Computing and Informatics, Vol. 45, 2026, 648–667, doi: 10.31577/cai_2026_3_648

DISTRIBUTED MULTI-AGENT D-TYPE ITERATIVE LEARNING ALGORITHM RESEARCH UNDER COMMUNICATION INTERFERENCE

Xiaoyan Cheng, Wei Tang*, Hui He, Hao Wang

Army Arms University of PLA
Beijing 100072, China
e-mail: 94180102@qq.com

Abstract. Iterative learning is based on the continuous iterative updating of model parameters, which does not need an accurate system model, and can realize the control of dynamic systems with high uncertainty characteristics in a limited time. It is widely used in many fields, such as robots, UAVs, transportation, networks, finance and medical treatment. The distributed multi-agent based on iterative learning distributes large-scale data on multiple nodes for parallel processing, which speeds up the learning speed and improves the robustness of the algorithm. However, the multi-agent system depends on the accurate information transmission between individuals to complete the target task. Due to the complexity of the environment and uncertain factors, such as data loss, delay, noise and other interference, it must be considered. In this paper, an iterative learning control protocol based on local data and a D-type iterative learning algorithm are designed for nonlinear time-varying multi-agent systems with communication interference under the premise that the initial state can be reset to zero. Simulation results show that the proposed algorithm can effectively solve the consistency problem and suppress any type of disturbance, which has strong theoretical and practical significance.

Keywords: Iterative learning control, distributed multi-agent, communication interference, consistency problem

* Corresponding author

1 INTRODUCTION

Iterative learning control (ILC) is a control system design method. It is a machine learning algorithm based on continuous iterative updating of model parameters. It does not need an accurate system model, and it can control dynamic systems with high uncertainty characteristics in a limited time [1]. Iterative learning control uses the learning strategy of 'learning in repetition', which has the mechanism of memory and correction. Its working mode is to control the controlled system, and then correct the unsatisfactory control signal according to the deviation between the output trajectory and the given trajectory, so as to generate a new control signal, and finally improve the tracking performance of the system. At present, it is widely used in many fields. With the rapid development and popularization of science and technology and the Internet, the distributed iterative learning algorithm uses distributed computing resources to learn large-scale data. It can distribute large-scale data on multiple nodes for parallel processing, so as to speed up the learning speed and improve the robustness of the algorithm [2]. However, in practical applications, data loss, delay, noise and other communication interference lead to errors or incomplete information exchange between agents. The research on iterative learning algorithms under communication interference can make the multi-agent system adjust adaptively according to the received information, and still maintain stability and consistency in the face of this interference, so as to improve the robustness of the system. This paper studies and designs a distributed iterative learning algorithm that can still operate effectively in the case of communication interference, to ensure the stability and accuracy of the whole system, which has strong theoretical and practical significance.

2 METHODOLOGY

2.1 Distributed Iterative Learning Algorithm

Distributed control utilizes local feedback information to control and coordinate multiple subsystems. If the communication network does not have limitations such as delay and packet loss, the performance of distributed control is intuitively inferior to centralized control. However, in practical network control, distributed control has more advantages such as robustness, scalability, and security, and centralized control [3, 4]. Therefore, studying distributed iterative learning algorithms is of great significance for solving the problem of networked data loss. An iterative learning algorithm is a learning algorithm that uses multiple-computer and multiple-core computing resources to complete big data learning, and it is usually applied to big data learning (such as Internet search, social networks, etc.) [5, 6]. The idea is to distribute big data across multiple machines or CPUs, with each machine (or CPU) responsible for local learning of the data. The learning results from each machine (or CPU) are aggregated for global parameter training, making full use of multiple machines and cores to significantly improve training speed and algorithm performance [7]. Specifically, the distributed iterative learning method includes the following steps:

1. Data distribution and segmentation: Divide the dataset into several smaller data blocks, distribute different data blocks to different computing nodes for computation, and reduce the computational workload and memory usage of each computing node.
2. Synchronization of initial model parameters. Each computing node initializes the model parameters locally, and then synchronizes its initialized model parameters to other nodes. The initial synchronization of model parameters is usually achieved through synchronization mechanisms such as parameter servers, which ensure that the initial model parameters across nodes remain consistent.
3. Iterative computation: Each node uses local data to calculate gradients and update parameters during the iteration process; Transmission of parameters.
4. Aggregation and synchronization of parameters: After all nodes complete their respective calculations, the server aggregates the parameters of each node, synchronizes these parameters to each node through weighted averaging, and updates each model parameter to a global model parameter.
5. Judging shutdown: By observing parameter variable information or objective function information, the current algorithm's shutdown is judged. If it meets the criteria, it stops, and if it does not meet the criteria, iterative calculation continues.
6. Result evaluation: Evaluate and validate the trained model to verify its performance and generalization ability.

The principle of the distributed iterative learning algorithm mainly involves the key steps such as distributed processing of data, synchronization and updating of model parameters, iterative calculation, and convergence judgment. Through the collaboration and synchronization of multiple computing nodes, the efficient processing and model training on large-scale datasets is achieved.

The completion of target tasks in multi-agent systems relies on accurate information transmission between individuals. However, due to the complexity of the environment and many uncertain factors, they are often subject to various signal interferences in practical applications, which can lead to signal errors, deterioration of the signal-to-noise ratio, and in severe cases, communication blockage [8, 9]. Therefore, considering the impact of communication disturbances is of great practical significance when studying multi-agent systems. This article considers a multi-agent system consisting of n followers and 1 virtual leader to solve the problem of networked data loss.

2.2 Description of the Controlled Object

In this paper, the communication topology diagram is used to describe the communication connection between agents, showing the information flow path and structure between agents, which can have ring, chain, tree, star and other structures. In multi-agent system, communication topology is very important for achieving consistency, coordination, and control strategy. Each topology has its advantages and limitations, which have an important impact on the performance, robustness, and scalability of multi-agent system. Choosing the appropriate communication topology depends on the specific application scenarios, communication constraints, and system design goals. For example, the star topology is easy to control, but the central node may become a bottleneck, while the grid topology provides high connectivity and robustness, but may require more communication resources and complex routing strategies. Therefore, the actual communication topology will vary according to the specific multi-agent system design and communication protocol. When designing the communication topology, it is necessary to consider the specific requirements of the system, such as communication range, energy consumption, network delay, etc. [10]. Consider the consensus problem $\mathcal{G}$ ($\mathcal{G} = (\mathcal{V} \cup \{v_0\}, \mathcal{E}, \mathcal{A})$) of a multi-agent system consisting of n followers and 1 virtual leader, and use a graph to represent the communication topology including the virtual leader.

Multi agent systems with dynamic characteristics:

$$\begin{cases} \dot{x}_{i,k}(t) = f\left(x_{i,k}(t)\right) + B(t)u_{i,k}(t), \\ y_{i,k}(t) = C(t)x_{i,k}(t), \end{cases} \qquad i \in \ell_n. \tag{1}$$

In Equation (1), $f(x_{i,k}(t))$ is a time-varying nonlinear term, where $k \in \mathbb{N}^+$ represents the number of iterations, i represents the i^{th} agent, and $t \in [0, T]$ is the time interval. The input, state, and output vectors are denoted as $u_{i,k}(t) \in \mathbb{R}^p$, $x_{i,k}(t) \in \mathbb{R}^{n_x}$, and $y_{i,k}(t) \in \mathbb{R}^m$, respectively. $B(t)$ and $C(t)$ are the system matrices of corresponding dimensions. In addition, I and 0 represent the identity matrix and zero matrix of appropriate dimensions, respectively. The notation $A > 0$ indicates that all elements of matrix A are positive (non-negative). Furthermore, $\ell_n = \{1, 2, \ldots, n\}$, $\mathbb{N}^+ = \{0, 1, 2, \ldots\}$, $\mathbf{1}_n = [1, 1, \ldots, 1]^T \in \mathbb{R}^n$, and $A \otimes B$ represents the Kronecker product of matrices A and B. The symbol $\operatorname{diag}\{x_1, x_2, \ldots, x_n\}$ denotes a diagonal matrix with elements x_i.

The above system includes a virtual leader ν_0, whose dynamic form is the same as that of the followers, $x_d(t)$, $u_d(t)$ and $y_d(t)$ is its state, controlling input and output variables. The main objective of this article is to design a controller based on iterative learning for the multi-agent system under consideration $u_{i,k}(k)$, in order to make Equation (2) hold true.

$$\lim_{k \to \infty} \left[y_d(t) - y_{i,k}(t)\right] = 0, \quad i \in \ell_n, \ t \in [0, T]. \tag{2}$$

For the above systems, the communication diagram of these systems can be described by Figure 1, where $u_{i,k}(t)$ is the control information and $N_i(t)$ represents the neighbor information of the nodes. Here, $y_0(t)$ is the information of the virtual leader, and the dotted line indicates that it is unclear whether node ν_i has received the information. $\omega_i(t)$ represents a communication disturbance. Let $z_{i,k}(t)$ be the output of node ν_i, and ν_j represents one of its neighboring nodes.

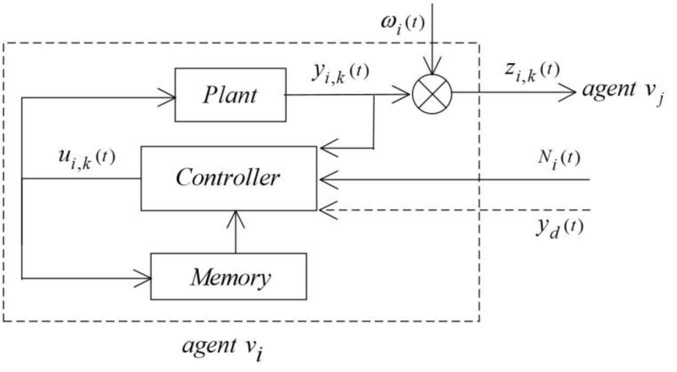


Figure 1. Communication topology diagram of the multi-agent system

Design a distributed protocol that can guide followers through virtual leaders to solve some data loss problems.

Assuming the following conditions for the nonlinear system that repeatedly operates within a finite time interval:

Assumption 1. The communication topology of the multi-agent system (2) is represented by the graph $\mathcal{G}$ shown in Figure 1. The graph $\mathcal{G}$ possesses a fixed topology structure containing a spanning tree, with the virtual leader ν_0 acting as the root node.

Assumption 2. For the multi-agent system (2) under consideration, the initial states of all agents can be reset to zero at each iteration. That is $x_{i,k}(0) = x_i(0) = x_d(0) = 0$, holds true for all $\ k \in \mathcal{N}_+,\ i \in \mathcal{I}_n$.

Assumption 3. $y_d(t)$ is achievable for the system. When $\omega_i(t) = 0$, there exist $u_d(t)$ and $x_d(0)$ that cause the output of the intelligent agent to be $y_d(t)$.

Assumption 4. The non-linear term $f(x_{i,k}(t))$ satisfies (2) and (2), i.e., $f(0) = 0$. Lipschitz condition:

$$\|f(x_{i,k+1}(t)) - f(x_{i,k}(t))\| \leq k_f \|x_{i,k+1}(t) - x_{i,k}(t)\| .$$

Here k_f is a certain constant.

Assumption 5. The matrices $B(t)$ and $C(t)$ are continuous, and $C(t)B(t)$ is full rank.

Assumption 6. The disturbance terms $\omega_i(t)$, $i \in I_n$, are continuous and differentiable.

Definition 1. Let $\|f\|$ be a norm of a vector f, such as the Euclidean norm.

The λ-norm $h : [0,T] \to \mathbb{R}^n$ of a vector function is defined as:

$$\|h\|_\lambda = \sup_{t\in[0,T]} \left\{ e^{-\lambda t}\|h(t)\| \right\}, \quad \lambda > 0.$$

This λ-norm satisfies: $\|c\|_\lambda = \|c\|$.

This $c \in \mathbb{R}^n$ is a constant vector with $\|h\|_\lambda \leq \sup_{t\in[0,T]} e^{-\lambda t}\|h(t)\| \leq e^{\lambda T}\|h\|$, $(t \in [0,T])$.

This $h : [0,T] \to \mathbb{R}^n$ is a vector function.

Definition 2. Assume $H : [0,T] \to \mathbb{R}^{p\times q}$ is a matrix function, the ∞-norm of H is defined as [8]:

$$\|H\|_\infty = \sup_{t\in[0,T]} \|H(t)\|.$$

Here, $\|\cdot\|_\infty$ represents the induced infinity norm of a matrix, analogous to the vector norm defined in Definition 1.

2.3 Design of D-Type Iterative Learning Controller

The output information of each follower is $z_{i,k}(t) = y_{i,k}(t) + \omega_i(t)$.

Express tracking error and control error as follows:

$$e_{i,k}(t) = y_d(t) - y_{i,k}(t) \quad \text{and} \quad \delta u_{i,k}(t) = u_d(t) - u_{i,k}(t).$$

A distributed control law based on D-type iterative learning control law is presented for the system, as follows:

$$u_{i,k+1}(t) = u_{i,k}(t) + \Psi(t)\left(\xi_{i,k}(t) - \dot{\eta}_{i,1}(t)\right). \tag{3}$$

Here $\Psi_k(t)$ is the time-varying learning gain matrix, where:

$$\xi_{i,k}(t) = \sum_{j\in N_i} a_{i,j}\left(z_{j,k}(t) - y_{i,k}(t)\right) + q_{i,0}\left(y_d(t) - y_{i,k}(t)\right), \tag{4}$$

$$\eta_{i,1}(t) = \sum_{j\in N_i} a_{i,j}\left(z_{j,1}(t) - y_{i,1}(t)\right). \tag{5}$$

This control protocol can be written in the vector form as follows:

$$u_{k+1}(t) = u_k(t) + \left((L+Q)\otimes\Psi(t)\right)\dot{e}_k(t). \tag{6}$$

2.4 Convergence Proof

Theorem 1. Consider a multi-agent system consisting of a virtual leader and n followers of Equation (2), whose relevant conditions satisfy Assumptions 1, 2, 3, 4, 5, 6, and whose distributed iterative learning control law is given by Equation (3). So, if the learning gain $\Psi_k(t)$ satisfies the inequality

$$\sup_{t\in[0,T]} \|I - ((L+Q)\otimes C(t)B(t)\Psi(t))\| < 1. \tag{7}$$

As the number of iterations k increases, the outputs of all follower agents converge to the output of the virtual leader, that is, Equation (2) holds.

Proof.

$$\begin{aligned} e_{k+1}(t) &= e_k(t) - (y_{k+1}(t) - y_k(t)) \\ &= e_k(t) - (I_n \otimes C(t))\left(x_{k+1}(t) - x_k(t)\right). \end{aligned} \tag{8}$$

And

$$\begin{aligned} x_{k+1}(t) - x_k(t) &= x_{k+1}(0) - x_k(0) \\ &\quad + \int_0^t \left[f(x_{k+1}(\tau)) - f(x_k(\tau)) + (I_n \otimes B(\tau))\left(u_{k+1}(\tau) - u_k(\tau)\right)\right] \mathrm{d}\tau. \end{aligned} \tag{9}$$

Here $f(x_k(t)) = [f(x_{1,k}(t)), f(x_{2,k}(t)), \ldots, f(x_{n,k}(t))]^T$

Because $x_{i,k}(0) = x_i(0) = x_d(0) = 0,\ i \in \ell_n,\ k \in N_+$. According to Equation (6), there are

$$\begin{aligned} x_{k+1}(t) - x_k(t) &= \int_0^t \left[f(x_{k+1}(\tau)) - f(x_k(\tau)) + ((L+Q)\otimes B(\tau)\Psi(\tau))\,\dot{e}_k(\tau)\right] \mathrm{d}\tau \\ &= \int_0^t \left[f(x_{k+1}(\tau)) - f(x_k(\tau))\right] \mathrm{d}\tau + ((L+Q)\otimes B(t)\Psi(\tau))\, e_k(t) \\ &\quad - \int_0^t \frac{d}{\mathrm{d}\tau}\left((L+Q)\otimes B(\tau)\Psi(\tau)\right) e_k(\tau)\mathrm{d}\tau. \end{aligned} \tag{10}$$

Substituting the above equation into Equation (7) yields

$$\begin{aligned} e_{k+1}(t) &= \left[I - ((L+Q)\otimes C(t)B(t)\Psi(t))\right] e_k(t) \\ &\quad - (I_m \otimes C(t))\left(\int_0^t \left[f(x_{k+1}(\tau)) - f(x_k(\tau))\right] \mathrm{d}\tau \right. \\ &\quad \left. - \int_0^t \frac{d}{\mathrm{d}\tau}\left((L+Q)\otimes B(\tau)\Psi(\tau)\right) e_k(\tau)\mathrm{d}\tau\right). \end{aligned} \tag{11}$$

Let

$$b_1 = \sup_{t\in[0,T]} \|I_m \otimes C(t)\|,$$

$$b_2 = \sup_{t\in[0,T]} \left\| \frac{d}{\mathrm{d}\tau}\left((L+Q)\otimes B(\tau)\Psi(\tau)\right)\right\|,$$

$$b_3 = \sup_{t\in[0,T]} \|(L+Q)\otimes B(t)\Psi(t)\|.$$

From Equation (11) and Definition 2:

$$\begin{aligned}\|e_{k+1}(t)\|_\infty \le \sup_{t\in[0,T]} &\|I-((L+Q)\otimes C(t)B(t)\Psi(t))\| \|e_k(t)\|_\infty \\ &+ b_1k_f\int_0^t \|x_{k+1}(\tau)-x_k(\tau)\|_\infty \mathrm{d}\tau + b_1b_2\int_0^t \|e_k(\tau)\|_\infty \mathrm{d}\tau.\end{aligned} \tag{12}$$

Multiplying both sides of the above equation by $e^{-\lambda t}$ yields:

$$\begin{aligned}\|e_{k+1}(\cdot)\|_\lambda \le& \sup_{t\in[0,T]} \|I-((L+Q)\otimes C(t)B(t)\Psi(t))\| \|e_k(\cdot)\|_\lambda \\ &+ b_1k_f\int_0^t e^{-\lambda(t-\tau)}\|x_{k+1}(\tau)-x_k(\tau)\|_\infty e^{-\lambda\tau}\mathrm{d}\tau \\ &+ b_1b_2\int_0^t e^{-\lambda(t-\tau)}\|e_k(\tau)\|_\infty e^{-\lambda\tau}\mathrm{d}\tau \\ \le& \sup_{t\in[0,T]} \|I-((L+Q)\otimes C(t)B(t)\Psi(t))\| \|e_k(\cdot)\|_\lambda \\ &+ b_1k_f\|x_{k+1}(\cdot)-x_k(\cdot)\|_\lambda \frac{1-e^{-\lambda T}}{\lambda} + b_1b_2\frac{1-e^{-\lambda T}}{\lambda}\|e_k(\cdot)\|_\lambda.\end{aligned} \tag{13}$$

Here $\int_0^t e^{-\lambda(t-\tau)}\mathrm{d}\tau = \frac{1-e^{-\lambda t}}{\lambda}$. Similarly, according to Equation (10), there are

$$\|x_{k+1}(\cdot)-x_k(\cdot)\|_\lambda \le \left(b_3+b_2\frac{1-e^{-\lambda t}}{\lambda}\right)\left(1-k_f\frac{1-e^{-\lambda t}}{\lambda}\right)^{-1}\|e_k(\cdot)\|_\lambda.$$

Substituting the above equation into Equation (12) yields

$$\begin{aligned}\|e_{k+1}(\cdot)\|_\lambda \le& \sup_{t\in[0,T]} \|I-((L+Q)\otimes C(t)B(t)\Psi(t))\| \|e_k(\cdot)\|_\lambda \\ &+ \frac{b_1k_f\left(b_3+b_2\frac{1-e^{-\lambda T}}{\lambda}\right)\frac{1-e^{-\lambda T}}{\lambda}}{1-k_f\frac{1-e^{-\lambda T}}{\lambda}}\|e_k(\cdot)\|_\lambda + b_1b_2\frac{1-e^{-\lambda T}}{\lambda}\|e_k(\cdot)\|_\lambda.\end{aligned} \tag{14}$$

Because λ can be arbitrarily selected, the last two terms on the right side of Equation (14) tend to zero and can be ignored when λ is chosen to be sufficiently large. Therefore, when condition (7) is satisfied, as the number of iterations increases ($k \to \infty$), the norm $\|e_k(\cdot)\|_\lambda$ will converge to 0, meaning that the objective Equation (2) holds. $\square$

3 RESULTS AND DISCUSSION

3.1 Experimental Design

Experiment 1 (Proof of Algorithm Feasibility)**.** Consider a multi-agent system consisting of four follower agents and a virtual leader.

This multi-agent communication topology is shown in Figure 2. It is a network composed of five nodes, which are labeled as ν_0, ν_1, ν_2, ν_3 and ν_4, respectively. The arrow in the figure indicates the flow direction of information, that is, the communication path from one node to another node. This type of graph is usually used to describe the communication structure and information flow in multi-agent systems.

In this particular graph, ν_0 is a central node, which has a direct communication connection with all peripheral nodes (ν_1, ν_2, ν_3 and ν_4). This means that ν_0 can send information directly to any peripheral node and receive information from any peripheral node. In addition, peripheral nodes also have certain communication capabilities. For example, ν_1 can directly send information to ν_3, while ν_2 can directly send information to ν_4. However, not all peripheral nodes can communicate with each other directly; For example, ν_1 cannot communicate directly with ν_2 or ν_4, but needs to exchange information indirectly through ν_0 as an intermediary.

This communication topology may have many uses in practical applications, such as in the design of distributed control systems, sensor networks or other systems requiring centralized management and decentralized execution. Through such a topology, efficient communication and information integration can be realized, and a certain degree of autonomy and fault tolerance can also be allowed.

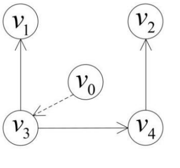


Figure 2. Communication topology used in Experiment

Its dynamic characteristics are as follows:

$$\begin{cases} \dot{x} = \begin{bmatrix} 0.6\sin(x_i(1)) + 0.3x_i(2) \\ -0.1\sin(x_i(2)) \end{bmatrix} + \begin{bmatrix} 1 & 1.5 \\ 1 & -1 \end{bmatrix} u_i, \\ y_i = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} x_i. \end{cases} \tag{15}$$

Here $x_i = [x_i(1), x_i(2)]^T$, the trajectory of the virtual leader is assumed to be:

$$y_d = \begin{bmatrix} y_{d1} \\ y_{d2} \end{bmatrix} = \begin{bmatrix} t + 3\sin(t) \\ t + 3\cos(2t) - 2 \end{bmatrix}, \quad \forall t \in [0, 7]. \tag{16}$$

The communication topology between intelligent agents is shown in Figure 2, and for simplicity, the weights are taken as 0 or 1. So the $\mathcal{G}$ Laplacian matrices L and Q of the graph are:

$$L = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{17}$$

The $L + Q$ eigenvalues of the matrix are $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$.
Let

$$\Psi(t) = 0.5(CB)^{-1} = \begin{bmatrix} 0.2667 & -0.0333 \\ -0.0667 & 0.1333 \end{bmatrix}. \tag{18}$$

Substituting the relevant quantities in the equation into condition (7) yields:

$$\sup_{t\in[0,T]} \|I - ((L + Q) \otimes C(t)B(t)\Psi(t))\| \approx 0.995 < 1.$$

Assuming the disturbance is:

$$\omega_1(t) = \begin{bmatrix} 0.3\cos(0.7t) \\ 0.5\sin(1.5t) \end{bmatrix}, \tag{19}$$

$$\omega_2(t) = \begin{bmatrix} 0.3\cos(0.6t) \\ 0.3\sin(0.7t) \end{bmatrix}, \tag{20}$$

$$\omega_3(t) = \begin{bmatrix} 0.5\sin(0.7t) - 0.2\cos(0.8t) \\ 0.2\cos(0.7t) \end{bmatrix}, \tag{21}$$

$$\omega_4(t) = \begin{bmatrix} 0.6\cos(0.6t) - 0.2\sin(0.6t) \\ 0.4\sin(1.5t) \end{bmatrix}. \tag{22}$$

Experiment 2 (Proof of Algorithm Improvement)**.** Still using the scenario from Experiment 1, consider a multi-agent system consisting of four follower agents and one virtual leader. Using its dynamic characteristics as follows:

$$\begin{cases} \dot{x} = \begin{bmatrix} 0.6\sin(x_i(1)) + 0.3x_i(2) \\ -0.1\sin(x_i(2)) \end{bmatrix} + \begin{bmatrix} 1 & 1.5 \\ 1 & -1 \end{bmatrix} u_i, \\ y_i = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} x_i. \end{cases} \tag{23}$$

Here $x_i = [x_i(1), x_i(2)]^T$, the trajectory of the virtual leader is assumed to be:

$$y_d = \begin{bmatrix} y_{d1} \\ y_{d2} \end{bmatrix} = \begin{bmatrix} t + 3\sin(t) \\ t + 3\cos(2t) - 2 \end{bmatrix}, \quad \forall t \in [0, 7]. \tag{24}$$

Using the above dynamic system, the first one does not use the designed distributed protocol as the original type, and the second one uses the designed distributed protocol as an improvement, both under the same disturbance.

The communication topology between intelligent agents is shown in Figure 2, and for simplicity, the weights are taken as 0 or 1. So the $\mathcal{G}$ Laplacian matrices L and Q of the graph are:

$$L = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{25}$$

The $L + Q$ eigenvalues of the matrix are $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$.

Let

$$\Psi(t) = 0.5(CB)^{-1} = \begin{bmatrix} 0.2667 & -0.0333 \\ -0.0667 & 0.1333 \end{bmatrix}. \tag{26}$$

Substituting the relevant quantities in the equation into condition (7) yields:

$$\sup_{t\in[0,T]} \|I - ((L + Q) \otimes C(t)B(t)\Psi(t))\| \approx 0.995 < 1.$$

Assuming the disturbance is:

$$\omega_1(t) = \begin{bmatrix} 0.3\cos(0.7t) \\ 0.5\sin(1.5t) \end{bmatrix},$$

$$\omega_2(t) = \begin{bmatrix} 0.3\cos(0.6t) \\ 0.3\sin(0.7t) \end{bmatrix},$$

$$\omega_3(t) = \begin{bmatrix} 0.5\sin(0.7t) - 0.2\cos(0.8t) \\ 0.2\cos(0.7t) \end{bmatrix},$$

$$\omega_4(t) = \begin{bmatrix} 0.6\cos(0.6t) - 0.2\sin(0.6t) \\ 0.4\sin(1.5t) \end{bmatrix}.$$

3.2 Experimental Results Display and Analysis

Results display and analysis of Experiment 1.

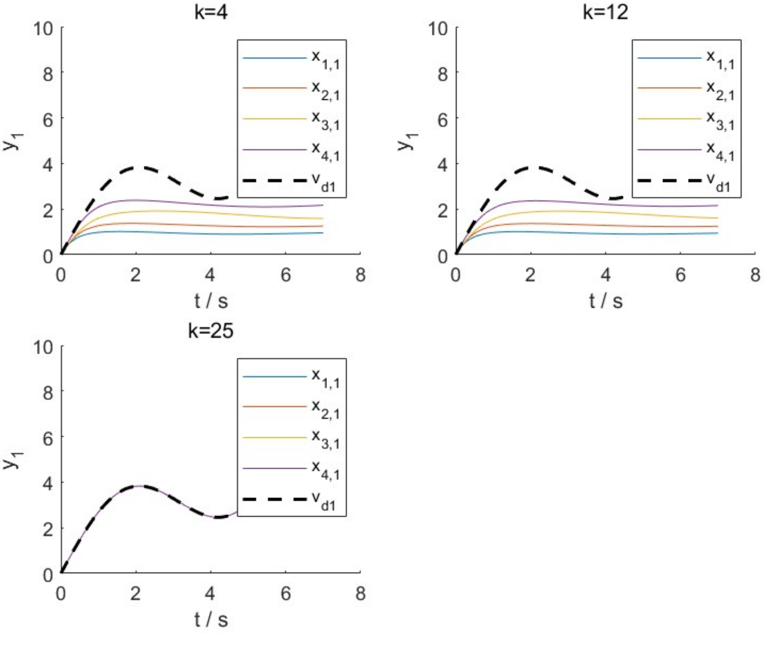


Figure 3. Actual output trajectory of y_1 for all improved intelligent agents

Figures 3, 4, and 5 present the simulation results regarding Theorem 1. From Figures 3 and 4, it can be seen that the outputs of all followers tracked the output

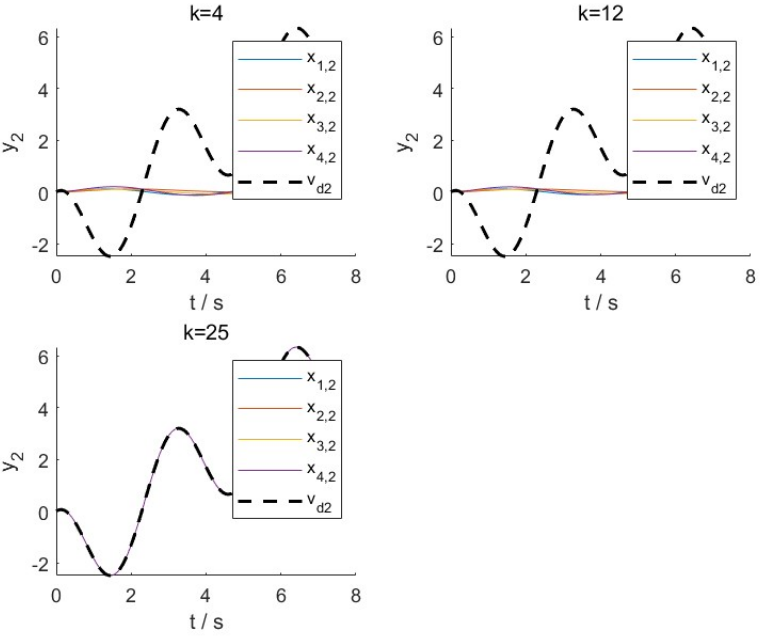


Figure 4. Actual output trajectory of y_2 for all agents of the improved model

of the virtual leader at $k = 25$. From Figure 5, it can be seen that $k = 25$ is the error criterion of 0, indicating successful suppression of repetitive interference and verification of the effectiveness of the experimental results.

Results display and analysis of Experiment 2.

Figures 6, 7 and 8 present comprehensive simulation results of the baseline iterative learning algorithm. A meticulous comparative analysis of these graphical representations reveals a distinct trend: as the iteration count k increases, the agent's motion trajectory progressively converges toward that of the virtual leader, demonstrating the algorithm's inherent convergence properties. In the original algorithmic framework, each agent's trajectory evolved autonomously, governed solely by the predefined virtual leader trajectory and external stochastic disturbances, without any substantive inter-agent communication or collaborative mechanisms. This architectural limitation resulted in a gradual reduction of the error norm over time; however, the absence of collective coordination led to suboptimal convergence rates and substantial residual errors. The enhanced algorithm addresses these limitations through the implementation of a sophisticated distributed control strategy. This innovative approach facilitates effective inter-agent communication and coordination via adjacency matrices, where each agent's trajectory is influenced not only by the virtual leader and external perturbations

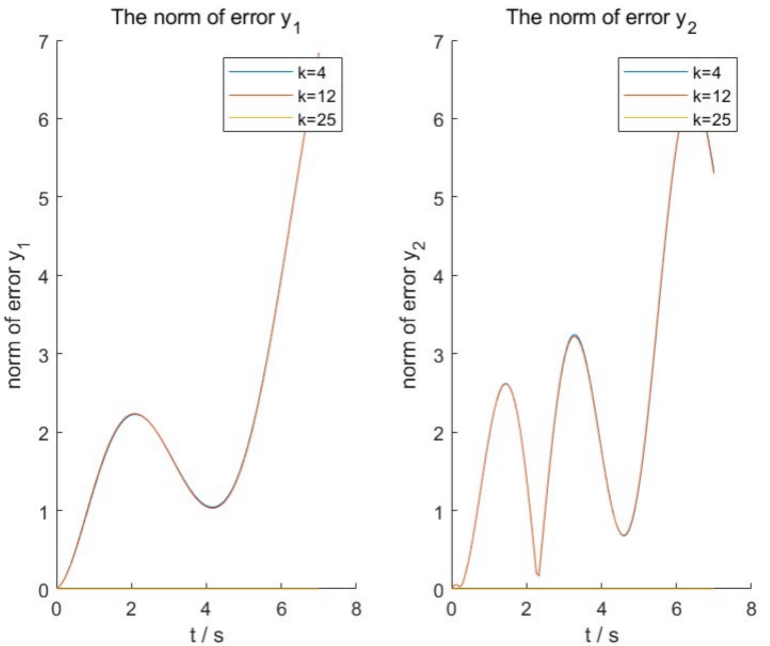


Figure 5. Error criteria for improved $k = 4$, $k = 12$, and $k = 25$

but also by the state information of adjacent agents. This distributed collaborative framework enables agents to maintain precise alignment with the virtual leader, thereby significantly enhancing system-wide synchronization and convergence dynamics. A comparative examination of Figures 6, 7 and 8 with Figures 3, 4 and 5 reveals substantial performance improvements post-optimization. The pre-optimization scenario was characterized by independent and divergent trajectory patterns across various iteration intervals, lacking essential synergistic interactions. This resulted in widespread trajectory distribution and substantial error norms. While the error norm exhibited a decreasing trend with increasing iterations, the convergence rate remained suboptimal. The post-optimization scenario, facilitated by the distributed D-type controller, demonstrates remarkable trajectory synchronization, tighter trajectory clustering, and significantly reduced error norms. The distributed control strategy has markedly accelerated error convergence, enabling agents to more effectively track the virtual leader's trajectory.

This algorithmic enhancement not only resolves the issue of data loss in iterative learning algorithms but also establishes a robust foundation for future applications. Particularly in the domain of unmanned systems, distributed iterative learning algorithms present extensive application potential. Through the implemen-

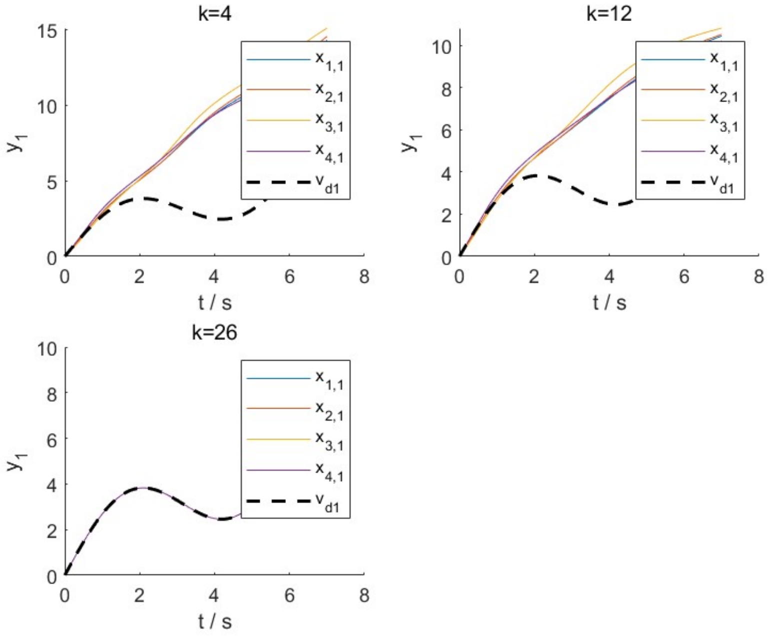


Figure 6. Actual output trajectories of y_1 for all agents of the primitive type

tation of a distributed model training framework, unmanned systems can achieve collective intelligence and collaborative learning capabilities. Even when confronted with challenges such as data loss or connectivity disruptions, the system maintains operational stability and efficiency. This framework enables unmanned systems to sustain global model updates through the exchange of critical parameters, ensuring continuous learning and effective decision-making in complex, dynamic environments.

The implementation of this advanced control strategy represents a significant paradigm shift in multi-agent system coordination, offering enhanced robustness and adaptability in various operational scenarios. The integration of distributed control mechanisms with iterative learning algorithms has demonstrated substantial improvements in system performance metrics, particularly in terms of convergence rates and error minimization. These advancements hold particular promise for applications requiring high-precision coordination and rapid adaptation to dynamic environmental conditions.

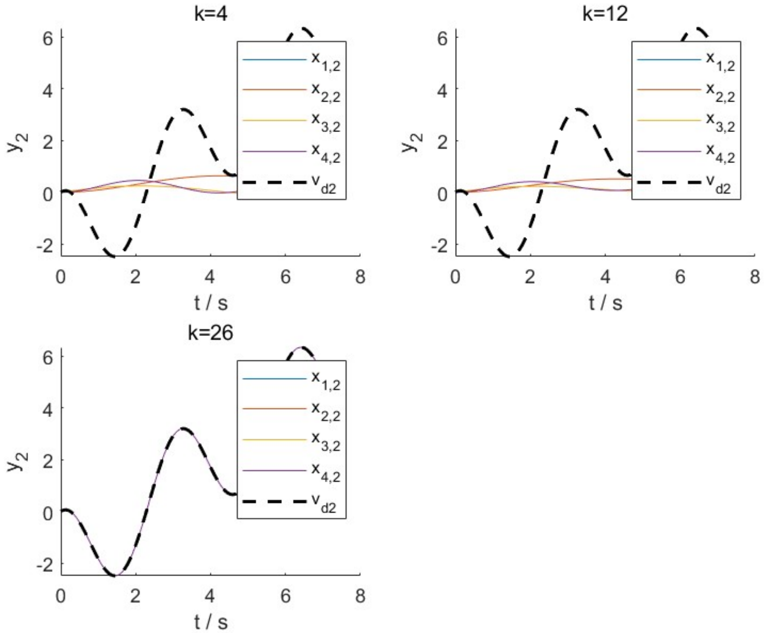


Figure 7. Actual output trajectories of y_2 for all agents of the primitive type

4 CONCLUSIONS

This study mainly focuses on optimizing distributed iterative learning algorithms, aiming to solve the problem of data loss in networked systems. By constructing a model based on multi-agent systems, which includes a virtual leader and several followers, this study proposes a D-type control strategy that utilizes node neighbor information and self-control information to regulate the system. In theory, even in the presence of interference and data loss, the proposed algorithm can achieve convergence as the number of iterations increases. The simulation results confirm that the improved algorithm can still run stably in the event of data loss, and exhibits faster convergence speed compared to the unimproved D-type iterative algorithm.

Although this research has made some achievements in optimizing the distributed iterative learning algorithm, there are still some shortcomings and limitations. The following is the revision and enrichment of the conclusion:

First of all, this study is based on some assumptions, such as a zero initial state and fixed topology, which may limit the application of the algorithm in the actual complex network environment. The following is a discussion and supplement of future research directions:

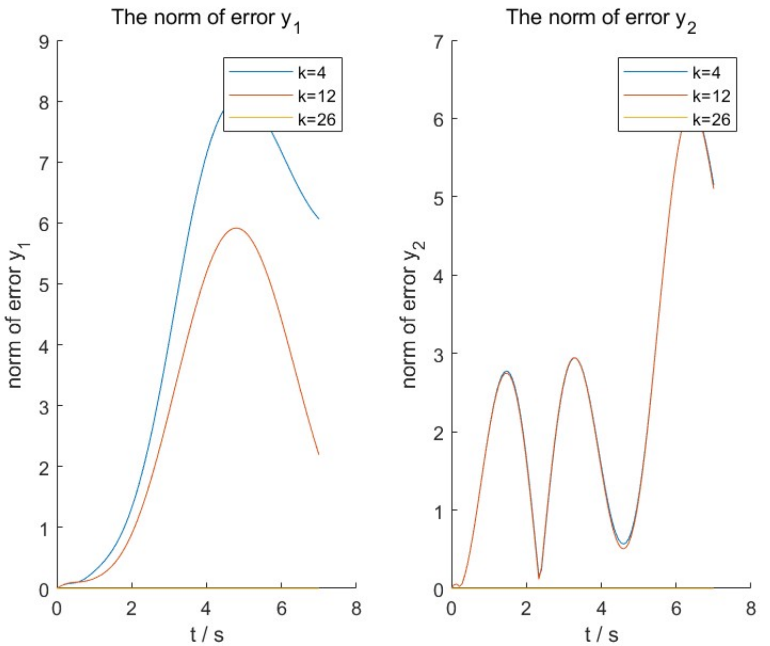


Figure 8. Error criteria for $k = 4$, $k = 12$, and $k = 26$ of the original type

1. This study verifies the effectiveness of the algorithm under the fixed topology, but the performance of the algorithm needs to be further verified when the topology is not fixed. Future research will focus on exploring the applicability of the algorithm in a non-fixed topology, and verify its effectiveness through simulation experiments.
2. In this study, a D-type control strategy is adopted, and its excellent performance under specific assumptions is shown. However, the comparative analysis of p-type and PD control strategies has not been involved in the study. Future work will include a comparative study of these different control strategies to determine better control strategies under different conditions.
3. In addition, this study has not applied the algorithm to more complex actual scenes. Future research will consider applying the algorithm to complex environments such as driverless systems to test its performance and adaptability in such scenarios.

To sum up, this study provides a new perspective for the optimization of distributed iterative learning algorithm, and provides a theoretical basis and practical guidance for solving the problem of network data loss. Future research will focus on solving the above limitations, broadening the application scope of the algorithm, enhancing its adaptability and practicability, so as to better serve the control requirements of the actual network system.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Data Sharing Agreement

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

REFERENCES

[1] Li, R.—Han, Z.: Survey of Iterative Learning Control. Control and Decision Making, Vol. 2005, 2005, No. 9, pp. 961–966, doi: 10.3321/j.issn:1001-0920.2005.09.001 (in Chinese).

[2] Huang, K.—Ai, Q.—Zhang, Y.—Zhou, J.: Multi Region Interconnected Distributed Power Flow Algorithm Based on Multi-Agent Systems. Power Automation Equipment, Vol. 40, 2020, No. 7, pp. 174–180, doi: 10.16081/j.epae.202005020 (in Chinese).

[3] Meng, D.: Convergence Conditions for Solving Robust Iterative Learning Control Problems under Nonrepetitive Model Uncertainties. IEEE Transactions on Neural Networks and Learning Systems, Vol. 30, 2019, No. 6, pp. 1908–1919, doi: 10.1109/TNNLS.2018.2874977.

[4] Wang, G.—Wang, R.—Yi, D.—Zhou, X.—Zhang, S.: Iterative Learning Formation Control via Input Sharing for Fractional-Order Singular Multi-Agent Systems with Local Lipschitz Nonlinearity. Fractal, Vol. 8, 2024, No. 6, Art. No. 347, doi: 10.3390/fractalfract8060347.

[5] Wang, C.—Zhou, Z.: Distributed Iterative Learning Consensus Tracking for Singular Partial Differential Multi-Agent Systems under Fixed and Iteration-Varying Topologies. Journal of the Franklin Institute, Vol. 361, 2024, No. 13, Art. No. 107030, doi: 10.1016/j.jfranklin.2024.107030.

[6] Xu, Q.—Mai, Q.—Wang, B.—Wan, K.—Wei, Y.: Distributed Adaptive Iterative Learning Control for 2D Multi Agent Systems. Electronics Letters, Vol. 60, 2024, No. 13, Art. No. e13263, doi: 10.1049/ell2.13263.

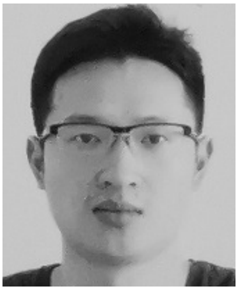
[7] Wang, H.—Li, H.: Event-Triggered Iterative Learning Control for Nonlinear Multi-Agent Systems with Data Random Packet Dropouts. Control Theory and Applications, Vol. 39, 2022, No. 9, pp. 1688–1698, doi: 10.7641/CTA.2022.10849 (in Chinese).

[8] Wang, Y.—Sun, J.: Research on Packet Loss Compensation of Networked Control System. Instrumentation User, Vol. 26, 2019, No. 11, pp. 91–93 (in Chinese).

[9] Wang, Y.—Mo, L.—Yang, T.: Time-Varying Constrained Consensus of Asynchronous Discrete-Time Multi-Agent Systems. In: Jia, Y. (Ed.): Advances in Guidance, Navigation and Control. Springer, Singapore, Lecture Notes in Electrical Engineering, Vol. 1283, 2024, pp. 13–24, doi: 10.1007/978-981-97-8650-3_2.

[10] Guo, G.—Kang, J.: A Review of Distributed Optimization of Multi-Agent Systems with Complex Dynamics. Control and Decision Making, Vol. 39, 2024, No. 7, pp. 2113–2124, doi: 10.13195/j.kzyjc.2023.1788 (in Chinese).

Xiaoyan Cheng received her Ph.D. degree in control science and engineering from the Yanshan University, Qinhuangdao, China, in 2020. She is currently Associate Professor with the PLA Army Arms University, China. Her research interests include cooperative control of multi-agent systems, swarm intelligence, and distributed optimization algorithms.



Wei Tang received his M.S. degree in weapon system utilization engineering from the Academy of Armored Force Engineering, Beijing, China, in 2014. He is currently Associate Professor with the PLA Army Arms University, China. His research interests include mission planning and intelligent control technologies.

Hui He is Professor at the PLA Army Arms University, China, received her Master's degree in language linguistics and applied linguistics from the Warwick University in the UK in 2009. Her research interests range from applied linguistics, semantics to second language acquisition.

Hao Wang received his B.S. degree from the Army Academy of Armored Forces, PLA, China, in 2024. His undergraduate research focused on distributed iterative learning algorithms.