Consensus Tracking via Iterative Learning for Multi-Agent Systems With Random Initial States

WEI CAO1, JINJIE QIAO2, AND MING SUN1, (Member, IEEE)

1College of Computer and Control Engineering, Qiqihar University, Qiqihar 161006, China
2College of Economics and Management, Qiqihar University, Qiqihar 161006, China

Corresponding author: Wei Cao (yiyuqq168@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 71803095 and Grant 61872204, in part by the Humanity and Social Science Youth Foundation of Ministry of Education under Grant 18YJC790130, in part by the Fundamental Research Funds in Heilongjiang Provincial Universities under Grant 135409311, and in part by the Natural Science Foundation of Heilongjiang Province under Grant LH2020G009.

ABSTRACT In this paper, a distributed consensus iterative learning control algorithm is proposed for the finite-time consensus tracking of multi-agent systems with random initial states. The tracking errors from the agent itself and its neighbors are applied to successively rectify the control protocol when only some agents can obtain the desired trajectory information. Additionally, a time interval is designed in the control protocol, and the random initial state errors are rectified, one-by-one, in the time interval. The interval can gradually shorten as the number of iterations increases. Furthermore, the convergence of the algorithm is theoretically proven by the contraction mapping method, and the convergence condition is derived. The proposed algorithm can cause the time interval to gradually shorten with the iterations increase, during the time, the desired trajectory cannot be consistently tracked due to the random initial state errors. As a consequence, the algorithm gradually widens the time interval for full consensus tracking of the multi agents. Finally, the simulation examples are provided to verify the effectiveness of the algorithm.

INDEX TERMS Multi-agent systems, iterative learning control, random initial state, finite-time, consensus.

I. INTRODUCTION Multi-agent systems are composed of several single agents. Each agent coordinates and cooperates to achieve a common goal by communicating with each other in accordance with the control protocol. Multi-agent systems can complete the tasks that are difficult to complete with single agents, and thus, they enhance the systems’ ability to solve complex problems. Additionally, multi-agent systems have the advantages of strong reliability, robustness and flexibility [1], [2]. With the development of sensors, computers, communications and other technologies, multi-agent systems have a wide range of applications in mobile sensors [3], drone group control [4], [5], event-triggered control of network systems [6], [7] and multi-robot formation control areas [8], [9].

Because of the collaborative control of multi-agent systems, the consensus problem is a key issue and is also one of the hot research topics in the control field. A so-called consensus means that individuals in a multi-agent system use their own and their neighbor’s information to construct a control protocol, so the state or output of all the individuals eventually converge [10]. In recent years, scholars have studied the consensus of multi-agent systems via Lyapunov functions, matrix theory, algebraic graph theory and other methods and have obtained profitable research results [11]–[15]. However, most current research results are about asymptotic consensus tracking with the increased times, rather than full consensus tracking in the entire finite time interval.

For the systems implementing repetitive tasks in finite time intervals, iterative learning controls can achieve the full tracking of the desired trajectory in the entire finite time interval [16]. The idea is to use the deviation signal from the actual output and the desired trajectory to successively rectify the control signal until zero error tracking is achieved in a finite time interval. Due to the design of the iterative learning controllers only needs less knowledge of the dynamic characteristics, and the controller has the advantages of having a simple structure and executing a small amount of online calculations. Thus the control technique has received a great amount of attention from researchers since the technique was
proposed by Arimoto [16], and it has obtained lots of research results [17]–[21].

To ensure the convergence of the algorithm, the existing research results often assume that the initial state of the system satisfies certain conditions when designing iterative learning controllers. These assumptions can be divided into four categories. In the first category, it is assumed that the initial state error is 0 [22]–[24], that is, the initial state of the system must be equal to the desired initial state at each iteration. In the second category, it is assumed that the initial state error is fixed [10], [25], [26], that is, the initial state at each iteration is fixed, but it may not be equal to the desired initial state. However, in the running process of the actual system, it is difficult to keep the initial state of each iteration completely consistent, so these two kinds of strict repetition conditions limit the application of iterative learning control. To get closer to the actual operation of the system, some scholars have proposed that the initial state error is concentrated in a bound [27], [28], which is the third kind of assumption for the initial state. This kind of assumption means that the tracking error can only converge to one bound and is related to the initial state error, so zero error tracking of the output to the desired trajectory cannot be achieved. To eliminate the influence of the initial error on tracking accuracy, an iterative learning control algorithm with an initial rectifying action is proposed in [29], and this algorithm can achieve complete tracking after a specified time interval. However, the algorithm still requires that the initial state error be bounded and cannot satisfy the case of a random initial state. In recent years, in order to further relax the restrictions on the initial positioning conditions, some scholars have allowed the initial state of each iteration to be a random value [17], [30], [31], which is the fourth kind of hypothesis. This kind of assumption can obtain zero-error complete tracking in a specified interval. However, to date, the fourth kind of hypothesis only analyses the convergence of a single agent, and does not solve the problem of consistent tracking of the multi-agent systems with a random initial state.

In light of the advantages of iterative learning controls, many scholars have applied iterative learning controls into the coordination controls of multi-agent systems [24], [10], [32]–[38]. In [10], the distributed iterative learning control method was designed to solve the formation control problem of discrete-time multi-agent systems. For a class of non-affine nonlinear multi-agent systems, a kind of iterative learning control algorithm was proposed in [32], if the number of iterations was sufficiently large, so that the agent outputs can reach consensus. For discrete-time linear multi-agent systems with communication delays and switching topologies, a kind of iterative learning control protocol was also devised in [33], obtaining consensus for multi-agent outputs in a finite time interval. Similarly, [34] presented a kind of iterative learning control protocol by using the Bernoulli sequence to describe the data dropout, so that the output of non-linear multi-agent systems with data dropout can achieve consensus tracking of the desired trajectory in a finite time. To solve the problem of consensus tracking for linear multi-agent systems with a variable communication topology, an input sharing iterative learning control protocol was put forward in [35]. In [36], for an output-saturated multi-agent system, an iterative learning control protocol was proposed under a fixed topology to achieve consensus tracking of the system output within a finite time, and the results were generalized to the multi-agent systems with switching topologies.

All the abovementioned results set the initial state of the system based on two assumptions in order to guarantee the convergence of the algorithms. The first assumes that the initial state of the system is exactly the same as the expected state at each iteration [24], [35]; the other assumes that the initial state stays fixed at each iteration [10], [32]–[34], [36], [37]. The two assumptions are too strict for multi-agent systems in practical operations. As a result, the above conclusions do not apply if the initial state of the multi-agent is different at each iteration. To relax the restrictions on the initial state of the multi-agents, the article presented recently in [38] designed a non-parametric uncertain multi-agent system using the Lyapunov method to design a consensus iterative learning controller to solve the consensus tracking problem for multi-agent systems with random initial states. However, [38] cannot cause the interval of the consensus tracking to gradually widen with the increase of iterations, but only achieves consensus tracking in a predetermined interval of a finite time interval. That is, this approach cannot fully achieve consensus zero error tracking throughout the finite time interval (0, T).

To resolve the finite-time consensus tracking problem of multi-agent systems with random initial states, a distributed consensus iterative learning control algorithm is proposed. The expected trajectory in this algorithm is generated by the introduced virtual leader, and the expected trajectory information is only available to some agents. Using the tracking error of the agent itself and its neighbours, the control protocols are rectified one-by-one. Additionally, a time interval is designed in the control protocol, and the random initial state errors are rectified one-by-one in this time interval, leading to shortening gradually the interval with the increase of the iterations. The contraction mapping method theoretically demonstrates that, after the designed time interval, the multi-agent output can be completely consistent in tracking the desired trajectory as the iterations increase, and the time interval of the full consensus tracking is gradually widened as the iterations increase, ultimately achieving full-consensus zero error tracking throughout the finite time interval (0, T). Finally, the effectiveness of the proposed algorithm is further verified by simulation examples.

II. PROBLEM DESCRIPTION

Consider a multi-agent system composed of N same-structure nonlinear agents, and assume that the initial state of each agent is random at each iteration and each agent runs repetitively in $t \in [0, T]$. The dynamic model of the $j$-th nonlinear
agent is expressed as follows:

\[
\begin{align*}
\dot{x}_{j,k}(t) &= f(x_{j,k}(t), t) + Bu_{j,k}(t), \\
y_{j,k}(t) &= Cx_{j,k}(t),
\end{align*}
\]  

(1)

where \(x_{j,k}(t) \in \mathbb{R}^n\), \(u_{j,k}(t) \in \mathbb{R}^r\), and \(y_{j,k}(t) \in \mathbb{R}^m\) represent the state, control input and output of the \(j\)-th agent, respectively. Nonlinear function \(f(x_{j,k}(t), t)\) is unknown, \(B \in \mathbb{R}^{r \times r}\) is a known input matrix, which is a right invertible matrix, \(C \in \mathbb{R}^{m \times n}\) is a known output matrix. Subscript \(k\) denotes the number of iterative learning, \(k \in \{1, 2, \ldots, N\}\) denotes the \(j\)-th agent, and \(t \in [0, T]\) denotes the finite time of the repetitive operation.

**Assumption 1:** Each agent’s nonlinear function \(f(\cdot, \cdot)\) satisfies the global Lipschitz condition for all \(t \in [0, T]\). That is, there exists a constant \(k_f > 0\) for each agent during \(t \in [0, T]\), then the following inequality holds:

\[
\|f(x_1(t), t) - f(x_2(t), t)\| \leq k_f \|x_1(t) - x_2(t)\|.
\]

(2)

**Assumption 2:** For each agent there exists a unique desired control \(u_d(t)\), accordingly the state and output are denoted as \(x_d(t)\) and \(y_d(t)\), i.e.,

\[
\begin{align*}
\dot{x}_d(t) &= f(x_d(t), t) + Bu_d(t), \\
y_d(t) &= Cx_d(t).
\end{align*}
\]

(3)

The desired trajectory is given for the agents by introducing the virtual leader into the study. Due to multi-agents exchange of information with each other through the communication topology, the information each agent can obtain depends on the communication topology structure between the multi-agents. In addition, thus, relevant knowledge of graph theory is introduced in order to lay the foundation for subsequent analysis.

The communication topology between the \(N\) agents is described as a weighted graph \(\xi = (\nu, \varepsilon, A)\), where \(A, \varepsilon \subseteq \nu \times \nu\) and \(\nu = \{v_1, v_2, \ldots, v_N\}\) denote the adjacency matrix, edge and node of the graph, respectively. Each node in \(\nu\) represents an agent. \(L = D - A\) is defined as the Laplacian matrix of graph \(\xi\), where \(D = \text{diag}(d_1, d_2, \ldots, d_N)\) holds, and \(d_i = \sum_{j \in N_i} a_{ij}\) is called as out degree of the node \(i\). \((i, j) \in \varepsilon\) is represented as an edge composed of nodes \(i\) and \(j\); the adjacency matrix \(A\) is represented as the connection of nodes \(i\) and \(j\); its diagonal element is \(a_{ii} = 0\). If there is a connection between nodes \(i\) and \(j\), then \(a_{ij} = a_{ji} > 0\); otherwise \(a_{ij} = 0\), where \(i \in N_i\), and \(N_i = \{j\}\) is the neighbour node collection of node \(i\).

When the virtual leader is introduced, if the leader is agent 0, then the new graph \(\xi' = (0 \cup \nu, \bar{\varepsilon}, \bar{A})\) is represented as the information relationships between all agents including the virtual leaders, where \(\bar{\varepsilon}\) and \(\bar{A}\) are denoted as the new edge and adjacency matrix, respectively. At this time, \(s_i\) represents the connection between the agent \(i\) and the virtual leader. \(s_i = 0\) indicates agent \(i\) has no connection with virtual leader; \(s_i > 0\) indicates that agent \(i\) has a direct connection with the virtual leader.

According to the aforementioned analysis, in order to make the multi-agent whose initial state is random at each iteration have the ability of consensus tracking of the desired trajectory in the finite time interval and its tracking interval widen gradually with the increase of iterations, the following distributed iterative learning control algorithm is designed as

\[
u_{j,k+1}(t) = u_{j,k}(t) + \Gamma \sum_{i \in N_j} a_{ij}(\ddot{e}_{j,k}(t))
- \dot{e}_{i,k}(t)) + s_j e_{j,k}(0) + \theta_{k,h}(t)X_{j,k}(0),
\]

(4)

\[
X_{j,k}(0) = \Gamma \sum_{i \in N_j} a_{ij}(e_{j,k}(0) - e_{i,k}(0)) + s_j e_{j,k}(0) + B_{R}^{-1}(x_{j,k}(0) - x_{j,k+1}(0)),
\]

(5)

\[
\theta_{k,h}(t) = \left\{ \begin{array}{ll}
2 \beta \frac{k}{h} \left(1 - \frac{\beta}{h} \right), & t \in [0, \frac{h}{\beta} ) \\
0, & t \in [\frac{h}{\beta}, T].
\end{array} \right.
\]

(6)

where \(\beta > 1\) is an adjustment parameter; \(0 < h < T\) is a predetermined time constant; \(\Gamma\) is a learning gain matrix; \(e_{j,k}(t) = y_d(t) - y_{j,k}(t)\) and \(e_{i,k}(t) = y_d(t) - y_{i,k}(t)\) are the tracking errors of the \(j\) and \(i\)-th agent, respectively; \(B_{R}^{-1}\) is the right inverse matrix of \(B\).

For convenience of analysis, denote

\[
\begin{align*}
x_k(t) &= [x_{1,k}^T(t), x_{2,k}^T(t), \ldots, x_{N,k}^T(t)]^T, \\
u_k(t) &= [u_{1,k}^T(t), u_{2,k}^T(t), \ldots, u_{N,k}^T(t)]^T, \\
y_k(t) &= [y_{1,k}^T(t), y_{2,k}^T(t), \ldots, y_{N,k}^T(t)]^T, \\
e_k(t) &= [e_{1,k}^T(t), e_{2,k}^T(t), \ldots, e_{N,k}^T(t)]^T, \\
f_k(t, t) &= [f(x_{1,k}^T(t), t), f(x_{2,k}^T(t), t), \ldots, f(x_{N,k}^T(t), t)]^T.
\end{align*}
\]

Thus, (1), (4) and (5) are rewritten into the following form

\[
\begin{align*}
\dot{x}_k(t) &= f(x_k(t), t) + (I \otimes B)u_k(t), \\
y_k(t) &= (I \otimes C)x_k(t), \\
u_{k+1}(t) &= u_k(t) + ((L + S) \otimes \Gamma) \dot{e}_k(t) + \theta_{k,h}(t)X_k(0),
\end{align*}
\]

(7)

\[
X_k(0) = ((L + S) \otimes \Gamma)e_k(0) + (I \otimes B_{R}^{-1})(x_k(0) - x_{k+1}(0)),
\]

(8)

\[
(9)
\]

where \(\otimes\) is a Kronecker product; \(I\) is identity matrix with a corresponding dimension; \(S = \text{diag}(s_1, s_2, \ldots, s_N)\).

The main objective of this paper is to design a distributed consensus iterative learning control algorithm (4) ~ (6) for a class of nonlinear multi-agent systems with random initial states at each iteration, such that the multi-agent output can fully and consistently track the desired trajectory in a finite time interval \(t \in [0, T]\) when the number of iterations is sufficiently large.

**Remark 1:** The introduced virtual leader can only provide the desired trajectory but cannot obtain the information from the agents, however, each agent can obtain the information from the neighbours (including the virtual leader).

**Remark 2:** This article assumes that only some agents can directly obtain the desired trajectory information, that is, only
the agents who have direct communication with the virtual leader can obtain the desired trajectory information in the communication topology.

### III. CONVERGENCE ANALYSIS

For convenience of the proof of the convergence of the proposed algorithm, the following related definitions and lemmas are given.

**Definition 1:** The $\lambda$-norm of a vector function $h(t)$ in a time interval $t \in [0, T]$ is defined as

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

where $\|\cdot\|$ is a norm on $\mathbb{R}^n$, $t \in [0, T]$ is a time variable.

**Lemma 1 [39]**

$$
\sup_{t \in [0, T]} \left( e^{-\lambda t} \int_0^t \|x(\tau)\| \, d\tau \right) \leq \frac{1}{\lambda} \|x\|_\lambda.
$$

The main results of this paper are given as follows:

**Theorem 1:** For the multi-agent system (7) satisfying Assumptions 1~2, when the initial state of each repetition is random, if the learning gain $\Gamma$ satisfies

$$
\|I - (L + S) \otimes (CB^T)\| = \rho < 1.
$$

Then the control algorithms (8) ~ (9) and (6) can cause the multi-agent output $y_i(t)$ to have the following characteristics:

i) In the interval $t \in [h/\beta^k, T]$, when the number of iterations $k \to \infty$, there exists $\lim_{k \to \infty} \|e_k\|_\lambda = 0$. That is, the output $y_k(t)$ is completely consistent in tracking the desired trajectory $y_d(t)$.

ii) In the interval $t \in [0, h/\beta^k]$, due to the influence of random initial state error, the output $y_k(t)$ cannot track the desired trajectory $y_d(t)$, but when $t = h/\beta^k$, the influence of the random initial state error will completely halt. In addition, the interval $[0, h/\beta^k)$ gradually shortens with the increase of the number of iterations, when $k \to \infty$, $h/\beta^k \to 0$, thus making $[h/\beta^k, T] \to (0, T]$, that is, the multi-agent output is completely consistent in tracking the desired trajectory in $t \in (0, T]$.

**Proof:** It is first proven that the control laws (8) ~ (9) and (6) make the tracking error of the system (7) equal zero in $t \in [h/\beta^k, T]$. Then, it is noted that due to the influence of the random initial state error, the time interval $t \in [0, h/\beta^k]$ in which the system output can not track the desired output is shortened with the increase of the number of iterations.

i) When $t \in [h/\beta^k, T]$, from the first expression of (7), we know the state error between the two adjacent iterations is

$$
x_{k+1}(t) - x_k(t) = x_{k+1}(0) + \int_0^t (f(x_{k+1}(\tau), \tau) + (I \otimes B)u_{k+1}(\tau)) \, d\tau - x_k(0) - \int_0^t (f(x_k(\tau), \tau) + (I \otimes B)u_k(\tau)) \, d\tau
$$

$$
= x_{k+1}(0) - x_k(0) + \int_0^t (f(x_{k+1}(\tau), \tau) - f(x_k(\tau), \tau)) \, d\tau + \int_0^t (I \otimes B)(u_{k+1}(\tau) - u_k(\tau)) \, d\tau.
$$

Deriving $(u_{k+1}(\tau) - u_k(\tau))$ from (8), and multiplying $(u_{k+1}(\tau) - u_k(\tau))$ by the left by $(I \otimes B)$, we obtain

$$
(I \otimes B)(u_{k+1}(\tau) - u_k(\tau)) = (I \otimes B)((L + S) \otimes \Gamma)\dot{e}_k(t) + (I \otimes B)\theta_{k,h}(t)X_k(0).
$$

Left multiplication of both sides of (9) by $(I \otimes B)$, we obtain

$$
(I \otimes B)X_k(0) = (I \otimes B)((L + S) \otimes \Gamma)e_k(0) + (I \otimes B)((L + S) \otimes \Gamma)\dot{e}_k(t) + (I \otimes B)\theta_{k,h}(t)X_k(0) + (x_k(0) - x_{k+1}(0)).
$$

From (13), we have

$$
x_{k+1}(0) - x_k(0) = (I \otimes B)((L + S) \otimes \Gamma)e_k(0) - (I \otimes B)X_k(0).
$$

Substituting (14) and (12) into (11), we have

$$
x_{k+1}(t) - x_k(t) = (I \otimes B)((L + S) \otimes \Gamma)e_k(0) - (I \otimes B)X_k(0)
$$

$$
+ \int_0^t (f(x_{k+1}(\tau), \tau) - f(x_k(\tau), \tau)) \, d\tau + (I \otimes B)\int_0^t ((L + S) \otimes \Gamma)\dot{e}_k(t) \, d\tau + (I \otimes B)\int_0^t \theta_{k,h}(\tau) \, d\tau - 1
$$

$$
= (I \otimes B)((L + S) \otimes \Gamma)e_k(0) + \int_0^t (f(x_{k+1}(\tau), \tau) - f(x_k(\tau), \tau)) \, d\tau + (I \otimes B)((L + S) \otimes \Gamma)\dot{e}_k(t) - (I \otimes B)X_k(0)
$$

$$
+ \int_0^t \theta_{k,h}(\tau) \, d\tau - 1
$$

$$
= (I \otimes B)((L + S) \otimes \Gamma)e_k(0) + (I \otimes B)((L + S) \otimes \Gamma)\dot{e}_k(t) + (I \otimes B)X_k(0) + \left(\int_0^t \theta_{k,h}(\tau) \, d\tau - 1\right).
$$

From the definition of tracking error, we know

$$
e_{k+1}(t) - e_k(t) = (y_d(t) - y_{k+1}(t)) - (y_d(t) - y_k(t))
$$

$$
= y_k(t) - y_{k+1}(t)
$$

$$
= (I \otimes C)(x_k(0) - x_{k+1}(0)).
$$
Substituting (15) into (16) yields
\[
e_{k+1}(t) - e_k(t) = (I \otimes C) \times \left( - \int_0^t (f(x_{k+1}(\tau), \tau) - f(x_k(\tau), \tau)) d\tau \right)
\]
\[
- (I \otimes B)(I + L) e_k(t)
\]
\[
- (I \otimes B)X_k(0) \left( \int_0^t \theta_{k,h}(\tau) d\tau - 1 \right)
\]
\[
= -(I \otimes C) \int_0^t (f(x_{k+1}(\tau), \tau) - f(x_k(\tau), \tau)) d\tau
\]
\[
- ((L + S) \otimes (CB^T)) e_k(t)
\]
\[
- (I \otimes CB)X_k(0) \left( \int_0^t \theta_{k,h}(\tau) d\tau - 1 \right).
\]
(17)

From (17), we obtain
\[
e_{k+1}(t) = (I - (L + S) \otimes (CB^T)) e_k(t)
\]
\[
- (I \otimes C) \int_0^t (f(x_{k+1}(\tau), \tau) - f(x_k(\tau), \tau)) d\tau
\]

Therefore, substituting (19) into (18) derives
\[
e_{k+1}(t) = (I - (L + S) \otimes (CB^T)) e_k(t)
\]
\[
- (I \otimes C) \int_0^t (f(x_{k+1}(\tau), \tau) - f(x_k(\tau), \tau)) d\tau.
\]
(20)

Taking the norm on both sides of (20) and combing them with Assumption 1, we have
\[
\|e_{k+1}(t)\| \leq \|(I - (L + S) \otimes (CB^T)) e_k(t)\|
\]
\[
+ \|I \otimes C\| \int_0^t \|f(x_{k+1}(\tau), \tau) - f(x_k(\tau), \tau)\| d\tau
\]
\[
\leq \rho \|e_k(t)\| + b_1 k_f \int_0^t \|\Delta x_{k+1}(\tau)\| d\tau,
\]
(21)

where \(\rho = \|(I - (L + S) \otimes (CB^T))\|, b_1 = \|I \otimes C\|, \Delta x_{k+1}(t) = x_{k+1}(t) - x_k(t)\). Then multiplying both sides of (21) by \(e^{-\lambda t}(\lambda > 0)\), and combing with definition of \(\lambda\)-norm and Lemma 1, we get
\[
\|e_{k+1}\|_\lambda \leq \rho \|e_k\|_\lambda + \frac{b_1 k_f}{\lambda} \|\Delta x_{k+1}\|_\lambda.
\]
(22)

Substituting (19) into (15) gets
\[
\Delta x_{k+1}(t) = x_{k+1}(t) - x_k(t) = ((L + S) \otimes B^T)e_k(t)
\]
\[
+ \int_0^t (f(x_{k+1}(\tau), \tau) - f(x_k(\tau), \tau)) d\tau.
\]
(23)

Taking the norm of both sides of (23) and according to Assumption 1, we obtain
\[
\|\Delta x_{k+1}(t)\| \leq \|(L + S) \otimes B^T\| \|e_k(t)\|
\]
\[
+ \int_0^t \|f(x_{k+1}(\tau), \tau) - f(x_k(\tau), \tau)\| d\tau
\]
\[
\leq k_f \int_0^t \|\Delta x_{k+1}(\tau)\| d\tau + b_2 \|e_k(t)\|,
\]
(24)

where \(b_2 = \|(L + S) \otimes B^T\|\). Similarly, multiplying both sides of (24) by \(e^{-\lambda t}(\lambda > 0)\), and combing with definition of \(\lambda\)-norm and Lemma 1, we obtain
\[
\|\Delta x_{k+1}\|_\lambda \leq b_2 \|e_k\|_\lambda + \frac{k_f}{\lambda} \|\Delta x_{k+1}\|_\lambda.
\]
(25)

When \(\lambda > k_f\), from (25) we get
\[
\|\Delta x_{k+1}\|_\lambda \leq \frac{b_2}{\lambda - k_f} \|e_k\|_\lambda.
\]
(26)

Similarly, substituting (26) into (22) gets
\[
\|e_{k+1}\|_\lambda \leq \left( \rho + \frac{b_1 b_2 k_f}{\lambda - k_f} \right) \|e_k\|_\lambda = \rho_1 \|e_k\|_\lambda,
\]
(27)

where \(\rho_1 = \rho + \frac{b_1 b_2 k_f}{\lambda - k_f}\). From (27) we can know when \(k = 1\), there exists
\[
\|e_2\|_\lambda \leq \rho_1 \|e_1\|_\lambda.
\]
(28)

When \(k = 2\), we have
\[
\|e_3\|_\lambda \leq \rho_1 \|e_2\|_\lambda \leq \rho_1^2 \|e_1\|_\lambda = \rho_1^2 \|e_1\|_\lambda.
\]
(29)

By analogy, we can get
\[
\|e_k\|_\lambda \leq \rho_1 \|e_{k-1}\|_\lambda \leq \rho_1^2 \|e_{k-2}\|_\lambda \leq \cdots \leq \rho_1^{(k-1)} \|e_1\|_\lambda.
\]
(30)

It is known from (30) that the \(\lambda\)-norm of the tracking error at each iteration is less than or equal to the \(\lambda\)-norm of the tracking error at the first iteration. It can be seen that the \(\lambda\)-norm of tracking error at different iterations does not depend on the number of iterations, as long as \(0 < \rho_1 < 1\) is satisfied, \(\lim_{k \to \infty} \|e_k\|_\lambda = \lim_{k \to \infty} \rho_1^{(k-1)} \|e_1\|_\lambda = 0\) holds as \(k \to \infty\). However, from the condition (10) in Theorem 1 we know \(0 < \rho_1 < 1\) holds if \(\lambda\) is sufficient large. Therefore, from (30) we know there exists \(\lim_{k \to \infty} \|e_k\|_\lambda = 0\). That is, the multi-agent system output \(y_h(t)\) can fully and consistently track the desired trajectory \(y_d(t)\) within \(t \in [h/\beta^k, T]\) as \(k \to \infty\).

ii) When \(t \in [0, h/\beta^k]\), the system output cannot track the desired trajectory due to the random initial state error at each iteration. We can analyse how the proposed algorithm guarantees that this time interval will shorten with the increase of iterations. From (6) we know that within \(t \in [0, h/\beta^k]\), there exists
\[
\int_0^t \theta_{k,h}(\tau)d\tau - 1 = - \frac{\beta^k}{h^2} t^2 + \frac{\beta^k}{h} t - 1
\]
\[
= - \left( \frac{\beta^k}{h} t - 1 \right)^2.
\]
(31)

From (31) we know \(\int_0^t \theta_{k,h}(\tau)d\tau - 1 = 0\) when \(t = h/\beta^k\). Therefore, from (18) we know the behaviour that the system
output cannot track the desired output due to the random initial state error, and there is a complete halt at all times when \( t = h/\beta^k \). And \( t = h/\beta^k \rightarrow 0 \), when \( k \rightarrow \infty \). That is, as the number of iterations increases, the time interval \([0, h/\beta^k]\) in which the output cannot track the desired output is shortened.

This completes the proof. \( \square \)

**Remark 3:** It can be seen from (18) and (19) that the effect of the initial state error on the tracking performance is eliminated with the help of the integral action of \( \theta_{k, h}(t) \). If \( \theta_{k, h}(t) \) is replaced with the pulse function \( \delta(t) \) is a function independent of the number of iterations), the influence of the initial state error can be instantly eliminated at the beginning of each iteration. This can not be achieved in the actual system when the output of the controller tends to infinity. However, \( \theta_{k, h}(t) \) in (4) approximates the instantaneous integration of the pulse function step-by-step with the increase of the number of iterations. Although \( \theta_{k, h}(t) \) will gradually increase with the increase of the number of iterations, and thus the output of the controller will be increased step-by-step. However, in practical engineering terms, the tracking error can meet the requirements, if the tracking error reaches a certain range of accuracy rather than requiring that the tracking error is zero: that is, the number of iterations is limited, not infinite. Therefore, from the point of view of practical engineering, the output of the controller designed in this paper will not reach infinity.

### IV. SIMULATION ANALYSIS

To testify the effectiveness of the proposed algorithm, consider a class of nonlinear multi-agent systems as follows: the system is composed of four same-structure agents and one virtual leader, its topology is shown in Fig.1. The system repetitively runs in a finite time interval, and the initial state is random at each iteration. The model of the \( j \)-th agent of the system \((j \in \{1, 2, 3, 4\})\) is

\[
\begin{align*}
\dot{x}_{j1,k}(t) &= -x_{j1,k}(t) + (x_{j2,k}(t))^2 + u_{j1,k}(t) \\
\dot{x}_{j2,k}(t) &= \cos(x_{j1,k}(t)x_{j2,k}(t)) + u_{j2,k}(t) \\
y_{j,k}(t) &= x_{j1,k}(t).
\end{align*}
\]

In Fig. 1, node 0 represents the virtual leader, which provides a given desired trajectory. From Fig. 1, we know only some of the agents (agent 1 and agent 3) can directly obtain the desired trajectory information, \( S = \text{diag}(2, 0, 2, 0) \), and Laplacian matrix is

\[
L = \begin{bmatrix}
1 & -1 & 0 & 0 \\
-1 & 1.6 & 0 & -0.6 \\
0 & 0 & 1 & -1 \\
0 & -0.6 & -1 & 1.6
\end{bmatrix}.
\]

The multi-agent system composed of (29) and Fig. 1, it is controlled by using algorithms (8)~(9) and (6). Assume that the repetitive run time is \( t \in [0, 1] \), and the desired output is \( y_d(t) = \sin(2\pi t) \). At each iteration, the initial state \( x_{j,0}(0) \) of each agent is randomly generated by the function \( \text{rand}(\cdot) \), i.e., \( x_{j,k}(0) = [\text{rand}(1), \text{rand}(1)]^T \). Here, \( \text{rand}(\cdot) \) is a random number between 0 and 1. The initial control is selected as \( u_{j,0}(t) = [0, 0]^T, \beta = 1.03, h = 0.1 \); the learning gain is \( \Gamma = [0.25, 0]^T \). At this time \( \rho = 0.5 < 1 \) satisfies the convergence condition (10). Its simulation results are shown as Figs. 2~7. Figs. 2~5 are, respectively, the output tracking curves for desired trajectory at the 10th, 20th, 30th, and 50th iterations. Fig. 6 is the consensus tracking curve of all the agents at the 60th iteration. Fig. 7 shows the maximum tracking errors of the four agents’ outputs at each iteration in the time interval \( t \in [h/\beta^k, T] \).

From Figs. 2~6 we know that under a random initial state condition, the output of each agent can achieve full consensus tracking in the time interval \( t \in [h/\beta^k, T] \) at each iteration, and the time interval in which the desired trajectory cannot be tracked can shorten with the increase of iteration. Additionally, Fig. 7 shows that the tracking error gradually tends to zero within the time interval \( [h/\beta^k, T] \), as the number of iterations increases.
The reason is that this algorithm designs a time interval $[0, t_1)$, in which the random initial state error is rectified, where $t_1 = h/\beta^k$ gradually decreases with the increase of the number of iterations. Therefore, this makes the interval $t \in [t_1, T]$ of the zero error tracking of the desired trajectory gradually widen with the increase of the iterations, to make the interval $[0, t_1)$ unable to track the desired trajectory and to gradually shorten, ultimately achieving the complete tracking of the desired trajectory over $(0, T]$ as $k \to \infty$. However, when $t_1 = h$ is a fixed value, the zero error tracking is achieved only over the fixed time interval $[t_1, T]$, but the time interval $[0, t_1)$ is unable to track the desired trajectory and cannot gradually shorten with the increase of iterations. The simulation result at the 60th iteration is shown as Fig. 8.

To see more clearly the influence of the decreasing rectifying interval on the control input as the number of iterations increases, we provide Figs. 9-13. Where Figs. 9∼10 show the variation curves of the maximum value of the control inputs with the increase of the number of iterations; Figs. 11∼12 show the time-dependent curves of the control inputs at the 50th iteration; Fig. 13 shows the variation curve of the tracking error with time at the 50th iteration.

It can be seen from Figs. 9∼10 that the maximum value of the control input gradually increases with the increase of the number of iterations, but it can be seen from Figs. 11∼12 that the maximum value of the control input only appears in the rectifying interval, while after the rectifying interval, the control input is basically close to 0. In addition, it can be seen from Fig. 13 that after 50 iterations, the rectifying interval of the initial state error is shortened from $[0, 0.1]$ to $[0, 0.02]$ at the beginning of the iteration, which shows that the influence interval of the initial state error on the tracking accuracy is shortened. Additionally, it can also be seen from Fig. 13 that the rectifying interval...
tracking error is close to 0 in the interval $[0.02, 1]$ s. This result shows that although the rectifying interval shortens with the increase of the number of iterations, the control input gradually increases in the rectifying interval. However, in practical engineering terms, the tracking error can meet the requirements, if the tracking error reaches a certain accuracy, rather than requiring the tracking error to equal 0. In other words, the tracking error can meet the need of the project after a limited number of iterations. Therefore, from the point of view of practical engineering, the control input will not infinitely increase in the rectifying interval.

At the same time, from the algorithms (4)∼(6) we can know when $h$, $L$, $S$, $\Gamma$ are fixed, the tracking precision in $[h \beta^k, T]$ will depend on the number of iterations $k$ and the precision increases gradually with the increase of $k$; the length of the rectifying interval $[0, h \beta^k]$ depends on the parameters $\beta$ and $k$. When the number of iterations $k$ is the same, the larger $\beta$ is, the shorter the rectifying interval is and the larger the control input at the initial rectifying time is; the smaller $\beta$ is, the longer the rectifying interval is and the smaller the control input at the initial rectifying time is.
In order to illustrate the influence of $\beta$ on the rectifying interval and the initial control input, the following simulation is analyzed when $\beta$ is different and other parameters are the same. When $\beta = 1.03$, the control input at the 50th iteration is shown in Figs. 11-12, and their tracking trajectories of the four agents are shown in Fig. 16; When $\beta = 1.003$, the control input at the 50th iteration is shown in Figs. 14-15, and their tracking trajectories of the four agents are shown in Fig. 17. By comparing Fig. 11 with Fig.14, and Fig.12 with Fig. 15, we can see at the same number of iterations, the control input of the initial time decreases obviously with the decrease of $\beta$. From Figs. 16 and 17, we can see the tracking precision in $[h/\beta^k, T]$ remains the same, but the rectifying interval is widened appropriately.

Therefore, in practical application, on the premise that the tracking precision meets the requirements (keeping a certain number of iterations unchanged), the rectifying interval is appropriately relaxed (a smaller $\beta$ is selected) to reduce the control input at the initial rectifying time. It can also be considered as a compromise according to the requirements of tracking precision and rectifying interval.

V. CONCLUSION

For a class of nonlinear multi-agent systems, a distributed iterative learning control protocol is proposed to solve consensus tracking control problems with the random initial states. To overcome the influence of the random initial state error on tracking performance, a time interval is designed in the control protocol, so that the influence of the random initial state error only affects this time interval, and the time interval is shortened as the number of iterations increases. The validity of the proposed algorithm is theoretically proved, and the convergence conditions are given. Theoretical analysis and simulation results show that the distributed control protocol proposed in this paper can make the output of the multi-agent system with a random initial state completely track the desired trajectory within a finite time interval $[0, T]$ after a sufficient number of iterations. Thus this effectively suppresses the influence of the random initial error on the tracking performance, and relaxes the restrictions of the consensus iterative learning control on the initial positioning conditions.

REFERENCES

[1] F. Zhou and Z. Wang, “Containment control of linear multi-agent systems with directed graphs and multiple leaders of time-varying bounded inputs,” IET Control Theory Appl., vol. 9, no. 16, pp. 2466–2473, Oct. 2015.
[2] W. Ren, R. W. Beard, and E. M. Atkins, “Information consensus in multivehicle cooperative control,” IEEE Control Syst. Mag., vol. 27, no. 2, pp. 71–82, Apr. 2007.
[3] M. S. Stankovic, K. H. Johansson, and D. M. Stipanovic, “Distributed seeking of Nash equilibria with applications to mobile sensor networks,” IEEE Trans. Autom. Control, vol. 57, no. 4, pp. 904–919, Apr. 2012.
[4] D. Wang, Q. Zong, B. Tian, F. Wang, and L. Dou, “Finite-time fully distributed formation reconfiguration control for UAV helicopters,” Int. J. Robust Nonlinear Control, vol. 28, no. 18, pp. 5943–5961, Dec. 2018.
[5] Y. Liu, J. M. Montenbruck, D. Zelazo, M. Odelga, S. Rajappa, H. H. Bulthoff, F. Allgower, and A. Zell, “A distributed control approach to formation balancing and maneuvering of multiple multicopter UAVs,” IEEE Trans. Robot., vol. 34, no. 4, pp. 870–882, Aug. 2018.
[6] K. Wang, E. Tian, J. Liu, L. Wei, and D. Yue, “Resilient control of networked control systems under deception attacks: A memory-event-triggered communication scheme,” Int. J. Robust Nonlinear Control, vol. 30, no. 4, pp. 1534–1548, Mar. 2020.
[7] E. Tian and C. Peng, “Memory-based event-triggering H-infinity load frequency control for power systems under deception attacks,” IEEE Trans. Cybern., vol. 50, no. 11, pp. 4610–4618, Nov. 2020.
[8] T. Liu and Z.-P. Jiang, “Distributed nonlinear control of mobile multi-robot formation control in dynamic environments,” Auton. Robots, vol. 43, no. 5, pp. 1075–1086, Apr. 2014.
[9] J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, and D. Rus, “Distributed multi-robot formation control in dynamic environments,” Auton. Robots, vol. 43, no. 5, pp. 1079–1100, Jun. 2019.
[10] D. Meng and Y. Jia, “Formation control for multi-agent systems through an iterative learning design approach,” *Int. J. Robust Nonlinear Control*, vol. 24, no. 2, pp. 340–361, Jan. 2014.

[11] Y. Cao, L. Zhang, C. Li, and M. Z. Q. Chen, “Observer-based consensus tracking of nonlinear agents in hybrid varying directed topology,” *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 2212–2222, Aug. 2017.

[12] X. Ge and Q.-L. Han, “Consensus of multiagent systems subject to partially accessible and overlapping Markovian network topologies,” *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 1807–1819, Aug. 2017.

[13] L. Cheng, Z.-G. Hou, and M. Tan, “A mean square consensus protocol for linear multi-agent systems with communication noises and fixed topologies,” *IEEE Trans. Autom. Control*, vol. 59, no. 1, pp. 261–267, Jan. 2014.

[14] T. Li and J.-F. Zhang, “Consensus conditions of multi-agent systems with time-varying topologies and stochastic communication noises,” *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2043–2057, Sep. 2010.

[15] Y. Zheng, Q. Zhao, J. Ma, and L. Wang, “Second-order consensus of hybrid multi-agent systems,” *Syst. Control Lett.*, vol. 125, pp. 51–58, Mar. 2019.

[16] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operation of robots by learning,” *J. Robotic Syst.*, vol. 1, no. 2, pp. 123–140, 1984.

[17] Q. Lv, Y. C. Fang, and X. Ren, “Iterative learning control for accelerated inhibition effect of initial state random error,” *Acta Autom. Sinica*, vol. 40, no. 7, pp. 1295–1302, Jul. 2014.

[18] Q. Chen, H. Shi, and M. Sun, “Echo state network-based backstepping adaptive iterative learning control for strict-feedback systems: An error-tracking approach,” *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3009–3022, Jul. 2020.

[19] L. Blanken and T. Oomen, “Multivariable iterative learning control design procedures: From decentralized to centralized, illustrated on an industrial printer,” *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 4, pp. 1534–1541, Jul. 2020.

[20] Y. Liu, W. Zhan, H. Gao, and H. Liu, “Vibration suppression of an Euler–Bernoulli beam by backstepping iterative learning control,” *IET Control Theory Appl.*, vol. 13, no. 16, pp. 2630–2637, Nov. 2019.

[21] Y. Zhang, Q. Yan, J. Cai, and X. Wu, “Adaptive iterative learning control for tank gun servo systems with input deadzone,” *IEEE Access*, vol. 8, pp. 63443–63451, 2020.

[22] X. Bu, Z. Fu, F. Yu, and Z. Hou, “Iterative learning control for a class of non-linear switched systems,” *IET Control Theory Appl.*, vol. 7, no. 3, pp. 470–481, Feb. 2013.

[23] P. Jiang, H. Chen, and L. C. A. Bamforth, “A universal iterative learning stabilizer for a class of MIMO systems,” *Automatica*, vol. 42, no. 6, pp. 973–981, Jun. 2006.

[24] S. Yang, J.-X. Xu, D. Huang, and Y. Tan, “Synchronization of heterogeneous multi-agent systems by adaptive iterative learning control,” *Asian J. Control*, vol. 17, no. 6, pp. 2091–2104, Nov. 2015.

[25] P. Gu, S. Tian, and Q. Liu, “Closed-loop iterative learning control for discrete singular systems with fixed initial shift,” *J. Syst. Sci. Complex.*, vol. 32, no. 2, pp. 577–587, Apr. 2019.

[26] Q. Fu, P. Gu, and J. Wu, “Decentralized iterative learning control for large-scale linear discrete-time systems with fixed initial state errors,” *IMA J. Math. Control Inf.*, vol. 35, no. 2, pp. 591–610, Jun. 2018.

[27] M. Sun and D. Wang, “Closed-loop iterative learning control for non-linear systems with initial shifts,” *Int. J. Adapt. Control Signal Process.*, vol. 16, no. 7, pp. 515–538, Sep. 2002.

[28] R. Chi, D. Zhang, X. Liu, Z. Hou, and S. Jin, “An anticipatory terminal iterative learning control approach with applications to constrained batch processes,” *Chin. J. Chem. Eng.*, vol. 21, no. 3, pp. 271–275, Mar. 2013.

[29] M. Sun and D. Wang, “Iterative learning control with initial rectifying action,” *Automatica*, vol. 38, no. 7, pp. 1177–1182, Jul. 2002.

[30] R. Chi, Z. Hou, and J. Xu, “Adaptive ILC for a class of discrete-time systems with iteration-varying trajectory and random initial condition,” *Automatica*, vol. 44, no. 8, pp. 2207–2213, Aug. 2008.

[31] Q. Yang, Q. Yan, J. Cai, J. Tian, and X. Guan, “Neural network-based error-tracking iterative learning control for tank gun control systems with arbitrary initial states,” *IEEE Access*, vol. 8, pp. 72179–72187, 2020.

[32] Q. Fu, X.-D. Li, L.-L. Du, G.-Z. Xu, and J.-R. Wu, “Consensus control for multi-agent systems with quasi-one-sided lipschitz nonlinear dynamics via iterative learning algorithm,” *Nonlinear Dyn.*, vol. 91, no. 4, pp. 2621–2630, Mar. 2018.

[33] D. Meng, Y. Jia, and J. Du, “Consensus seeking via iterative learning for multi-agent systems with switching topologies and communication time-delays: Consensus seeking via iterative learning for multi-agent systems with switching topologies and communication time-delays,” *Int. J. Robust Nonlinear Control*, vol. 26, no. 17, pp. 3772–3790, Nov. 2016.

[34] X. Deng, X. Sun, and S. Liu, “Iterative learning control for leader-following consensus of nonlinear multi-agent systems with packet dropout,” *Int. J. Control, Autom. Syst.*, vol. 17, no. 8, pp. 2135–2144, Aug. 2019.

[35] S. Yang, J.-X. Xu, and X. Li, “Iterative learning control with input sharing for multi-agent systems,” *Syst. Control Lett.*, vol. 94, pp. 97–106, Aug. 2016.

[36] J. Q. Liang, X. H. Bu, J. Liu, and W. Qian, “Iterative learning consensus tracking control for a class of multi-agent systems with output saturation,” *Control Theory Appl.*, vol. 35, no. 6, pp. 786–794, Jun. 2018.

[37] J. S. Li and J. M. Li, “Consensus seeking in multi-agent systems by the iterative learning control,” *Control Theory Appl.*, vol. 29, no. 8, pp. 1037–1077, Oct. 2012.

[38] Q. Z. Yan, M. X. Sun, and H. Li, “Consensus-error-tracking learning control for nonparametric uncertain multi-agent systems,” *Control Theory Appl.*, vol. 33, no. 6, pp. 793–799, 2016.

[39] M. X. Sun and B. J. Huang, *Iterative Learning Control*. Beijing, China: National Defense Industry Press, 1999, pp. 283–286.