



*Research article*

# Distributed data-driven iterative learning control for multi-agent systems with unknown input-output coupled parameters

Duhui Chang and Yan Geng\*

School of Science, Xi'an Polytechnic University, Xi'an 710048, China

\* **Correspondence:** Email: 20180607@xpu.edu.cn.

**Abstract:** This article studies a distributed data-driven iterative learning control (ILC) strategy based on the identified input–output coupled parameters (IOCPs) to address the consensus trajectory tracking problem of discrete time-varying multi-agent systems (MASs). First, by leveraging the repeatability of the control system, a special learning scheme is designed by using system input and output data to identify the unknown IOCPs. Then the reciprocal of the identified IOCPs is selected as the learning gain to construct the ILC law of the MASs. Second, the case of measurement noise in the MASs is considered, where the maximum allowable control deviation is incorporated into the learning mechanism for identification of the IOCPs, thereby minimizing adverse effects of the noise on the learning scheme's performance and bolstering robustness. Finally, three numerical simulations are employed to validate the effectiveness of the designed IOCP identification method and iterative learning control strategy.

**Keywords:** iterative learning control; IOCPs; multi-agent systems; parameter identification

---

## 1. Introduction

With the development and advancement of control technology, iterative learning control (ILC) was first proposed as an intelligent learning mechanism in [1] to enhance the efficiency of systems. The core idea of ILC is that the previously collected input–output data from the repeatedly operated dynamic system are utilized to produce the control inputs for the subsequent operation. This process aims to obtain the subsequent control input of the system, ultimately achieving precise tracking of the desired trajectory. Due to the simple and effective structure, ILC has been successfully applied to

autonomous vehicles [2], robot manipulators [3], mechatronic systems [4], computer numerical control machine tool [5], multi-agent systems (MASs) [6], and other fields.

On the other hand, the level of automation in the production process has increased and the cooperative control of MASs has garnered significant attention and interest. These systems have the advantages of autonomy, distribution, coordination, and learning ability, and they have been widely used in the fields of multi-robot systems [7], power-supply systems [8], traffic control [9], distributed coordination [10], smart grids [11], etc. These papers primarily concentrate on how to devise the appropriate control program to attain consensus control within the system.

Since the ILC method was first introduced into multi-agent formation in [12], the advantages of using the distributed ILC method to address the consensus tracking issue in MASs have been analyzed. MASs improve work efficiency through communication and coordination between agents, constantly adjusting and updating the behavior, enabling all agents to converge or achieve consistent tracking of the desired trajectory within a specified timeframe. The application of ILC to MASs promoted research into the consensus tracking problem. Subsequently, many researchers have studied the cooperative control problem of repetitive systems. Meng and Jia [13] researched the application of a P-type ILC to MASs, and Dai et al. [14] studied the P-type ILC to solve the time-delay problem of MASs. In a few words, the ILC method has shown good performance in previous MAS control. Furthermore, several multi-agent control approaches rooted in ILC have proven effective in practical applications [15].

In particular, the improvement of actual production requirements has led to upgrading of the system's complexity, and the complexity of the system has also escalated. Numerous problems such as unknown models, unknown or uncertain system parameters, measurement noise, and other problems have affected the development and application of MASs. For the nonlinear system, Hou and Jin [16] proposed a model-free adaptive iterative learning control method to control the system to complete the tracking task. This method was introduced into the MASs in Bu et al. [17] to solve the trajectory tracking problem, and a system with iteration-varying topologies was considered. For a linear system with unknown parameters, Liu and Ruan [18] constructed an adaptive parameter estimation algorithm by using the input and output data of the system in the form of the Markov matrix. In a similar way, Geng et al. [19] transformed a class of multi-phase batch processes into a switched linear system, and then used the strategy of ILC to control the system. Furthermore, Lin et al. [20] proposed a point-to-point iterative learning control strategy that addresses the optimal consensus problem at specified data points for heterogeneous networked agents with iteration-switching topologies. Hence, how to design an appropriate ILC scheme for the MASs with unknown parameters is a challenging problem.

Within the conventional framework of ILC, the identified input–output coupled parameters (IOCPs) are defined as the resulting product of the output and input matrices pertaining to a single-input–single-output (SISO) system, serving as the only system information for ILC construction. Zhang et al. [21] designed an iterative learning mechanism that exploits the inherent repeatability of the control system to estimate the unknown IOCPs accurately. This mechanism involves running the repetitive system twice under controlled conditions. By comparing the system's outputs during these two runs, the researchers can infer the IOCP information, which is then used to design an effective ILC algorithm. This approach allows for the implementation of ILC in systems where the IOCP is not known beforehand, thereby enhancing the applicability and flexibility of ILC techniques. On the basis of the previous research, Liu et al. [22] formulated an iterative learning approach for determining an unknown IOCP by utilizing the repeatability of the control system and input–output data. Additionally, they introduced a refinement of this scheme to enhance its robustness against various disturbances,

such as measurement noise, system noise, and initial state variations. This enhancement incorporates a maximum allowable control deviation to ensure that the learning process remains stable and robust in practical applications. Most of the existing work prioritizes the study of uncertain systems for parameter identification. Inspired by this, we considered the appropriate iterative learning mechanism to identify the IOCPs and then establish the ILC algorithm by designing an appropriate learning gain to ensure that the system can accurately track the desired trajectory.

For the MASs, distributed ILC achieves control of MASs where only a subset of agents has direct access to the desired trajectory and only neighboring agents can communicate. Hock and Schoellig [23] focused on enabling a team of quadrotors to track a desired trajectory within a given formation. They presented a distributed ILC approach and proved the stability of any causal learning function when its gains meet a scalar condition. Pakshin et al. [24] designed an ILC law for the MASs with random perturbations, based on minimizing the deviations from a reference model and also based on the theory of stochastic stability of repetitive processes. Inspired by this, we were able to devise a distributed data-driven ILC strategy with the aim of effectively addressing MASs. This approach utilizes data analytics and advanced control algorithms to ensure the MASs achieve optimal performance in complex and dynamic conditions.

The primary focus of this article is as follows. At first, a distributed data-driven iterative learning scheme is designed, based on input and output signals to identify the IOCPs for time-varying discrete MASs with unknown parameters. To tackle the consensus trajectory tracking problem, we design an ILC by selecting the reciprocal of the IOCPs as the learning gain. Second, we analyze MASs with random measurement noise by introducing a maximum allowable control deviation. The approach mitigates the adverse effects of noise on the iterative learning strategy's performance and enhances the robustness of the mechanism.

The primary achievements of this paper can be summarized as follows:

1) The discrete linear MASs are transformed into the lower triangular parameter matrix form; this simplifies the system's structure for further analysis. An  $N$ -step iterative learning mechanism is devised, leveraging repetitive system traits and input–output data to identify the unknown IOCPs. This data-driven approach suits complex systems with uncertainties and improves the accuracy of identifying the parameters.

2) Considering the measurement noise prevalent in industrial applications, a strategy is introduced by setting the maximum allowable control deviation. This minimizes noise's adverse impact on the learning scheme's performance to bolster the system's robustness under noisy conditions.

3) The reciprocal of the identified IOCPs is selected as the learning gain. The proposed ILC is constructed, based on the input–output data of the system, and also features a simple structure and strong applicability, thus enhancing the system's practicability and robustness. Rigorous mathematical proofs demonstrate that the designed distributed data-driven ILC strategy is capable of achieving consistent trajectory tracking.

The organization of the remainder of this paper is as follows. Section 2 introduces the MASs and graph theory. Section 3 describes the IOCP identification mechanism and ILC for MASs. Section 4 discusses the case of measurement noise in the system. The simulation and conclusion of this paper are given in Sections 5 and 6, respectively.

## 2. Preliminaries and problem formulation

### 2.1. Preliminaries

The communication topology of MASs can be represented and described as a graph  $\zeta = (v, \varepsilon, \mathcal{A})$  with the nodes  $v = \{v_1, v_2 \cdots v_M\}$  and the edges  $\varepsilon \subseteq v \times v$ . The matrix  $\mathcal{A}$  is the adjacency matrix of the graph  $\zeta$ ; its main diagonal entries are all 0. When agent  $i$  is connected to agent  $j$ ,  $a_{ij} \neq 0$ , or else  $a_{ij} = 0$ . Note that  $\mathcal{D}$  is the degree matrix of graph  $\zeta$ ,  $\mathcal{D} = \text{diag}(d_1, d_2, \dots, d_M)$ , where  $d_i = \sum_{j \in M_i} a_{ji}$  (the  $j \in M_i$  represents the set of agents associated with agent  $i$ ). The Laplacian matrix of graph  $\zeta$  is  $\mathcal{L} = \mathcal{D} - \mathcal{A}$ .

The virtual leader is denoted as Agent 0. The graph  $\bar{\zeta}$  contains  $M$  agents and one virtual leader. In the graph  $\bar{\zeta}$ , if the agent  $i$  and the leader 0 are connected,  $s_i \neq 0$ ; otherwise,  $s_i = 0$ . Here,  $S = \text{diag}(s_1, s_2, \dots, s_M)$  denotes the degree matrix of graph  $\bar{\zeta}$ . We write  $P = \mathcal{L} + S$ .

**Assumption 1:** In this paper, the graph  $\zeta$  is considered to be connected, and  $S$  has at least one nonzero diagonal element term.

**Remark 1.** In a connected graph  $\zeta$ , any two agents are linked by a path. The matrix  $S$  must possess at least one nonzero diagonal element, implying that there must be at least one agent capable of receiving the target trajectory signal from the leader.

### 2.2. Problem formulation

The linear discrete time-varying SISO MAS with  $M$  agents is described as

$$\begin{cases} \mathbf{x}_{j,k}(t+1) = \mathbf{A}_j(t)\mathbf{x}_{j,k}(t) + \mathbf{b}_j(t)u_{j,k}(t), t \in T, \\ y_{j,k}(t) = \mathbf{c}_j(t)\mathbf{x}_{j,k}(t), t \in T^+, \end{cases} \quad (2.1)$$

where  $T = \{0, 1, \dots, N-1\}$ ,  $T^+ = \{1, 2, \dots, N\}$ ,  $N$  denotes the total sampling number,  $j = 1, 2, \dots, M$  is the label number of the agents,  $t$  and  $k$  represent time and the iteration number,  $\mathbf{x}_{j,k}(t) \in R^n$  is the system's state,  $u_{j,k}(t) \in R$  is the control input, and  $y_{j,k}(t) \in R$  denotes the system output of the agent  $j$ .  $\mathbf{A}_j(t) \in R^{n \times n}$ ,  $\mathbf{b}_j(t) \in R^{n \times 1}$  and  $\mathbf{c}_j(t) \in R^{1 \times n}$  are the unknown constant matrix, column vector, and row vector, respectively. Given a desired trajectory  $y_d(t)$ ,  $t \in T^+$ , the distributed data-driven ILC scheme is constructed under the following two basic assumptions.

**Assumption 2:** For the MAS (2.1), a given desired trajectory  $y_d(t)$ ,  $t \in T^+$  is realizable. For any specified desired state  $\mathbf{x}_{j,d}(t)$ ,  $t \in T^+$ , there is always a particular control input  $u_{j,d}(t)$ ,  $t \in T$  that can be found to achieve that state:

$$\begin{cases} \mathbf{x}_{j,d}(t+1) = \mathbf{A}_j(t)\mathbf{x}_{j,d}(t) + \mathbf{b}_j(t)u_{j,d}(t), t \in T, \\ y_d(t) = \mathbf{c}_j(t)\mathbf{x}_{j,d}(t), t \in T^+, \end{cases} \quad (2.2)$$

which implies that the IOCPs are  $\mathbf{c}_j(t+1)\mathbf{b}_j(t) \neq 0$ ,  $t \in T$ .

**Assumption 3:** The initial condition  $\mathbf{x}_{j,k}(0)$  fulfills the condition  $\mathbf{x}_{j,k}(0) = \mathbf{x}_{j,0}$  for all  $k$ , where  $\mathbf{x}_{j,0}$  is an arbitrarily given point. In other words, the initial state is precisely reset at the start of each iteration. Without loss of generality, we set  $\mathbf{x}_{j,k}(0) = 0$ .

**Remark 2.** Assumption 3 is a fundamental condition for ILC, which is used to guarantee the consistency of tracking in the ILC algorithm. Definitely, in industrial engineering applications, the system's initial state shift is a common problem. On the other hand, the repetitive initial state is a fundamental postulation in theoretical research on ILC. In this paper, we only consider the initial state without a shift.

For the MASs, since the parameters of the system are unknown, by leveraging the system's repeatability and designing an appropriate iterative learning strategy, we can obtain the exact values of the unknown IOCPs on the basis of the input and output data. On the basis of the assumptions above, we can design an iterative learning mechanism for the IOCPs of unknown systems and construct the ILC law for the MASs to achieve the consensus tracking task.

### 3. Distributed data-driven iterative learning control for the system

#### 3.1. $N$ -step iterative learning identification for IOCPs

In this section, we design an appropriate iterative learning mechanism for each agent to identify the IOCPs.

**Theorem 1:** For the MAS in (2.1), if each agent abides by Assumptions 2 and 3, then the IOCPs  $\{\mathbf{c}_j(1)\mathbf{b}_j(0), \mathbf{c}_j(2)\mathbf{b}_j(1), \dots, \mathbf{c}_j(N)\mathbf{b}_j(N-1)\}$  can be obtained by constructing an appropriate iterative learning mechanism.

**Proof:** First, define the super-vectors as follows:

$$\mathbf{u}_{j,k} = [u_{j,k}(0), u_{j,k}(1), \dots, u_{j,k}(N-1)]^T,$$

$$\mathbf{y}_{j,k} = [y_{j,k}(1), y_{j,k}(2), \dots, y_{j,k}(N)]^T.$$

Thus, the system in (2.1) can be characterized as

$$\mathbf{y}_{j,k} = \mathbf{G}_j \mathbf{u}_{j,k}, \quad (3.1)$$

$$\text{where } \mathbf{G}_j = \begin{pmatrix} \mathbf{c}_j(1)\mathbf{b}_j(0) & 0 & \dots & 0 \\ \mathbf{c}_j(2)\mathbf{A}_j(1)\mathbf{b}_j(0) & \mathbf{c}_j(2)\mathbf{b}_j(1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_j(N) \prod_{p=1}^{N-1} \mathbf{A}_j(p) \mathbf{b}_j(0) & \mathbf{c}_j(N) \prod_{p=2}^{N-1} \mathbf{A}_j(p) \mathbf{b}_j(1) & \dots & \mathbf{c}_j(N)\mathbf{b}_j(N-1) \end{pmatrix}.$$

Next, we introduce each step of the iterative learning mechanism.

Step 0: The control input signals of each agent are arbitrarily given as

$$\mathbf{u}_{j,0} = \{u_{j,0}(0), u_{j,0}(1), \dots, u_{j,0}(N-1)\}, \quad (3.2)$$

and the outputs of MAS (2.1)  $\mathbf{y}_{j,0} = \{y_{j,0}(1), y_{j,0}(2), \dots, y_{j,0}(N)\}$  are generated by

$$\mathbf{y}_{j,0} = \mathbf{G}_j \mathbf{u}_{j,0}. \quad (3.3)$$

Step 1: Select the control input signals  $\mathbf{u}_{j,1} = \{u_{j,1}(0), u_{j,1}(1), \dots, u_{j,1}(N-1)\}$  that satisfy

$$\mathbf{u}_{j,1} = \{u_{j,0}(0) + 1, u_{j,0}(1), u_{j,0}(2), \dots, u_{j,0}(N-1)\},$$

and the outputs of MAS (2.1)  $\mathbf{y}_{j,1} = \{y_{j,1}(1), y_{j,1}(2), \dots, y_{j,1}(N)\}$  are generated by

$$\mathbf{y}_{j,1} = \mathbf{G}_j \mathbf{u}_{j,1}. \quad (3.4)$$

Subtracting (3.3) from (3.4) yields

$$\mathbf{y}_{j,1} - \mathbf{y}_{j,0} = \mathbf{G}_j (\mathbf{u}_{j,1} - \mathbf{u}_{j,0}) \Rightarrow y_{j,1}(1) - y_{j,0}(1) = c_j(1)b_j(0). \quad (3.5)$$

Step 2: Select the control input signals  $\mathbf{u}_{j,2} = \{u_{j,2}(0), u_{j,2}(1), \dots, u_{j,2}(N-1)\}$  that satisfy

$$\mathbf{u}_{j,2} = \{u_{j,0}(0), u_{j,0}(1) + 1, u_{j,0}(2), \dots, u_{j,0}(N-1)\}, \quad (3.6)$$

and the outputs of MAS (2.1)  $\mathbf{y}_{j,2} = \{y_{j,2}(1), y_{j,2}(2), \dots, y_{j,2}(N)\}$  are generated by

$$\mathbf{y}_{j,2} = \mathbf{G}_j \mathbf{u}_{j,2}. \quad (3.7)$$

Subtracting (3.3) from (3.7) gives

$$\mathbf{y}_{j,2} - \mathbf{y}_{j,0} = \mathbf{G}_j (\mathbf{u}_{j,2} - \mathbf{u}_{j,0}) \Rightarrow y_{j,1}(2) - y_{j,0}(2) = c_j(2)b_j(1). \quad (3.8)$$

.....

Step  $N$ : Select the control input signals  $\mathbf{u}_{j,N} = \{u_{j,N}(0), u_{j,N}(1), \dots, u_{j,N}(N-1)\}$  that satisfy

$$\mathbf{u}_{j,N} = \{u_{j,0}(0), u_{j,0}(1), \dots, u_{j,0}(N-1) + 1\}, \quad (3.9)$$

and the outputs of MAS (2.1)  $\mathbf{y}_{j,N} = \{y_{j,N}(1), y_{j,N}(2), \dots, y_{j,N}(N)\}$  are generated by

$$\mathbf{y}_{j,N} = \mathbf{G}_j \mathbf{u}_{j,N}. \quad (3.10)$$

Subtracting (3.3) from (3.10) gives

$$\mathbf{y}_{j,N} - \mathbf{y}_{j,0} = \mathbf{G}_j (\mathbf{u}_{j,N} - \mathbf{u}_{j,0}) \Rightarrow y_{j,N}(N) - y_{j,0}(N) = c_j(N)b_j(N-1). \quad (3.11)$$

According to the steps above, when the initial states of each agent in the system are fixed, we can obtain the IOCPs  $\{\mathbf{c}_j(1)\mathbf{b}_j(0), \mathbf{c}_j(2)\mathbf{b}_j(1), \dots, \mathbf{c}_j(N)\mathbf{b}_j(N-1)\}$  of each agent accurately in the MAS by applying the iterative learning mechanism.

**Remark 3.** Here, for  $t \in T^+$ , it is obvious that  $y_{j,t}(t+1) - y_{j,0}(t+1) = \mathbf{c}_j(t+1)\mathbf{b}_j(t)$ . After running the system  $N$  times, we can accurately get the unknown IOCPs of the system. This iterative learning mechanism can only be applied to systems with fixed initial states and in the absence of noise. We will continue to study other types of systems.

### 3.2. Distributed data-driven iterative learning control

In the context of iterative learning control, the control input for subsequent iterations is refined through learning from the input data and errors encountered in the preceding iteration. Let  $\mathbf{e}_{j,k}(t+1) = y_d(t+1) - y_{j,k}(t+1)$  denote the tracking error, where  $y_d(t+1)$  is the desired trajectory of the system and can be regarded as the virtual leader. For the MAS (2.1), when combined with the communication topology, the agent's consistency tracking error can be described as  $\xi_{j,k}(t+1)$ , defined as

$$\xi_{j,k}(t+1) = \sum_{i \in M_j} a_{i,j} (y_{i,k}(t+1) - y_{j,k}(t+1)) + s_j (y_d(t+1) - y_{j,k}(t+1)). \quad (3.12)$$

This can be reformulated to express it in terms of tracking errors

$$\xi_{j,k}(t+1) = \sum_{i \in M_j} a_{i,j} (\mathbf{e}_{j,k}(t+1) - \mathbf{e}_{i,k}(t+1)) + s_j \mathbf{e}_{j,k}(t+1). \quad (3.13)$$

Even though the parameters of the MASs remain unknown, the previously identified unknown IOCPs of these systems are utilized. The reciprocal of the IOCPs is selected as the learning gain for designing ILC to control the MASs. In our study, we focus on the iterative learning control strategy applicable to MASs, which is designed utilizing the IOCPs as follows:

$$u_{j,k+1}(t) = u_{j,k}(t) + \frac{1}{y_{j,t}(t+1) - y_{j,0}(t+1)} \xi_{j,k}(t+1), (t \in T^+). \quad (3.14)$$

Before proving Theorem 2, it is essential to introduce the definition of the  $\lambda$ -norm.

**Definition:** The  $\lambda$ -norm of the discrete-time vector function  $\mathbf{h}: \{1, 2, \dots, N\} \rightarrow R^n$  is defined as

$$\|\mathbf{h}(t)\|_\lambda = \sup_{t \in \{1, 2, \dots, N\}} \{\lambda^t \|\mathbf{h}(t)\|\}, \quad 0 < \lambda < 1,$$

where  $\|\cdot\|$  is a vector norm on  $R^n$ .

**Theorem 2:** Suppose that the MAS (2.1) satisfies Assumptions 1–3. If the communication topology matrix of the MAS (2.1) satisfies  $\|(I_{NM} - (P \otimes I_N))\| = \rho < 1$ , then the iterative learning control strategy (3.14) can precisely track the desired trajectory.

**Proof:** For convenience of the proof, the system (2.1) can be solved by the induction method

$$\mathbf{x}_{j,k}(t) = \Phi_j(t, 0)\mathbf{x}_{j,k}(0) + \sum_{s=0}^{t-1} \Phi_j(t, s+1)\mathbf{b}_j(s)\mathbf{u}_{j,k}(s), \quad (3.15)$$

where  $\Phi_j(t, s)$  is the state transition matrix, which is determined by  $A_j(t)$ , and  $\Phi_j(t, s)$  satisfies

$$\Phi_j(t, s) = A_j(t-1)A_j(t-2) \cdots A_j(s), t > s \text{ and } \Phi_j(s, s) = I. \quad (3.16)$$

For the convenience of analysis, we write

$$\mathbf{x}_k(t) = [\mathbf{x}_{1,k}(t), \mathbf{x}_{2,k}(t), \dots, \mathbf{x}_{M,k}(t)]^T,$$

$$\Phi(t, s) = \text{diag}(\Phi_1(t, s), \Phi_2(t, s), \dots, \Phi_M(t, s)),$$

$$\mathbf{u}_k(t) = [u_{1,k}(t), u_{2,k}(t), \dots, u_{M,k}(t)]^T,$$

$$\mathbf{y}_d(t) = [y_{d,1}(t), y_{d,2}(t), \dots, y_{d,M}(t)]^T,$$

$$\mathbf{y}_k(t) = [y_{1,k}(t), y_{2,k}(t), \dots, y_{M,k}(t)]^T,$$

$$\mathbf{e}_k(t+1) = [e_{1,k}(t), e_{2,k}(t), \dots, e_{M,k}(t)]^T,$$

$$\mathbf{C}(t+1) = \text{diag}(\mathbf{c}_1(t+1), \mathbf{c}_2(t+1), \dots, \mathbf{c}_M(t+1)).$$

The compact representation can also be utilized to express the solution of the system (2.1)

$$\mathbf{x}_k(t) = \Phi(t, 0)\mathbf{x}_k(0) + \sum_{s=0}^{t-1} \Phi(t, s+1)\mathbf{b}(s)\mathbf{u}_k(s), \quad (3.17)$$

The agent consistency tracking error (3.13) and the iterative learning control law (3.14) can be expressed as follows:

$$\xi_k(t+1) = (P \otimes I_N)\mathbf{e}_k(t+1), \quad (3.18)$$

$$\mathbf{u}_{k+1}(t) = \mathbf{u}_k(t) + \mathbf{H}(t)(P \otimes I_N)\mathbf{e}_k(t+1), \quad (3.19)$$

where  $\mathbf{H}(t) = \text{diag}\left(\frac{1}{y_{1,t}(t+1)-y_{1,0}(t+1)}, \frac{1}{y_{2,t}(t+1)-y_{2,0}(t+1)}, \dots, \frac{1}{y_{M,t}(t+1)-y_{M,0}(t+1)}\right)$ .

By calculating the deviation in tracking,

$$\begin{aligned} \mathbf{e}_{k+1}(t+1) &= \mathbf{y}_d(t+1) - \mathbf{y}_k(t+1) \\ &= \mathbf{e}_k(t+1) - \mathbf{c}(t+1)[\mathbf{x}_{k+1}(t+1) - \mathbf{x}_k(t+1)] \\ &= \mathbf{e}_k(t+1) - \mathbf{C}(t+1) \sum_{s=0}^t \Phi(t+1, s+1)\mathbf{b}(s)(\mathbf{u}_{k+1}(s) - \mathbf{u}_k(s)) \\ &= \mathbf{e}_k(t+1) - \mathbf{C}(t+1) \sum_{s=0}^t \Phi(t+1, s+1)\mathbf{b}(s)\mathbf{H}(s)(P \otimes I_N)\mathbf{e}_k(s+1) \end{aligned} \quad (3.20)$$

By taking out the  $s = t$  item, we get

$$\begin{aligned} \mathbf{e}_{k+1}(t+1) &= \mathbf{e}_k(t+1) - \mathbf{C}(t+1)\mathbf{b}(t)\mathbf{H}(t)(P \otimes I_N)\mathbf{e}_k(t+1) \\ &\quad - \sum_{s=0}^{t-1} (\mathbf{C}(t+1)\Phi(t+1, s+1)\mathbf{b}(s))\mathbf{H}(s)(P \otimes I_N)\mathbf{e}_k(s+1) \\ &= (\mathbf{I}_{NM} - (P \otimes I_N))\mathbf{e}_k(t+1) - \sum_{s=0}^{t-1} \mathbf{C}(t+1)\Phi(t+1, s+1)\mathbf{b}(s)\mathbf{H}(s)(P \otimes I_N)\mathbf{e}_k(s+1) \end{aligned} \quad (3.21)$$

Taking the norm in both sides of (3.21), we have



$$\|e_{k+1}(t+1)\| \leq \|(I_{NM} - (P \otimes I_N))\| \|e_k(t+1)\| + \sum_{s=0}^{t-1} \|C(t+1)\Phi(t, s+1)\mathbf{b}(s)\mathbf{H}(s)(P \otimes I_N)\| \|e_k(s+1)\|. \quad (3.22)$$

We now write

$$\|(I_{NM} - (P \otimes I_N))\| = \rho, \quad (0 < \rho < 1),$$

$$\sup_{\substack{t \in \{1, 2, \dots, N\} \\ s \in \{0, 1, 2, \dots, t-1\}}} \|C(t+1)\Phi(t, s+1)\mathbf{b}(s)\mathbf{H}(s)(P \otimes I_N)\| = \sigma.$$

For each  $t \in \{1, 2, \dots, N\}$ , multiply by  $\lambda^{t+1}$  on both sides of Eq (3.22), where  $0 < \lambda < 1$ , which yields

$$\begin{aligned} \lambda^{t+1} \|e_{k+1}(t+1)\| &\leq \rho \lambda^{t+1} \|e_k(t+1)\| + \sigma \sum_{s=0}^{t-1} \lambda^{t-s} \lambda^{s+1} \|e_k(s+1)\| \\ &\leq \rho \lambda^{t+1} \|e_k(t+1)\| + \sigma \sum_{s=0}^{t-1} \lambda^{t-s} \sup_{s \in \{1, 2, \dots, N\}} \{\lambda^{s+1} \|e_k(s+1)\|\} \\ &\leq \rho \lambda^{t+1} \|e_k(t+1)\| + \sigma \frac{\lambda(1-\lambda^N)}{1-\lambda} \sup_{t \in \{1, 2, \dots, N\}} \{\lambda^{t+1} \|e_k(t+1)\|\} \end{aligned} \quad (3.23)$$

Therefore, we obtain

$$\sup_{t \in \{1, 2, \dots, N\}} \{\lambda^{t+1} \|e_{k+1}(t+1)\|\} \leq \left( \rho + \sigma \frac{\lambda(1-\lambda^N)}{1-\lambda} \right) \sup_{t \in \{1, 2, \dots, N\}} \{\lambda^{t+1} \|e_k(t+1)\|\}. \quad (3.24)$$

According the definition of the  $\lambda$ -norm, we have

$$\|e_{k+1}(t+1)\|_\lambda \leq \left( \rho + \sigma \frac{\lambda(1-\lambda^N)}{1-\lambda} \right) \|e_k(t+1)\|_\lambda. \quad (3.25)$$

Under the condition  $0 < \rho < 1$  and if we select a sufficiently small  $\lambda$ , this yields

$$0 < \left( \rho + \sigma \frac{\lambda(1-\lambda^N)}{1-\lambda} \right) < 1. \quad (3.26)$$

By (3.25) and (3.26), we have

$$\lim_{k \rightarrow \infty} \|e_{k+1}(t+1)\|_\lambda = 0. \quad (3.27)$$

This means that the agents can accurately track the desired trajectory. Therefore, if  $\|(I_{NM} - (P \otimes I_N))\| = \rho < 1$  is guaranteed, the MAS can complete the consistency tracking task of the desired trajectory under the ILC law (3.14).

#### 4. Distributed data-driven iterative learning control for the system with measurement noise

In Part 3, for MASs with a fixed initial state and no noise, IOCPs can be accurately obtained by designing an iterative learning mechanism, but in the actual environment, the system often contains different types of noise factors. In this section, we consider the situation when the MASs incorporate measurement noise and identify the IOCPs of the system through a suitable iterative learning

mechanism. Then, by integrating it into the ILC law, we aim to achieve consistent tracking of the desired trajectory for the MASs.

We account for the inclusion of measurement noise within system (2.1) with

$$\begin{cases} \mathbf{x}_{j,k}(t+1) = \mathbf{A}_j(t)\mathbf{x}_{j,k}(t) + \mathbf{b}_j(t)u_{j,k}(t), t \in T, \\ \mathbf{y}_{j,k}(t) = \mathbf{c}_j(t)\mathbf{x}_{j,k}(t) + \omega_{j,k}(t), t \in T^+, \end{cases} \quad (4.1)$$

where  $\omega_{j,k}(t)$  is measurement noise that satisfies Assumption 4.

**Assumption 4:** The measurement noise  $\omega_{j,k}(t)$  is independent and identically distributed. The expectation and variance of  $\omega_{j,k}(t)$  are represented as  $E\{\omega_{j,k}(t)\} = 0$  and  $E\{|\omega_{j,k}(t)|^2\} = \chi_j^2$ , respectively, and the noise signal is bounded as  $|\omega_{j,k}(t)| \leq \varpi_j$ .

#### 4.1. *N*-step iterative learning identification for IOCPs

To mitigate the impact of noise on the iterative learning process and bolster its robustness, this part introduces the concept of the maximum allowable control deviation, denoted  $\delta_{j,t}$ , in the design of the iterative learning mechanism. The maximum allowable control deviation  $\delta_{j,t}$  ( $t \in T$ ) is defined as the difference between the maximum and minimum allowable control signals. If  $\delta_{j,t} = 0$ , we are unable to obtain valuable information regarding the controlled system. Therefore, we assume  $\delta_{j,t} > 0$ , ( $t \in T$ ) in this part.

**Theorem 3:** Assume that the MAS (4.1) is subject to the Assumptions 3 and 4. The IOCPs  $\{\mathbf{c}_j(1)\mathbf{b}_j(0), \mathbf{c}_j(2)\mathbf{b}_j(1), \dots, \mathbf{c}_j(N)\mathbf{b}_j(N-1)\}$  can then be captured by designing a proper iterative learning mechanism through running the system (4.1)  $N$  times. Moreover, the expectation and variance of the IOCPs are

$$E\left\{\frac{\mathbf{y}_{j,t}(t+1) - \mathbf{y}_{j,0}(t+1)}{\delta_{j,t}}\right\} = \mathbf{c}_j(t+1)\mathbf{b}_j(t), (t \in T^+),$$

$$E\left\{\left|\frac{\mathbf{y}_{j,t}(t+1) - \mathbf{y}_{j,0}(t+1)}{\delta_{j,t}} - \mathbf{c}_j(t+1)\mathbf{b}_j(t)\right|^2\right\} = \frac{2(\chi_j)^2}{(\delta_{j,t})^2}, (t \in T^+).$$

We define the super-vectors as follows:

$$\boldsymbol{\omega}_{j,k} = [\omega_{j,k}(1), \omega_{j,k}(2), \dots, \omega_{j,k}(N)]^T.$$

Thus, the system (4.1) can be described as

$$\mathbf{y}_{j,k} = \mathbf{G}_j \mathbf{u}_{j,k} + \boldsymbol{\omega}_{j,k}. \quad (4.2)$$

The detailed process is summarized as follows:

Step 0: The control input signals of each agent are arbitrary given as

$$\mathbf{u}_{j,0} = \{u_{j,0}(0), u_{j,0}(1), \dots, u_{j,0}(N-1)\}, \quad (4.3)$$

Then, the outputs of the MAS (4.1)  $\mathbf{y}_{j,0} = \{y_{j,0}(1), y_{j,0}(2), \dots, y_{j,0}(N)\}$  are generated by

$$\mathbf{y}_{j,0} = \mathbf{G}_j \mathbf{u}_{j,0} + \boldsymbol{\omega}_{j,0}. \quad (4.4)$$

Step 1: Select the control input signals  $\mathbf{u}_{j,1} = \{u_{j,1}(0) + \delta_{j,1}, u_{j,1}(1), \dots, u_{j,1}(N-1)\}$  that satisfy

$$\mathbf{u}_{j,1} = \{u_{j,0}(0) + \delta_{j,1}, u_{j,0}(1), u_{j,0}(2), \dots, u_{j,0}(N-1)\}, \quad (4.5)$$

where  $\delta_{j,1}$  is the maximum allowable control deviation.

Then, the outputs of the MAS (4.1)  $\mathbf{y}_{j,1} = \{y_{j,1}(1), y_{j,1}(2), \dots, y_{j,1}(N)\}$  are generated by

$$\mathbf{y}_{j,1} = \mathbf{G}_j \mathbf{u}_{j,1} + \boldsymbol{\omega}_{j,1}. \quad (4.6)$$

Subtracting (4.4) from (4.6) yields

$$\begin{aligned} \mathbf{y}_{j,1} - \mathbf{y}_{j,0} &= \mathbf{G}_j (\mathbf{u}_{j,1} - \mathbf{u}_{j,0}) - (\boldsymbol{\omega}_{j,1} - \boldsymbol{\omega}_{j,0}), \\ \Rightarrow \frac{y_{j,1}(1) - y_{j,0}(1)}{\delta_{j,1}} &= \mathbf{c}_j(1) \mathbf{b}_j(0) + \frac{\omega_{j,1}(1) - \omega_{j,0}(1)}{\delta_{j,1}}. \end{aligned} \quad (4.7)$$

According to Assumption 4 of measurement noise, the mean and variance are calculated as

$$\begin{cases} E \left\{ \frac{y_{j,1}(1) - y_{j,0}(1)}{\delta_{j,1}} \right\} = \mathbf{c}_j(1) \mathbf{b}_j(0), \\ E \left\{ \left| \frac{y_{j,1}(1) - y_{j,0}(1)}{\delta_{j,1}} - \mathbf{c}_j(1) \mathbf{b}_j(0) \right|^2 \right\} = \frac{2(\chi_j)^2}{(\delta_{j,1})^2}. \end{cases} \quad (4.8)$$

Step 2: Select the control input signals  $\mathbf{u}_{j,2} = \{u_{j,2}(0), u_{j,2}(1), \dots, u_{j,2}(N-1)\}$  that satisfy

$$\mathbf{u}_{j,2} = \{u_{j,0}(0), u_{j,0}(1) + \delta_{j,2}, u_{j,0}(2), \dots, u_{j,0}(N-1)\}, \quad (4.9)$$

where  $\delta_{j,2}$  is the maximum allowable control deviation.

Then, the outputs of the MASs (4.1)  $\mathbf{y}_{j,2} = \{y_{j,2}(1), y_{j,2}(2), \dots, y_{j,2}(N)\}$  are generated by

$$\mathbf{y}_{j,2} = \mathbf{G}_j \mathbf{u}_{j,2} + \boldsymbol{\omega}_{j,2}. \quad (4.10)$$

Subtracting (4.4) from (4.10) yields

$$\begin{aligned} \mathbf{y}_{j,2} - \mathbf{y}_{j,0} &= \mathbf{G}_j (\mathbf{u}_{j,2} - \mathbf{u}_{j,0}) - (\boldsymbol{\omega}_{j,2} - \boldsymbol{\omega}_{j,0}), \\ \Rightarrow \frac{y_{j,2}(2) - y_{j,0}(2)}{\delta_{j,2}} &= \mathbf{c}_j(2) \mathbf{b}_j(1) + \frac{\omega_{j,2}(2) - \omega_{j,0}(2)}{\delta_{j,2}}. \end{aligned} \quad (4.11)$$

By the measurement noise limitation in Assumption 4, the mean and variance of the noise are

calculated as follows:

$$\begin{cases} E \left\{ \frac{y_{j,2}(2) - y_{j,0}(2)}{\delta_{j,2}} \right\} = \mathbf{c}_j(2) \mathbf{b}_j(1), \\ E \left\{ \left| \frac{y_{j,2}(s) - y_{j,0}(s)}{\delta_{j,2}} - \mathbf{c}_j(2) \mathbf{b}_j(1) \right|^2 \right\} = \frac{2(\chi_j)^2}{(\delta_{j,2})^2}. \end{cases} \quad (4.12)$$

Step  $N$ : By using the method above, analogous to the case when  $t = N$ , select the control input signals  $\mathbf{u}_{j,N} = \{u_{j,N}(0), u_{j,N}(1), \dots, u_{j,N}(N-1)\}$  that satisfy

$$\mathbf{u}_{j,N} = \{u_{j,N}(0), u_{j,N}(1), u_{j,0}(2), \dots, u_{j,0}(N-1) + \delta_{j,N}\},$$

Then, the outputs of the MAS (4.1)  $\mathbf{y}_{j,N} = \{y_{j,N}(1), y_{j,N}(2), \dots, y_{j,N}(N)\}$  are generated by

$$\mathbf{y}_{j,N} = \mathbf{G}_j \mathbf{u}_{j,N} + \boldsymbol{\omega}_{j,N}. \quad (4.13)$$

Subtracting (4.4) from (4.13) yields

$$\begin{aligned} \mathbf{y}_{j,N} - \mathbf{y}_{j,0} &= \mathbf{G}_j (\mathbf{u}_{j,N} - \mathbf{u}_{j,0}) - (\boldsymbol{\omega}_{j,N} - \boldsymbol{\omega}_{j,0}), \\ \Rightarrow \frac{y_{j,N}(N) - y_{j,0}(N)}{\delta_{j,N}} &= \mathbf{c}_j(N) \mathbf{b}_j(N-1) + \frac{\omega_{j,N}(N) - \omega_{j,0}(N)}{\delta_{j,N}}. \end{aligned} \quad (4.14)$$

By the measurement noise limitation in Assumption 4, the mean and variance are calculated as

$$\begin{cases} E \left\{ \frac{y_{j,N}(N) - y_{j,0}(N)}{\delta_{j,N}} \right\} = \mathbf{c}_j(N) \mathbf{b}_j(N-1), \\ E \left\{ \left| \frac{y_{j,N}(N) - y_{j,0}(N)}{\delta_{j,N}} - \mathbf{c}_j(N) \mathbf{b}_j(N-1) \right|^2 \right\} = \frac{2(\chi_j)^2}{(\delta_{j,N})^2}. \end{cases} \quad (4.15)$$

According to the steps above, when the initial states of each agent in the system are fixed, we can obtain the IOCPs  $\{\mathbf{c}_j(1) \mathbf{b}_j(0), \mathbf{c}_j(2) \mathbf{b}_j(1), \dots, \mathbf{c}_j(N) \mathbf{b}_j(N-1)\}$  of each agent accurately in the MAS by applying the identification mechanism.

**Remark 4.** Owing to the randomness of the noise signal, measurement of the output is challenging. Hence, we obtain IOCPs by repeating the experiment to obtain the mean value. Suppose that we repeat the experiment  $Q$  times. We can calculate the mean as the approximate value  $\mathbf{c}_j(t+1) \mathbf{b}_j(t)$ , that is,

$$\mathbf{c}_j(t+1) \mathbf{b}_j(t) = \frac{1}{Q} \sum_{q=1}^Q \frac{y_{j,t}^q(t+1) - y_{j,0}^q(t+1)}{\delta_{j,t}},$$

where the superscript  $q$  represents the number of experiments. The accuracy of IOCPs is improved through multiple experiments.

#### 4.2. Distributed data-driven iterative learning control

The ILC law (3.14), which is based on the estimated IOCPs, is utilized to regulate the MAS (4.1) with measurement noise and ensure consistent tracking of the desired trajectory. For simplicity, write it as

$$z_j(t) = E \left\{ \frac{y_{j,t(t+1)} - y_{j,0(t+1)}}{\delta_{j,t}} \right\}.$$

According to Theorem 3, we can derive

$$u_{j,k+1}(t) = u_{j,k}(t) + \frac{1}{z_j(t)} \xi_{j,k}(t+1), (t \in T^+). \quad (4.16)$$

**Theorem 4:** Assume that the iterative learning control strategy (4.16) is utilized for the MAS (4.1) under Assumptions 1–4. The tracking error is bounded if the communication topology matrix of the MAS (4.1) satisfies  $\|(I_{NM} - (P \otimes I_N))\| = \rho < 1$ .

**Proof:** We calculate the solution of the system (4.1) as follows

$$y_{j,k}(t+1) = \mathbf{c}_j(t+1)\Phi_j(t+1,0)\mathbf{x}_{j,k}(0) + \mathbf{c}_j(t+1)\sum_{s=0}^t \Phi_j(t+1,s+1)\mathbf{b}_j(s)u_{j,k}(s) + \omega_{j,k}(t). \quad (4.17)$$

Utilizing the result of the calculation (3.17), we can represent the solution of the system (4.1) as

$$\mathbf{y}_k(t+1) = \mathbf{c}(t+1)\mathbf{x}_k(t+1) + \boldsymbol{\omega}_k(t+1), \quad (4.18)$$

where  $\boldsymbol{\omega}_k(t+1) = [\omega_{1,k}^T(t+1), \omega_{2,k}^T(t+1), \dots, \omega_{M,k}^T(t+1)]^T$ .

By computing the tracking error, we have

$$\begin{aligned} \mathbf{e}_{k+1}(t+1) &= (I_{NM} - \mathbf{c}(t+1)\mathbf{b}(t)\mathbf{Z}(t)(P \otimes I_N))\mathbf{e}_k(t+1) \\ &\quad - \sum_{s=0}^{t-1} \mathbf{c}(t+1)\boldsymbol{\Phi}(t+1,s+1)\mathbf{b}(s)\mathbf{Z}(s)(P \otimes I_N)\mathbf{e}_k(s+1) - (\boldsymbol{\omega}_{k+1}(t+1) - \boldsymbol{\omega}_k(t+1)), \end{aligned} \quad (4.19)$$

where  $\mathbf{Z}(t) = \text{diag} \left( \frac{1}{z_1(t)}, \frac{1}{z_2(t)}, \dots, \frac{1}{z_M(t)} \right)$ .

The demonstration adheres to the pattern of Theorem 2. Consequently, for the sake of brevity, the comprehensive steps have been excluded.

Write  $\Delta\boldsymbol{\omega}_{k+1}(t+1) = (\boldsymbol{\omega}_{k+1}(t+1) - \boldsymbol{\omega}_k(t+1))$ . On the basis of the noise from Assumption 4, we know  $\|\Delta\omega_{j,k+1}(t+1)\| \leq 2N\varpi_j$ , where  $\Delta\boldsymbol{\omega}_{k+1}(t+1)$  satisfies

$$\|\Delta\boldsymbol{\omega}_{k+1}(t+1)\| \leq 2NM\varpi = \tilde{\omega}. \quad (4.20)$$

Taking the norm in the both sides of (4.19) gives

$$\begin{aligned} \|\mathbf{e}_{k+1}(t+1)\| &\leq \|I_{NM} - \mathbf{c}(t+1)\mathbf{b}(t)\mathbf{Z}(t)(P \otimes I_N)\| \|\mathbf{e}_k(t+1)\| \\ &\quad + \sum_{s=0}^{t-1} \|\mathbf{c}(t+1)\boldsymbol{\Phi}(t+1,s+1)\mathbf{b}(s)\mathbf{Z}(s)(P \otimes I_N)\| \|\mathbf{e}_k(s+1)\| + \tilde{\omega} \end{aligned} \quad (4.21)$$

For each  $t \in \{1, 2, \dots, N\}$ , by multiplying by  $\lambda^{t+1}$  on both sides of the inequation (4.21), with  $0 < \lambda < 1$ , while applying the similar derivation process from (3.23) to (3.24), we get

$$\sup_{t \in \{1, 2, \dots, N\}} \{\lambda^{t+1} \|\mathbf{e}_{k+1}(t+1)\|\} \leq \left( \rho + \sigma \left( \frac{\lambda(1-\lambda^N)}{1-\lambda} \right) \right) \sup_{t \in \{1, 2, \dots, N\}} \{\lambda^{t+1} \|\mathbf{e}_k(t+1)\|\} + \tilde{\omega}. \quad (4.22)$$

Let  $\rho_1 = \left( \rho + \sigma \left( \frac{\lambda(1-\lambda^N)}{1-\lambda} \right) \right)$ . Since the condition is  $0 < \rho < 1$ , by choosing a sufficiently small  $\lambda$ ,

we have  $0 < \rho_1 < 1$ .

Thus, by (4.22), we have

$$\|e_{k+1}(t+1)\|_\lambda \leq \rho_1 \|e_k(t+1)\|_\lambda + \tilde{\omega}, \quad (4.23)$$

which implies that

$$\lim_{k \rightarrow \infty} \|e_{k+1}(t+1)\|_\lambda \leq \frac{\tilde{\omega}}{1-\rho_1}. \quad (4.24)$$

Consequently, the tracking error is confined within a certain limit, thereby concluding the proof.

## 5. Simulation

To assess the efficacy of the introduced algorithm, a linear time-varying discrete MAS composed of one virtual leader and four agents is considered.

*Example 1. The multi-agent system without measurement noise*

Replace the desired trajectory with a virtual leader  $y_d(t) = 1.1 * \sin((\pi t)/20) + 0.25 \cos((\pi t)/20)$ . The parametric equations of each agent are expressed as

$$\begin{cases} \mathbf{x}_{1,k}(t+1) = \begin{pmatrix} 1 & 0.009 & 0 \\ 0 & 1 & 0.009 \\ -0.0540 & -0.2711 & 0.237 - 0.27e^{-2t} \end{pmatrix} \mathbf{x}_{1,k}(t) + \begin{pmatrix} 0 \\ 0 \\ 0.009 \end{pmatrix} u_{1,k}(t), t \in T \\ y_{1,k}(t) = (6 \ 30 \ 84 + 30e^{-2t}) \mathbf{x}_{1,k}(t), t \in T^+ \end{cases} \quad (5.1)$$

$$\begin{cases} \mathbf{x}_{2,k}(t+1) = \begin{pmatrix} 1 & 0.009 & 0 \\ 0 & 1 & 0.009 \\ -0.0540 & -0.2711 & 0.237 - 0.27e^{-2t} \end{pmatrix} \mathbf{x}_{2,k}(t) + \begin{pmatrix} 0 \\ 0 \\ 0.009 \end{pmatrix} u_{2,k}(t), t \in T \\ y_{2,k}(t) = (6 \ 30 \ 55 + 30e^{-2t}) \mathbf{x}_{2,k}(t), t \in T^+ \end{cases} \quad (5.2)$$

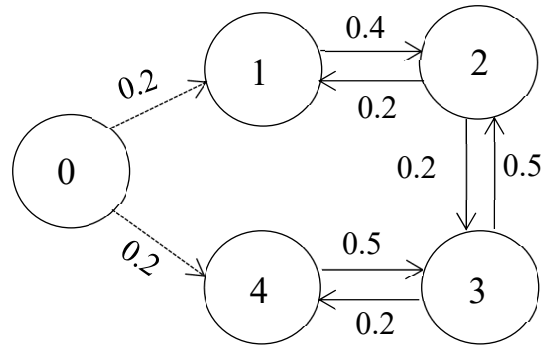
$$\begin{cases} \mathbf{x}_{3,k}(t+1) = \begin{pmatrix} 1 & 0.009 & 0 \\ 0 & 1 & 0.009 \\ -0.0540 & -0.2711 & 0.237 - 0.27e^{-2t} \end{pmatrix} \mathbf{x}_{3,k}(t) + \begin{pmatrix} 0 \\ 0 \\ 0.009 \end{pmatrix} u_{3,k}(t), t \in T \\ y_{3,k}(t) = (6 \ 30 \ 75 + 20e^{-2t}) \mathbf{x}_{3,k}(t), t \in T^+ \end{cases} \quad (5.3)$$

$$\begin{cases} \mathbf{x}_{4,k}(t+1) = \begin{pmatrix} 1 & 0.009 & 0 \\ 0 & 1 & 0.009 \\ -0.0540 & -0.2711 & 0.237 - 0.27e^{-2t} \end{pmatrix} \mathbf{x}_{4,k}(t) + \begin{pmatrix} 0 \\ 0 \\ 0.009 \end{pmatrix} u_{4,k}(t), t \in T \\ y_{4,k}(t) = (6 \ 25 \ 60 + 30e^{-2t}) \mathbf{x}_{4,k}(t), t \in T^+ \end{cases} \quad (5.4)$$

where  $T \in [0,79]$ ,  $T^+ \in [1,80]$ .

As shown in Figure 1, 0 represents the virtual leader and represents the desired trajectory of the system; the remaining four are Agents 1 to 4. According to Figure 1, it is notable that Agents 1 and 4 have direct access to the desired trajectory information from the virtual leader 0. Agents 2 and 3 can indirectly obtain information by communicating with Agents 1 and 4. The degree matrix  $S$  and the Laplacian matrix  $\mathcal{L}$  are calculated as follows:

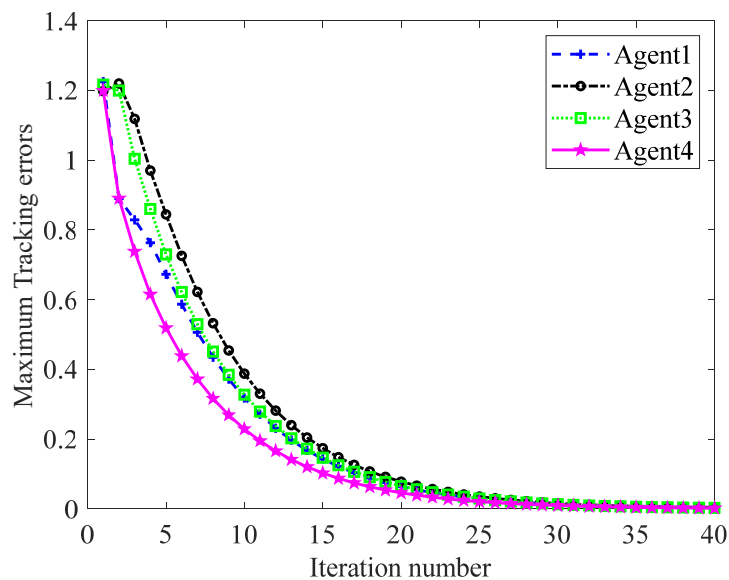
$$S = \text{diag}(0.2 \ 0 \ 0 \ 0.2), \mathcal{L} = \begin{pmatrix} 0.4 & -0.4 & 0 & 0 \\ -0.2 & 0.7 & -0.5 & 0 \\ 0 & -0.2 & 0.7 & -0.5 \\ 0 & 0 & -0.2 & 0.2 \end{pmatrix}.$$



**Figure 1.** Communication topology between agents.

Thus, the convergence condition is satisfied by  $\|(I_{NM} - (P \otimes I_N))\| = \rho = 0.9428 < 1$ .

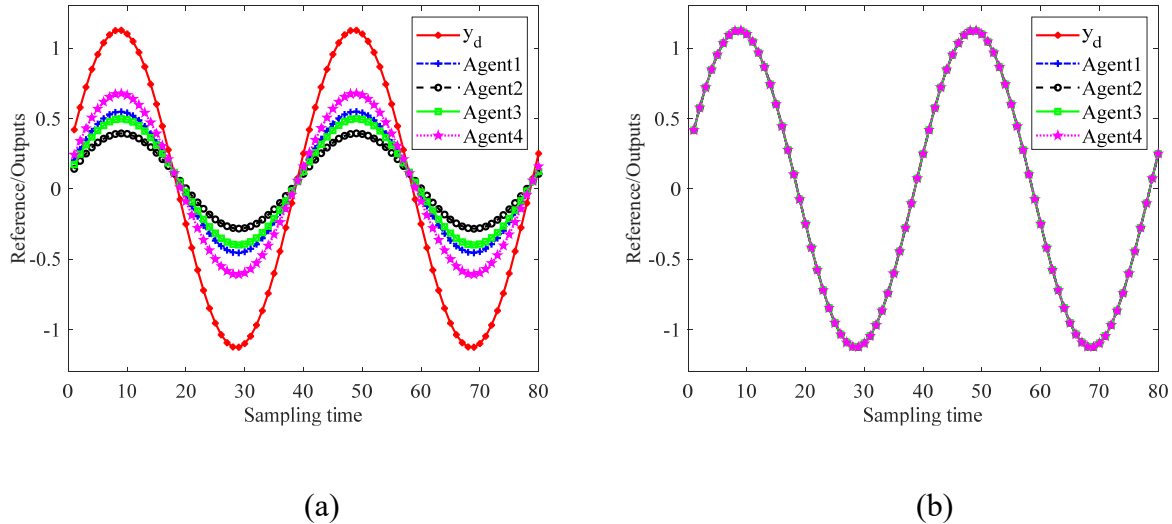
Suppose that the IOCP identification mechanism in Section 3.1 and the ILC law (3.14) are applied to the system (3.1). The initial input states of each agent are set as  $\mathbf{x}_{1,k}(0) = \mathbf{x}_{2,k}(0) = \mathbf{x}_{3,k}(0) = \mathbf{x}_{4,k}(0) = \mathbf{0}$ . The initial input control signals of each agent are chosen as  $u_{1,0}(t) = u_{2,0}(t) = u_{3,0}(t) = u_{4,0}(t) = 0.1$ . The results are depicted in Figures 2 and 3.



**Figure 2.** Tracking errors of each agent.

As shown in Figure 2, the tracking error of each agent is demonstrated. It can be seen that under the control of the ILC law (3.14), the tracking error of each agent decreases gradually and converges to zero as the number of iteration increases.

Figure 3(a),(b) displays the outputs of the MAS at the 5th and 40th iterations. As the number of iterations rises, the ILC law guides the output of all followers closer to the desired trajectory. After 40 iterations, all agents have achieved successful tracking of the desired trajectory. Thereby, the effectiveness of the algorithm is validated.



**Figure 3.** Outputs of agents at (a) the 5th iteration and (b) the 40th iteration.

*Example 2. The multi-agent systems (5.1)–(5.4) with measurement noise*

In this part, we verify the effectiveness of the proposed algorithm (4.16) in the MASs (5.1)–(5.4) with measurement noise.

On the basis of the MASs (5.1)–(5.4), the initial conditions and topology are set as in Example 1. Assume the measurement noise is subject to Gaussian distribution, with the expectation being 0 and the variance being  $0.01^2$ , i.e.,  $\omega_{j,k}(t) \sim N(0, 0.01^2)$ .

We take Agent 1 for observation to illustrate the validity of Theorem 3. In addition, in order to illustrate the influence of different maximum allowable control deviations  $\delta_{1,t}$  on the approximation errors of IOCPs, we consider three situations:  $\delta_{1,t} = 5, 10, 20$ .

When the measurement noise is involved, we use the mean of 100 group experiments to get the approximate IOCPs with different maximum allowable control deviations. Figure 4 shows the errors of the expected observed values compared with the real values.

$$\frac{1}{100} \sum_{q=1}^{100} \frac{y_{j,t}^q(t+1) - y_{j,0}^q(t+1)}{\delta_{j,t}} - \mathbf{c}_j(t+1)\mathbf{b}_j(t). \quad (5.5)$$

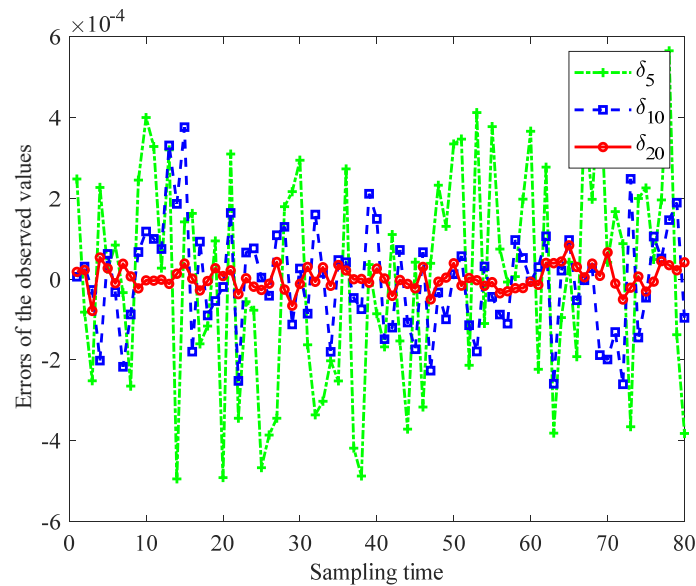
Figure 5 displays the variances of the observed values in the presence of measurement noise with different maximum allowable control deviations to evaluate the discreteness of observed values.

$$\frac{1}{100} \sum_{q=1}^{100} \left[ \frac{y_{j,t}^q(t+1) - y_{j,0}^q(t+1)}{\delta_{j,t}} - \mathbf{c}_j(t+1)\mathbf{b}_j(t) \right]^2. \quad (5.6)$$

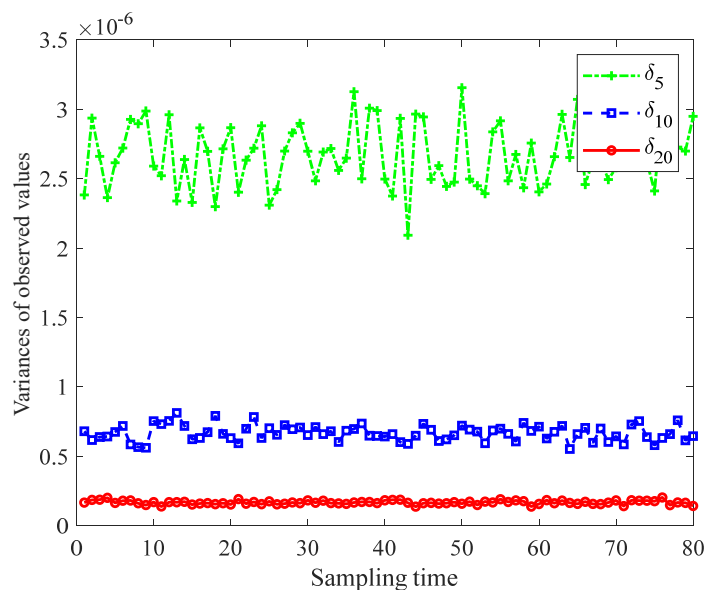
The results presented in both Figures 4 and 5 clearly reveal that the approximation error of the



IOCPs exhibits an inverse relationship with the permissible maximum control error. In other words, as the maximum allowable control deviation increases, the approximate error of IOCPs tends to decrease.

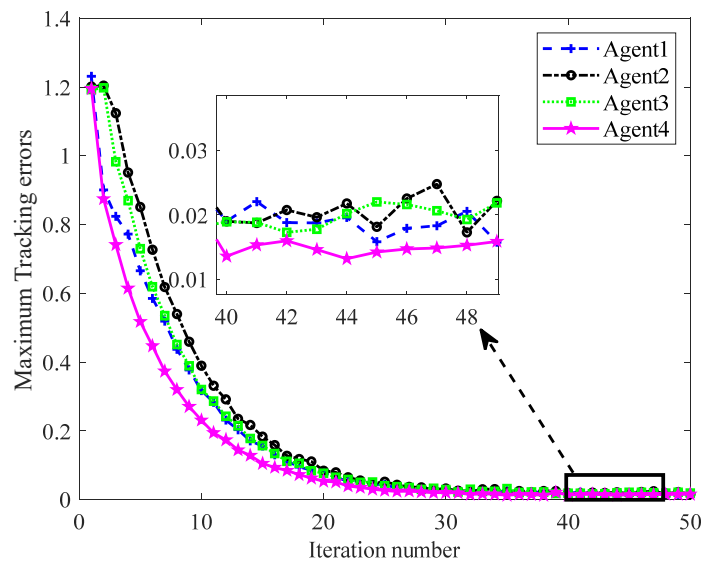


**Figure 4.** Errors of the observed values with different values of  $\delta_{1,t}$ .



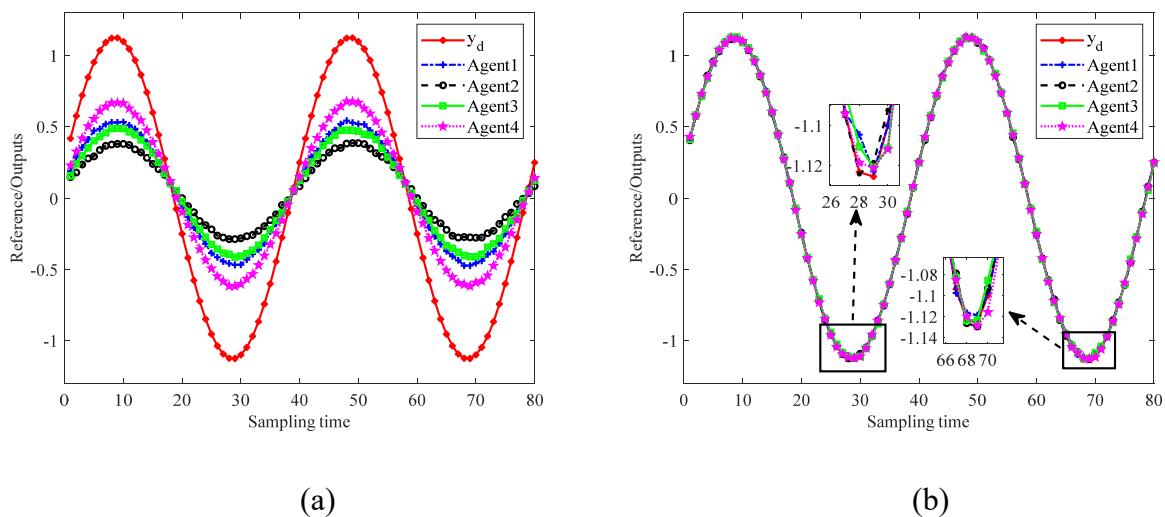
**Figure 5.** Variance of the observed values with different values of  $\delta_{1,t}$ .

In Figure 6, the tracking error of each agent is precisely displayed. Under the control of the ILC law (4.16), it can be clearly observed that the tracking error of each agent gradually converges to a fixed boundary as the number of iterations increases. This shows the effectiveness and reliability of the proposed algorithm in this paper.



**Figure 6.** Tracking error of MASs with measurement noise.

In Figure 7(a),(b), the outputs of the MASs are demonstrated at the 5th and 40th iterations, respectively. It can be observed that the output of all followers tends towards the desired trajectory. This result of the simulation experiment is consistent with the boundedness of Theorem 4; it also illustrates that the proposed ILC (4.16) is effective.



**Figure 7.** Outputs of the agents at (a) the 5th iteration and (b) the 40th iteration.

### Example 3. Engineering practice

To verify the feasibility of the distributed data-driven ILC algorithm, we take the Parrot AR unmanned aerial vehicle (UAV) [25] with periodic operation features as the simulation object. To conduct the yaw model identification, a step response experiment was designed and the relevant flight data were recorded. The mathematical model in transfer function form is as follows:

$$\frac{\psi(s)}{U_\psi(s)} = \frac{k\omega_n^2}{s^3 + 2\zeta\omega_n s^2 + \omega_n^2 s} \quad (5.7)$$

The speed of the yaw angle  $\dot{\psi}$  of the UAV is related to the control input. The dynamic model for yaw orientation represented in state space is given as follows:

$$\mathbf{x}(t) = \begin{bmatrix} \psi \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix}, U_\psi(t) = u(t), \dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \\ \dddot{\psi} \end{bmatrix}, y(t) = [1 \ 0 \ 0]\mathbf{x}(t) \quad (5.8)$$

If we consider the MASs consisting of four UAVs, the state space model of each UAV is obtained as follows:

$$\begin{cases} \dot{\mathbf{x}}_{j,k}(t) = \begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \\ \dddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \mathbf{x}_{j,k}(t) + \begin{bmatrix} 0 \\ 0 \\ k\omega_n^2 \end{bmatrix} U_{j,k}(t), j = \{1,2,3,4\} \\ y_{j,k}(t) = [1 \ 0 \ 0]\mathbf{x}_{j,k}(t) \end{cases} \quad (5.9)$$

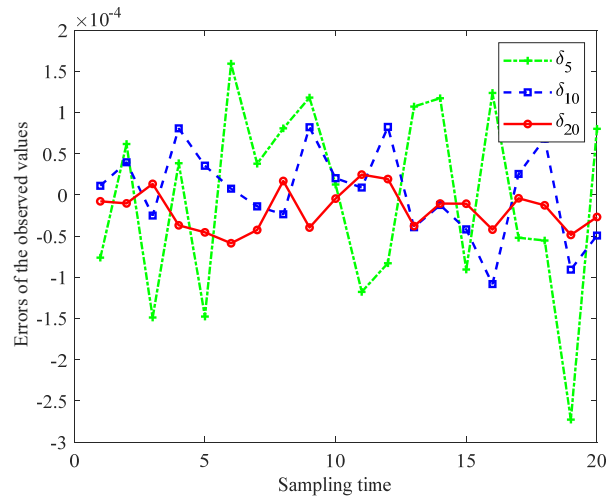
Set  $k = 106.3128$ ,  $\omega_n = 15.0994$ ,  $\zeta = 1.1002$ . Using the zero-order holder for discretization and setting the sampling period and sampling step as  $T = 1s$ ,  $T_s = 0.05s$ , we get the following discrete system:

$$\begin{cases} \mathbf{x}_{j,k}(t+1) = \begin{bmatrix} 1.0000 & 0.0468 & 0.0007 \\ 0 & 0.8314 & 0.0222 \\ 0 & -5.0676 & 0.0929 \end{bmatrix} \mathbf{x}_{j,k}(t) + \begin{bmatrix} 0.3409 \\ 17.9218 \\ 538.753 \end{bmatrix} U_{j,k}(t), j = \{1,2,3,4\}. \\ y_{j,k}(t) = [1 \ 0 \ 0]\mathbf{x}_{j,k}(t) + \omega_{j,k}(t) \end{cases} \quad (5.10)$$

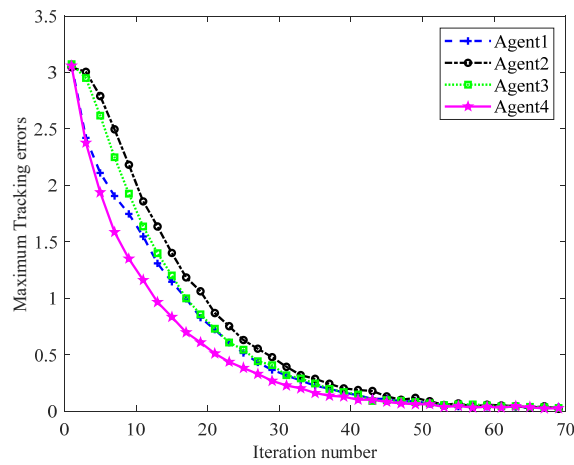
Assume that the four agents have the same structure. The communication topology and initial states are identical to Example 1. It is easy to calculate  $\rho = 0.9428 < 1$ . In addition, we set the system noise as  $\omega_{j,k}(t) \sim N(0, 0.02^2)$ . The initial input control signal of each agent is chosen as 0.03. Let  $y_d(t) = 1.1 * \sin((\pi t)/10) + 0.25 \cos((\pi t)/10)$ . The results are depicted in Figures 8–10.

We consider three situations,  $\delta_t = 5, 10, 20$ . Figure 8 shows the errors of the expected observed values compared with the real values. Figure 9 shows the convergence of the tracking errors of the UAV systems under the ILC law (4.16). Figure 10 displays the output of each agent.

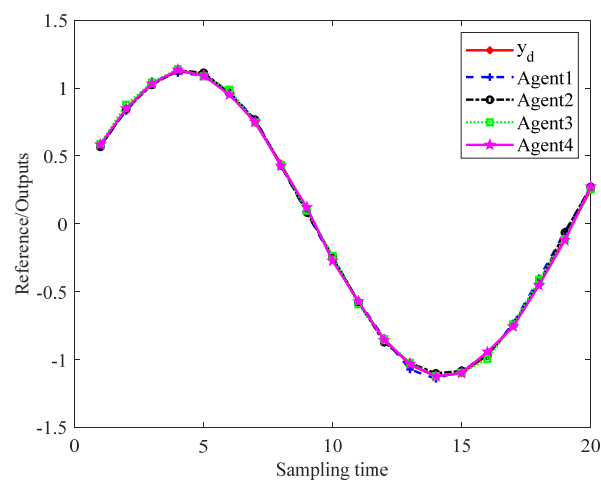
As the number of iterations increases, the tracking error of the agents gradually tends to a fixed boundary. After 60 iterations, the output of the agents approaches the desired trajectory. Therefore, the proposed control law (4.16) can effectively control the UAV to complete the task of tracking the desired trajectory. In general, it is suitable for practical engineering practice.



**Figure 8.** Errors of the observed values with different values of  $\delta_t$ .



**Figure 9.** Tracking errors of each agent.



**Figure 10.** Output of the agents at the 60th iteration.

## 6. Conclusions

This paper investigated the distributed data-driven ILC strategy for linear time-varying MASs with unknown parameters. First, a  $N$ -step iterative learning mechanism was designed to identify the unknown IOCPs by using the repetitive control system and input and output data. Then the reciprocal of the identified IOCPs was selected as the learning gain to construct the ILC law. Second, we considered the presence of measurement noise in MASs, and the maximum allowable control deviation was introduced to minimize the adverse effect of measurement noise on identification of the IOCPs. Rigorous theoretical analysis was used to verify the effectiveness of the proposed ILC protocol. The effectiveness of the proposed method was illustrated by three examples. This investigation provides a feasible scheme for the MASs to track a trajectory.

### Use of AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

### Acknowledgments

This research was supported in part by the Science and Technology Program Project of Xi'an City (22GXFW0037)

### Conflict of interest

The authors declare there is no conflict of interest.

### References

1. S. Arimoto, S. Kawamura, F. Miyazaki, Bettering operation of robots by learning, *J. Rob. Syst.*, **1** (1984), 123–140. <https://doi.org/10.1002/rob.4620010203>
2. J. Huang, W. Wang, X. Su, Adaptive iterative learning control of multiple autonomous vehicles with a time-varying reference under actuator faults, *IEEE Trans. Neural Networks Learn. Syst.*, **32** (2021), 5512–5525. <https://doi.org/10.1109/TNNLS.2021.3069209>
3. S. Saab, D. Shen, M. Orabi, D. Kors, R. Jaafar, Iterative learning control: Practical implementation and automation, *IEEE Trans. Ind. Electron.*, **69** (2022), 1858–1866. <https://doi.org/10.1109/TIE.2021.3063866>
4. L. Zhang, W. Chen, J. Liu, C. Wen, A robust adaptive iterative learning control for trajectory tracking of permanent-magnet spherical actuator, *IEEE Trans. Ind. Electron.*, **63** (2015), 291–301. <https://doi.org/10.1109/TIE.2015.2464186>
5. D. Yang, K. Lee, H. Ahn, J. H. Lee, Experimental application of a quadratic optimal iterative learning control method for control of wafer temperature uniformity in rapid thermal processing, *IEEE Trans. Semicond.*, **16** (2003), 36–44. <https://doi.org/10.1109/TSM.2002.807740>
6. M. Zhou, J. Wang, D. Shen, Iterative learning control for continuous-time multi-agent differential inclusion systems with full learnability, *Chaos, Solitons Fractals*, **174** (2023), 113895. <https://doi.org/10.1016/j.chaos.2023.113895>

7. M. Naserian, A. Ramazani, A. Khaki-Sedigh, A. Moarefianpour, Fast terminal sliding mode control for a nonlinear multi-agent robot system with disturbance, *Syst. Sci. Control Eng.*, **8** (2020), 328–338. <https://doi.org/10.1080/21642583.2020.1764408>
8. N. Pavlov, A. Petrochenkov, A. Romodin, A multi-agent approach for modeling power-supply systems with MicroGrid, *Russ. Electr. Eng.*, **92** (2021), 637–643. <https://doi.org/10.3103/S1068371221110110>
9. Y. Zhang, Y. Zhou, H. Lu, H. Fujita, Cooperative multi-agent actor–critic control of traffic network flow based on edge computing, *Future Gener. Comput. Syst.*, **123** (2021), 128–141. <https://doi.org/10.1016/j.future.2021.04.018>
10. Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, L. Carlone, Kimera-multi: Robust, distributed, densemetric-semantic SLAM for multi-robot systems, *IEEE Trans. Rob.*, **38** (2022), 2022–2038. <https://doi.org/10.1109/TRO.2021.3137751>
11. P. Wang, M. Govindarasu, Multi-agent based attack resilient system integrity protection for smart grid, *IEEE Trans. Smart Grid*, **11** (2020), 3447–3456. <https://doi.org/10.1109/TSG.2020.2970755>
12. H. Ahn, Y. Chen, Iterative learning control for multi-agent formation, in *2009 Institute of Chemistry Chinese Academy of Sciences-Society of Instrument and Control Engineers*, IEEE, (2009), 3111–3116.
13. D. Meng, Y. Jia, Formation control for multi-agent systems through an iterative learning design approach, *Int. J. Robust Nonlinear Control*, **24** (2014), 340–361. <https://doi.org/10.1002/rnc.2890>
14. X. Dai, C. Wang, S. Tian, Q. Huang, Consensus control via iterative learning for distributed parameter models multi-agent systems with time-delay, *J. Franklin Inst.*, **356** (2019), 5240–5259. <https://doi.org/10.1016/j.jfranklin.2019.05.015>
15. Y. Chen, J. Sun, Distributed optimal control for multi-agent systems with obstacle avoidance, *Neurocomputing*, **173** (2016), 2014–2021. <https://doi.org/10.1016/j.neucom.2015.08.085>
16. Z. Hou, S. Jin, A novel data-driven control approach for a class of discrete-time nonlinear systems, *IEEE Trans. Control Syst. Technol.*, **19** (2010), 1549–1558. <https://doi.org/10.1109/TCST.2010.2093136>
17. X. Bu, Q. Yu, Z. Hou, W. Qian, Model free adaptive iterative learning consensus tracking control for a class of nonlinear multi-agent systems, *IEEE Trans. Syst. Man Cybern.: Syst.*, **49** (2017), 677–686. <https://doi.org/10.1109/TSMC.2017.2734799>
18. C. Liu, X. Ruan, Input-output-driven gain-adaptive iterative learning control for linear discrete-time-invariant systems, *Int. J. Robust Nonlinear Control*, **31** (2021), 8551–8568. <https://doi.org/10.1002/rnc.5753>
19. Y. Geng, S. Wang, X. Ruan, The convergence of data-driven optimal iterative learning control for linear multi-phase batch processes, *Mathematics*, **10** (2022), 2304. <https://doi.org/10.3390/math10132304>
20. N. Lin, R. Chi, B. Huang, Event-triggered ILC for optimal consensus at specified data points of heterogeneous networked agents with switching topologies, *IEEE Trans. Cybern.*, **52** (2021), 8951–8961. <https://doi.org/10.1109/TCYB.2021.3054421>
21. Y. Zhang, J. Liu, X. Ruan, Iterative learning control for uncertain nonlinear networked control systems with random packet dropout, *Int. J. Robust Nonlinear Control*, **29** (2019), 3529–3546. <https://doi.org/10.1002/rnc.4568>

22. J. Liu, W. Chen, X. Ruan, Y. Zheng, Data-based iterative learning mechanism for unknown input-output coupling parameters/matrices, *Int. J. Robust Nonlinear Control*, **30** (2020), 1275–1297. <https://doi.org/10.1002/rnc.4827>
23. A. Hock, A. Schoellig, Distributed iterative learning control for multi-agent systems: Theoretic developments and application to formation flying, *Auton. Robot.*, **43** (2019), 1989–2010. <https://doi.org/10.1007/s10514-019-09845-4>
24. P. Pakshin, A. Kuposov, J. Emelianova, Iterative learning control of a multiagent system under random perturbations, *Autom. Remote Control*, **81** (2020), 483–502. <https://doi.org/10.1134/S0005117920030078>
25. G. Ortega, F. Muñoz, E. Quesada, L. R. Garcia, P. Ordaz, Implementation of leader-follower linear consensus algorithm for coordination of multiple aircrafts, in *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, IEEE, (2015), 25–32. <https://doi.org/10.1109/RED-UAS.2015.7440987>



AIMS Press

©2025 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)