satisfy

$$0 < \gamma \rho(G R^{-1} G^T Q) < 2 \quad \Leftrightarrow \quad 0 < \gamma < \frac{2}{\rho(G R^{-1} G^T Q)}$$

■

This confirms that there always exists a suitably small γ which can be selected without knowledge of other agents.

Remark 2: Direct computation of $\rho(GR^{-1}G^TQ)$ may incur a heavy load which increases with N. However an upper bound can be estimated from individual agent bounds as

$$\begin{aligned} \rho(GR^{-1}G^TQ) &= \rho(G_1 R_1^{-1} G_1^T Q + \cdots + G_n R_n^{-1} G_n^T Q) \\ &\leq \rho(G_1 R_1^{-1} G_1^T Q) + \cdots + \rho(G_n R_n^{-1} G_n^T Q) \\ &= \|G_1(z)\|_\infty^2 + \|G_2(z)\|_\infty^2 + \cdots + \|G_n(z)\|_\infty^2 \end{aligned}$$

when scalar weighing matrices R_i, Q are chosen and the (along-the-trial) transfer-function of agent i is $G_i(z) = R_i^{-1} C_i (zI - A_i)^{-1} B_i Q$. It follows that a sufficient condition for (11) is

$$0 < \gamma < \frac{2}{\|G_1(z)\|_\infty^2 + \|G_2(z)\|_\infty^2 + \cdots + \|G_n(z)\|_\infty^2}.$$

C. Minimal control energy

Having established that Algorithm 1 satisfies output convergence properties (6), (9), we next confirm input convergence properties (7), (8) hold. First the required limiting input, u^*, is found as follows:

Theorem 2: Solution, u^*, to optimisation (8) is given by

$$u^* = R^{-1} G^T (G R^{-1} G^T)^{-1} y_d \tag{15}$$

with corresponding minimal input energy

$$J(u^*) = ||R^{-1} G^T (G R^{-1} G^T)^{-1} y_d||^2. \tag{16}$$

Proof: Embedding (1), (2) in (8) with $G = [G_1, G_2, \cdots, G_n]$ means that the optimality criterion is:

$$\min_{u \in \mathbb{R}^{nN}} \{J(u)\} = \min_{u \in \mathbb{R}^{nN}} \{\frac{1}{2} u^T R u + \lambda^T Q (G u - y_d)\} \tag{17}$$

where λ is the Lagrange multiplier. For each subsystem the initial input value $x_i(0)$ is 0. The solution satisfies

$$\left. \frac{\partial J(u)}{\partial u} \right|_{u=u^*} = R u^* + G^T Q \lambda = 0, \tag{18}$$

and recalling from (9) that $Gu^* = y_d$ it follows that

$$\begin{aligned} y_d = -G R^{-1} G^T Q \lambda \;\Rightarrow\; \lambda &= -(G R^{-1} G^T Q)^{-1} y_d \\ &= -Q^{-1} (G R^{-1} G^T)^{-1} y_d. \end{aligned} \tag{19}$$

Substituting (19) into (18) and manipulating produces the optimal input (15), with corresponding minimal input energy

$$\begin{aligned} J(u^*) &= ||R^{-1} G^T Q (G R^{-1} G^T Q)^{-1} y_d||^2 \\ &= ||R^{-1} G^T (G R^{-1} G^T)^{-1} y_d||^2. \end{aligned} \tag{20}$$

■

Having defined solution u^*, the next theorem establishes that Algorithm 1 solves the decentralized ILC problem.

Theorem 3: Let Algorithm 1 be applied to multi-agent system (1), (2) with γ satisfying (11) and the initial input for each agent given by $u_{i,0} = 0$, $i = 1, 2, \cdots, n$. Then the decentralized ILC problem of Definition 1 is solved, and in particular $y^* = y_d$ and u^* is given by (15).

Proof: Error relation (13) gives rise to relation

$$e_k = (I - \gamma G R^{-1} G^T Q)^k y_d$$

and from (10) and $u_{i,0} = 0$, $i = 1, 2, \cdots, n$, it follows that

$$\begin{aligned} u_{k+1} &= u_k + \gamma R^{-1} G^T Q (I - \gamma G R^{-1} G^T Q)^k y_d \\ &= \gamma R^{-1} G^T Q \sum_{p=0}^{k} (I - \gamma G R^{-1} G^T Q)^p y_d. \end{aligned}$$

Hence if (11) holds it follows from Theorem 2 that

$$\begin{aligned} \lim_{k \to \infty} u_{k+1} &= \gamma R^{-1} G^T Q (I - (I - \gamma G R^{-1} G^T Q))^{-1} y_d \\ &= \gamma R^{-1} G^T Q (\gamma G R^{-1} G^T Q)^{-1} y_d \\ &= R^{-1} G^T (G R^{-1} G^T)^{-1} y_d = u^* \end{aligned}$$

■

Theorem 3 confirms that Algorithm 1 solves the decentralized ILC problem, and thereby provides perfect tracking while minimising the total input norm $\sum_{i=1}^{n} \|u_{i,k}\|_{R_i}^2$. In the simplest case of $R_i = I$, this corresponds to the total energy of all agents. The next theorem shows how more general signal norms can be incorporated into the framework.

Theorem 4: Let the minimisation objective function within (8) of Algorithm 1 have the form

$$J(u_k) = \sum_{i=1}^{n} \|F_i u_{i,k}\|^2 \tag{21}$$

where F_i is the lifted representation of an along-the-trial operator $F_i(z)$. Then Theorem 3 holds if the components of R_i are selected as $R_i = F_i^T F_i$. Furthermore, if $F_i(z)$

is invertible then Algorithm 1 ILC update (10) can then be equivalently implemented by

$$u_{i,k+1} = u_{i,k} + \gamma H_i H_i^T G_i^T Q e_k, \quad 1, 2, \cdots n \tag{22}$$

in which H_i is the lifted representation of $H_i(z) = F_i(z)^{-1}$.

Proof: For any full rank matrix A, $(A^\top A)^{-1} = A^{-1}(A^{-1})^\top$. Unlifted and lifted inverses satisfy $F(z) = H(z)^{-1} \Leftrightarrow F = H^{-1}$. ∎

Implementation (22) is especially useful when minimising the norm of velocity, acceleration and jerk input and output signals due to the simple form of $H_i(z)$. The next section considers the alternative case in which energy is required to be balanced between agents.

IV. Balancing the workload between agents

The previous section shows that the proposed decentralised gradient ILC algorithm solves the collaborative tracking problem. Furthermore, with appropriate choice of initial inputs, the algorithm will converge to the minimum control energy solution, which is appealing in practice. However, the total control energy required might not be uniformly distributed among all agents. In this section, an energy balancing algorithm is proposed such that the required energy to achieve the collaborative tracking is uniformly distributed. This is achieved by appropriate selection of the weighting parameters $R_i, i = 1, \dots, n$ in the norm definition. For notational simplicity, the widely used scalar weighting, i.e.

$$R_i = r_i \times I$$

where r_i is a positive scalar and I is an identity matrix of appropriate dimensions, is now considered.

Theorem 5: Applying Algorithm 1 to system (1) and (2) with zero initial inputs for all agents, the following choice of weighting matrices

$$R_i = r_i^* \times I,\ i = 1, \dots, n$$

and γ satisfying (11) solves the collaborative tracking problem with the additional property that the converged input energy is balanced among the agents, i.e.

$$||u_i^*||_{R_i}^2 = c^*,\ \forall 1 \le i \le n \tag{23}$$

where $r_i^*, i = 1, \dots, n$ and c^* are solutions of the following optimization problem:

$$(r_1^*, \dots, r_n^*, c^*) = \arg \min_{r_i > 0, c > 0} \{\|h(r_1, \dots, r_n) - c \times \vec{1}\|\}$$

in which $h(r_1, \dots, r_n) \in R^n$ with ith element defined as

$$h_i(r_1, \dots, r_n) = \left\| M_i \begin{pmatrix} r_1^{-1}I & & \\ & \ddots & \\ & & r_n^{-1}I \end{pmatrix} G^T Q \times \left(G \begin{pmatrix} r_1^{-1}I & & \\ & \ddots & \\ & & r_n^{-1}I \end{pmatrix} G^T Q \right)^{-1} y_d \right\|_{r_i I}^2 \tag{24}$$

where $M_i = [0, \dots, I, \dots, 0]$ is a block matrix (compatible with G) whose ith block is an identity matrix and all others zero matrices, and

$$\vec{1} = [1, 1, \dots, 1]^T \in R^n.$$

Proof: The first part of the theorem follows directly from Theorem 3. To show the second part, i.e. with the chosen weighting parameter the converged energy is balanced between the agents, first note that the optimal value of the optimisation problem is zero. We show this by contradiction. If this is not the case, that is, the optimal value is $\beta^* > 0$ with optimal solution r_i^* and c^*, there must exist αr_i^* and αc^* with a performance index value $\alpha\beta^*$ (by some algebraic operations of the performance index). Choosing $0 < \alpha < 1$ produces a performance index value less than β^*, contradicting the assumption that β^* is optimal. Therefore, optimal value of the optimisation problem is zero.

Now note that the performance index is the norm of the difference between the required energy of each agent. The optimal value of zero implies they all have the same converged energy, i.e. (23). This completes the proof. ∎

The above theorem provides a way to balance the energy between the agents in the minimum control energy solution. It is worth pointing out that the optimisation problem in Theorem 4 generally does not admit a closed form solution. To solve it, standard optimisation tools can be used, e.g. Matlab $fmincon$ solver. Also note the optimisation problem does not have a unique solution either - any solution can be used to meet the energy balancing requirement.

V. Numerical Simulations

In this section, numerical examples are given to demonstrate the effectiveness of the proposed algorithm.

A. Convergence performance of the proposed algorithm

First, the example in [7] is used, which comprises a multiple actuator system manipulating a mass M using n actuators. Each agent (actuator) has dynamics G_i and generates an individual force y_i and the total force generated is $y = \sum_{i=1}^n y_i$. In this example, $n = 4$ and each agent's dynamics are assumed to be the first order system

$$G_i(s) = \frac{\tau_i}{\tau_i s + 1} \tag{25}$$

where $\tau_i = 1 - \frac{0.5}{i}$, $i = 1, 2, 3, 4$. The system is sampled using a zero-order hold and a sampling time of $0.1s$ The trial length is $6s$. Zero initial conditions are assumed for all agents and the desired total output is $y_d(t) = \sin(t)$.

The tracking error norm convergence performance of Algorithm 1 over 40 iterations with weighting matrices $R_i = I, 1 \le i \le n$, $Q = I$ and $\gamma = 0.1, 0.2, 0.4, 0.65$ (respectively) is shown in Fig. 1. From the figure, it can be seen that the proposed algorithm solves the collaborative tracking problem and achieves monotonic convergence in the tracking error norm. The system's total output on a few representative iterations (for $\gamma = 0.65$) is shown in Fig. 2,

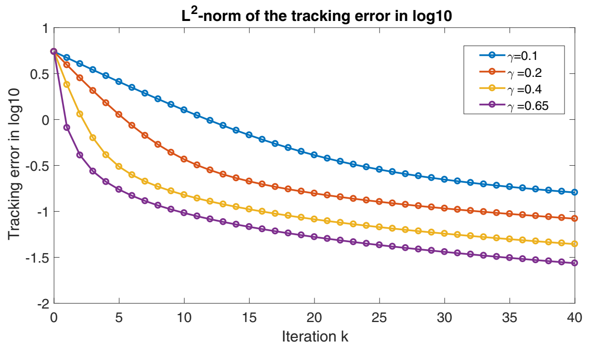

Fig. 1. Tracking error norm convergence with $\gamma = 0.1, 0.2, 0.4, 0.65$

Fig. 2. Total system output tracking results with $\gamma = 0.65$

together with the desired reference, which again confirms the above observations.

The system's total input energy over 40 iterations is shown in Fig. 3, together with the theoretical minimum control energy obtained using (16). It can be seen that the total input energy converges to the theoretical minimum energy value, verifying the predictions in Theorem 3.

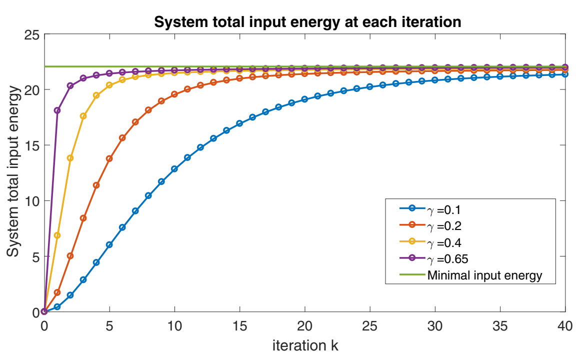

Fig. 3. Total system input energy with $\gamma = 0.1, 0.2, 0.4, 0.65$

A close inspection of the above results shows that although perfect tracking of the reference can be achieved and that minimum energy solution is obtained, the distribution of the energy needed among the agents is not uniform. This can be seen in Fig. 4 for $\gamma = 0.65$ which shows the distribution of each agent's control energy over 40 iterations. It can be seen that agent 4 provides the most energy to complete the collaborative tracking task while agent 1 provides the least. This could potentially cause problems, e.g. overloading and hence damaging agent 4 while agent 1 is underloaded and can be better utilised to avoid this happening.

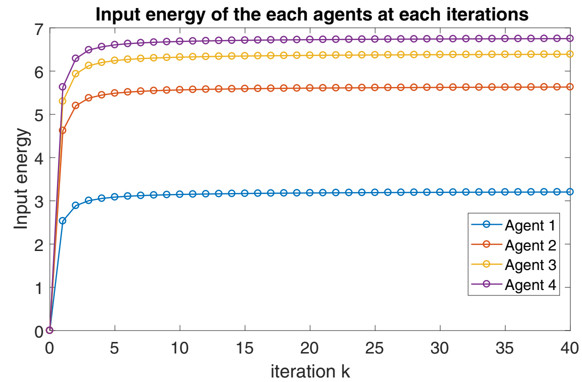

Fig. 4. Input energy for each agent with $\gamma = 0.65$

To address this issue, the energy balancing algorithm proposed in Section IV is used. Here weighting parameters

$$[r_1, r_2, r_3, r_4] = [1, 0.75574674, 0.70951277, 0.69010607]$$

are a solution of the optimisation problem in Theorem 5 and the same choice of Q, γ. The energy evolution of each agent is shown in Fig. 4. From this figure, it is clear that the balancing algorithm performs very well and the input energies are almost identical for each agent, verifying the results in Theorem 5.

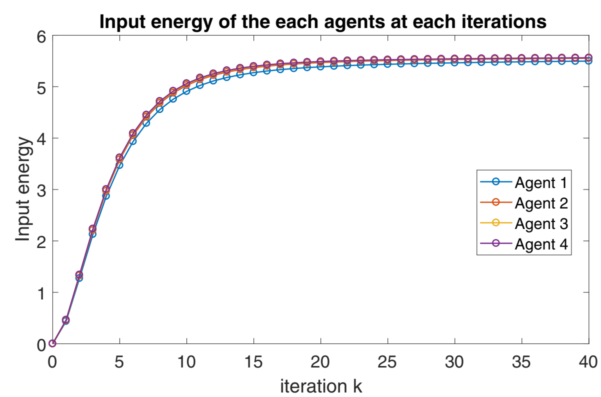

Fig. 5. Input energy for each agent

B. Non-minimum phase system

The proposed gradient-based ILC algorithm can be applied to non-minimum phase (NMP) systems. Note that the inverse based ILC algorithm will have difficulties due to the unstable nature of the inverse. Consider the following NMP system

$$G_i(s) = \frac{3\tau_i - s}{s^2 + 6\tau_i s + 5} \tag{26}$$

where $\tau_i = 1 - \frac{0.5}{i}$, $i = 1, 2, 3, 4$. All other parameter settings are identical to the previous examples and the tracking error norms are shown in Fig. 6 demonstrating good tracking performance.

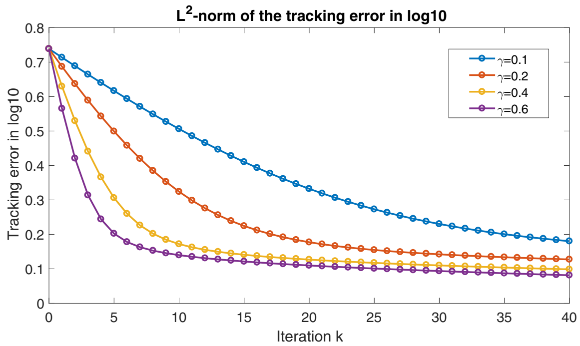

Fig. 6. Tracking error norm for NMP agents by gradient ILC with $\gamma = 0.1, 0.2, 0.4, 0.6$

C. Robustness of the proposed algorithm

In addition to achieving minimum control energy, balancing agent workload and being applicable to NMP systems, a particular advantage of the proposed algorithm over the inverse based ILC [7] is that it is less sensitive to model uncertainties. This is demonstrated by the example below.

Consider the following system

$$G_i(s) = \frac{s + 3\tau_i}{s^2 + 6\tau_i s + 5} \tag{27}$$

where $\tau_i = 1 - \frac{0.5}{i}$, $i = 1, 2, 3, 4$. Assume that only the following model (same for all agents) is available for ILC design

$$G_i(s) = \frac{0.1s + 3}{s^2 + 6s + 5}.$$

Applying inverse based ILC algorithm and the proposed gradient based ILC algorithm with the same learning gain $\gamma = 0.25$ (and the same settings for the gradient algorithm as the previous examples) produces the tracking error norms shown in Fig. 7.

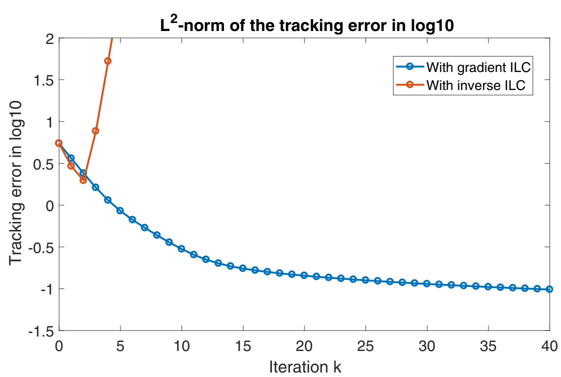

Fig. 7. Tracking error norm convergence for inverse based and gradient based ILC

From Fig. 7, we can see that with model uncertainties, the proposed gradient algorithm still achieves monotonic convergence in the tracking error norm and solves the collaborative tracking problem. On the other hand, the tracking error norm of inverse based ILC initially decreases but quickly diverges after two iterations, illustrating that the proposed gradient based algorithm is less sensitive to model uncertainties than inverse based ILC, which is often crucial in practical applications.

VI. CONCLUSIONS AND FUTURE WORK

This paper proposes a general decentralised ILC design framework for the collaborative tracking problem where a number of agents work together to achieve a global objective. The proposed design framework enables a wide range of new ILC algorithms to be developed. To exemplify the proposed design framework, a decentralised gradient based ILC algorithm is developed. The proposed algorithm achieves perfect collaborative tracking of the desired reference, and at the same time minimum input energy with appropriately chosen initial inputs. We further develop an energy balancing algorithm to ensure the input energy among the agents is uniformly distributed so that the no agent is overloaded/underloaded. The performance of the proposed algorithm is verified via numerical simulations.

Compared to the existing inverse based ILC algorithm, the proposed gradient based ILC is less sensitive to model uncertainties, as partially demonstrated by numerical simulations. However, a rigorous examination of the algorithm's robustness properties is needed. Furthermore, system constraints, e.g. input saturation, which are widely existing in practice, has not been considered in this paper. Techniques in constrained ILC design, e.g. [12] can be used to incorporate the constraints into the design. These form part of future research and will be reported separately.

REFERENCES

[1] S. Knorn, Z. Chen, and R. H. Middleton. Collective control of multiagent systems. *IEEE Transactions on Control of Network Systems*, 3(4):334–347, 2016.

[2] S. Zheng, Q. Zhang, R. Zheng, B. Huang, Y. Song, and X. Chen. Combining a multi-agent system and communication middleware for smart home control: A universal control platform architecture. *Sensors*, 2135:1–25, 2017.

[3] J. Choi, S. Oh, and R. Horowitz. Distributed learning and cooperative control for multi-agent systems. *Automatica*, 45(12):2802–2814, 2009.

[4] W. Chen, S. Hua, and S. S. Ge. Consensus-based distributed cooperative learning control for a group of discrete-time nonlinear multi-agent systems using neural networks. *Automatica*, 50(9):2254–2268, 2014.

[5] D. Mirza, P. Naughton, C. Schurgers, and R. Kastner. Real-time collaborative tracking for underwater networked systems. *Ad Hoc Networks*, 34:196–210, 2015.

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

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

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

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

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

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

[12] B. Chu, C. T. Freeman, and D. H. Owens. A novel design framework for point-to-point ilc using successive projection. *IEEE Transactions on Control Systems Technology*, 23(3):1156–1163, 2015.