



Consensus Control of Multi-agent Robot System with State Delay Based on Fractional-Order Iterative Learning Control Algorithm

Yanghua Gao^{1*}, Weidong Lou¹, Hailiang Lu¹, Yonghua Jia²

¹ Information Center, China Tobacco Zhejiang Industrial Co., Ltd., Hangzhou 310008, China

² Hangzhou Hikrobot Technology Co., Ltd., Hangzhou 310052, China

Corresponding Author Email: yhgao@zju.edu.cn

<https://doi.org/10.18280/jesa.530603>

ABSTRACT

Received: 12 June 2020

Accepted: 25 September 2020

Keywords:

multi-agent robot system, fractional-order iterative learning control (FOILC), state delay, consensus control

This paper mainly explores the consensus control of multi-agent robot system with repetitive motion under the constraints of a leader and fixed topology. To realize the consensus control, a fractional order iterative learning control (FOILC) algorithm was designed under the mode of distributed open-closed-loop proportional-derivative alpha (PD^α). The uniform convergence of the algorithm in finite time was discussed, drawing on fractional calculus, graph theory, and norm theory, resulting in the convergence conditions. Theoretical analysis shows that, with the growing number of iterations, each agent can choose the appropriate gain matrix, and complete the tracking task in finite time. The effectiveness of the proposed method was verified through simulation.

1. INTRODUCTION

The multi-agent robot system can complete large and complex fieldwork through communication, coordination, and cooperation. The efficiency and performance of the system are much better than those of a single robot. As a result, the multi-agent robot system has been widely used in mobile robot handling, three-dimensional (3D) printing, and welding [1, 2].

The consensus control, as a representative problem of the collaborative control of the multi-agent robot system, aims to design a suitable control algorithm to regulate the state or output of each robot in a finite time. The coordination and consensus between multiple agents have attracted much attention from the academia, yielding fruitful results [3-6]. However, most consensus control methods for multiple agents are based on integer-order descriptions of the agents, that is, the differential equations of the robots are of integer-order [7, 8].

In complex natural environments (e.g. material mechanics, motor systems, and robots), it is more accurate to describe robot dynamics with fractional calculus than integer calculus [9, 10]. Strictly speaking, the integer-order system is a special case of the fractional-order system. Therefore, many scholars engaging in multi-agent robot system have turned their attention from the integer-order model to the fractional-order model [11-14].

Yu et al. [11, 12] pioneered the study of consensus control of fractional-order multi-agent system (FOMAS), and judged the convergence of fractional multiple agents with a leader. Later, Bai et al. [13] explored the consensus control of FOMAS in fixed reference states and time-varying reference states. Ma et al. [14] introduced consensus control to fractional dynamics, and developed a consensus control algorithm based on relative output measurements. By graph theory and Lyapunov method, Yu et al. [11, 12] defined the criteria for

the consensus control of nonlinear FOMAS with a leader, put forward the stability theory on fractional differential systems, and derived the matrix inequalities based on the theory. Yu et al. [15] presented the necessary and sufficient conditions for the consensus control of FOMAS with leaders, and designed an observer to solve the consensus problem. Chai et al. [16] and Sun et al. [17] realized consensus control of FOMAS with adaptive control, and sliding mode control, respectively.

Most of the above studies focus on the consensus control with gradual convergence. However, the FOMAS with repetitive motions, such as the coordinated operation of multiple manipulators on a production line, generally require the control algorithm to converge within a finite time. The methods of the above research cannot satisfy this requirement.

Among the current control methods, iterative learning control can complete the tracking task in a finite time. This method has been applied to various models of integer-order multi-agent system (IOMAS) [18-20]. But there is little report on the fractional-order iterative learning control (FOILC) of FOMAS. To make matters worse, the actual FOMAS often face state delays induced by digital signal processors (DSPs), communication processors, and other processors. Hence, it is of engineering significance to solve the consensus control of FOMAS with state delay.

Considering the state delay in FOMAS, this paper proposes several methods to solve the consensus control of FOMAS composed of robots, namely, FOLIC algorithms under the mode of open-closed-loop proportional-derivative alpha (PD^α), closed-loop PD^α , and, open-loop PD^α . Based on graph theory, norm theory, and fractional calculus, the finite-time convergence of the proposed algorithms was theoretically proved for FOMAS with state delay. Finally, the open-closed-loop PD^α FOLIC algorithm was verified to be the most effective consensus control method for FOMAS with state delay through numerical simulation.

2. PRELIMINARIES

2.1 Graph theory

For a multi-agent system of n fractional agents, the communication links between the multiple agents constitute a network. The network can be expressed as a directed weighted graph $G=\{V, E, M\}$, where $V=\{v_1, \dots, v_n\}$ is the set of nodes (agents), and $E \in V \times V$ is the set of edges (communication links). Let $I=\{1, 2, \dots, n\}$ be the index set of the nodes. Then, the adjacency matrix of the network can be described as $M=(a_{ik})_{n \times n}$, where a_{ik} is the weight of the edge between node i and node k . If node i transmits information to node k , then $a_{ik}>0$; otherwise $a_{ik}=0$. It was also assumed that no node in the network commits self-connection: $a_{ii}=0$ for $i \in I$.

Let $N_i=\{k: a_{ik}>0\}$ be the set of neighboring nodes of node i , and $D=diag\{d_i, i=1, \dots, N\}$, where $d_i = \sum_{k=1}^N a_{ik}$ is the sum of the elements in row i of adjacency matrix M . Hence, the Laplacian matrix of graph G can be obtained as $L=D-M$. For two nodes i and k , if a set of subscripts $\{k_1, \dots, k_l\}$ satisfies $a_{ik_1} > 0, a_{ik_1} > 0, \dots, a_{k_l k} > 0$, then there exists an information transmission channel between the two nodes, such that node k can receive the information from node i . Node i is globally reachable if there is a path for it to reach any other node in the graph.

Lemma 1. For an FOMAS of n autonomous individuals, if the edges form a directed weighted network, and if there is at least one globally reachable node, then the rank of the Laplacian matrix L of the directed weighted graph is $N-1$, with a zero eigenvalue corresponding to the eigenvector $\xi_0=c[1, \dots, 1]^T$.

2.2 Fractional calculus and norm theory

The fractional calculus adopted for this research can be defined as follows:

Definition 1. In the interval $[t_0, t]$, the α -order fractional derivatives of function $f(t)$ can be defined as:

$${}_{t_0} D_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^t (t-\tau)^{\alpha-1} f(\tau) d\tau \quad (1)$$

where, $\alpha>0$; $\Gamma(\cdot)$ is gamma function $\Gamma(\alpha) = \int_0^\infty e^{-t} t^{\alpha-1} dt$.

In the interval $[t_0, t]$, the α -order fractional Riemann-Liouville and Caputo derivatives on the function $f(t)$ can be expressed as:

$${}_{t_0}^{RL} D_t^\alpha f(t) = \frac{d^{[\alpha]+1}}{dt^{[\alpha]+1}} \left[{}_{t_0} D_t^{-(\alpha-[\alpha]+1)} f(t) \right] \quad (2)$$

$${}_{t_0}^C D_t^\alpha f(t) = {}_{t_0} D_t^{-(\alpha-[\alpha]+1)} \left[\frac{d^{[\alpha]+1}}{dt^{[\alpha]+1}} f(t) \right] \quad (3)$$

where, α is any positive real number; $[\alpha]$ is the integer part of α .

Lemma 2. For the continuous function $f(x(t), t)$, there exist:

$$\begin{cases} {}_{t_0}^C D_t^\alpha x(t) = f(x(t), t), 0 < \alpha < 1 \\ x(t_0) = x(0) \end{cases} \quad (4)$$

Then, the equivalent Volterra nonlinear integral equation for the initial value problem can be expressed as:

$$x(t) = x(0) + \frac{1}{\Gamma(\alpha)} \int_{t_0}^t (t-\tau)^{\alpha-1} f(x(\tau), \tau) d\tau \quad (5)$$

Let \mathbb{R} be any set of real numbers, \mathbb{C} be any set of complex numbers, and \mathbb{N} be the set of natural numbers. For a vector $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$, the l_p norm can be represented as $\|x\|_p$, with $1 \leq p \leq \infty$. In fact, when p equals 1, 2, and ∞ , there exist $\|x\|_1 = \sum_{k=1}^n |x_k|$, $\|x\|_2 = \sqrt{x^T x}$, and $\|x\|_\infty = \max_{k=1, \dots, n} |x_k|$. Then, matrix $A \in \mathbb{R}^{n \times n}$ $\|A\|$ represents the matrix norm, and $\rho(A)$ characterizes the spectral radius of matrix A . In addition, the Kronecker product is denoted as \otimes , and the $m \times m$ identity matrix as I_m .

2.3 FOMAS model with state delay

Suppose the FOMAS with leader contains N agents, numbered as 1, 2, 3..., N . Each agent can be modeled as:

$$\begin{cases} D^\alpha x_{i,j}(t) = Ax_{i,j}(t-\tau) + Bu_{i,j}(t) \\ y_{i,j}(t) = Cx_{i,j}(t) \end{cases} \quad (6)$$

where, $t \in [0, T]$ (if $t \in [-h, 0]$, then $x_{i,j}(t) = \psi(t)$; $\alpha \in (0, 1)$; i is the number of iterations; $D^\alpha x_{i,j}(t)$ is the α -order derivative of $x_{i,j}(t)$; $x_{i,j}(t) \in \mathbb{R}^m$ is the vector of the j -th agent; $u_{i,j}(t) \in \mathbb{R}^{m_1}$ and $y_{i,j}(t) \in \mathbb{R}^{m_2}$ are the input and output, respectively; A, B and C are constant matrices.

Formula (6) can be written in the multi-agent vector form as

$$\begin{cases} D^\alpha x_i(t) = (I_N \otimes A) x_i(t-\tau) + (I_N \otimes B) u_i(t) \\ y_i(t) = (I_N \otimes C) x_i(t) \end{cases} \quad (7)$$

where, $x_i(t) = [x_{i,1}(t)^T, x_{i,2}(t)^T, \dots, x_{i,N}(t)^T]^T$; $u_i(t) = [u_{i,1}^T(t), u_{i,2}^T(t), \dots, u_{i,N}^T(t)]^T$; $y_i(t) = [y_{i,1}^T(t), y_{i,2}^T(t), \dots, y_{i,N}^T(t)]^T$; $\xi_i(t) = [\xi_{i,1}(t)^T, \xi_{i,2}(t)^T, \dots, \xi_{i,N}(t)^T]^T$; \otimes is the Kronecker product. Here, the desired trajectory $y_d(t)$ is generated by the leader in formula (6) within the time interval $[0, T]$. Then, the leader can be defined as:

$$\begin{cases} D^\alpha x_d(t) = Ax_d(t-\tau) + Bu_d(t) \\ y_d(t) = Cx_d(t) \end{cases} \quad (8)$$

where, $u_d(t)$ is the desired input.

Due to the limitations on communication or sensors, the leader can only communicate with some of the multiple agents. Let $G\{V, E\}$ be the communication network, and zero be the serial number of the leader. Then, the entire graph containing the leader can be expressed as $\bar{G} = \{0 \cup V, \bar{E}\}$, where \bar{E} is the corresponding edge in the new graph. Our main task is to design an FOILC algorithm to ensure that each agent converges to the desired trajectory in the new graph \bar{G} .

The distributed error $\zeta_{i,j}(t)$ can be defined as:

$$\begin{aligned} \xi_{i,j}(t) &= \sum_{k \in N_j} a_{j,k} (y_{i,k}(t) - y_{i,j}(t)) \\ &+ d_k (y_d(t) - y_{i,j}(t)) \end{aligned} \quad (9)$$

where, $a_{j,k}$ is the weight of the (j, k) -th neighborhood of the adjacency matrix M ; N_j is the domain set of the j -th agent; d_k is the sum of the elements in row i of the adjacency matrix. If $d_j=1$, then the j -th agent can receive information from the leader, i.e., $(0, j) \in \bar{E}$; if $d_j=0$, then the j -th agent cannot receive information from the leader. The error can be expressed as $e_{i,j}(t)=y_d(t)-y_{i,j}(t)$.

Then, the open-closed PD^α FOILC can be designed as:

$$\begin{aligned} \mathbf{u}_{i+1,j}(t) = & \mathbf{u}_{i,j}(t) + \Gamma_{p1} \boldsymbol{\xi}_{i,j}(t) + \Gamma_{D1} \boldsymbol{\xi}_{i,j}^{(\alpha)}(t) \\ & + \Gamma_{p2} \boldsymbol{\xi}_{i+1,j}(t) + \Gamma_{D2} \boldsymbol{\xi}_{i+1,j}^{(\alpha)}(t) \end{aligned} \quad (10)$$

From formula (9) and the error expression, the distributed error can be further defined as:

$$\boldsymbol{\xi}_{i,j}(t) = \sum_{k \in N_j} a_{j,k} (e_{i,j}(t) - e_{i,k}(t)) + d_k e_{i,j}(t) \quad (11)$$

Let $\mathbf{x}_i(t) = [x_{i,1}(t)^T, x_{i,2}(t)^T, \dots, x_{i,N}(t)^T]^T$, $\mathbf{e}_i(t) = [e_{i,1}(t)^T, e_{i,2}(t)^T, \dots, e_{i,N}(t)^T]^T$, and $\boldsymbol{\xi}_i(t) = [\xi_{i,1}(t)^T, \xi_{i,2}(t)^T, \dots, \xi_{i,N}(t)^T]^T$ be the column vectors. Then, formula (11) can be rewritten as:

$$\boldsymbol{\xi}_i(t) = ((L+D) \otimes \mathbf{I}_m) \mathbf{e}_i(t) \quad (12)$$

where, L is the Laplacian matrix of graph \bar{G} ; $D = \text{diag}\{d_i, i=1, \dots, N\}$.

From formula (12), the controller in formula (10) can be written as a matrix vector:

$$\begin{aligned} \mathbf{u}_{i+1}(t) = & \mathbf{u}_i(t) + ((L+D) \otimes \Gamma_{p1}) \mathbf{e}_i(t) \\ & + ((L+D) \otimes \Gamma_{D1}) \mathbf{e}_i^{(\alpha)}(t) \\ & + ((L+D) \otimes \Gamma_{p2}) \mathbf{e}_{i+1}(t) \\ & + ((L+D) \otimes \Gamma_{D2}) \mathbf{e}_{i+1}^{(\alpha)}(t) \end{aligned} \quad (13)$$

For convenience, the root locus of matrix $L+D$ can be defined as $\lambda_j, j=1, 2, \dots, N$.

3. CONVERGENCE ANALYSIS

Several assumptions were put forward to facilitate the convergence analysis.

Assumption 1. During the repeated runs on the interval $[0, T]$, the initial state of each agent (6) is the desired initial state. That is, for all k , $x_{i,k}(0) = x_d(0)$ holds.

Remark 1. To ensure tracking performance, the initial state in Assumption 2 must be established in the design of iterative learning control. However, if the assumption does not hold, it is impossible to achieve desired tracking without an optimal initial condition. The iterative learning control of the initial state is detailed by Shiping [21].

Assumption 2. The graph \bar{G} contains a spanning tree with the leader being the root.

Remark 2. Assumption is a prerequisite for the consensus control of FOMAS, in which all followers can reach the leader. Otherwise, the isolated agents cannot track the leader's trajectory, for the control inputs are inaccurate due to the absence of data.

Theorem. Assumptions 1 and 2 hold for the FOMAS agent (7). Under the conditions of graph \bar{G} and controller (13), if the gain matrices Γ_{p1} , Γ_{p2} , Γ_{D1} , and Γ_{D2} satisfy:

$$\begin{aligned} (1) & \Gamma(\alpha) - \|(\cdot)^{\alpha-1}(\mathbf{I}_n \otimes A)\|_1 > 0; \\ (2) & 0 < \rho_1 < 1, 0 < \rho_2 < 1, \frac{\rho_1}{\rho_2} < 1; \end{aligned}$$

where, $\rho_1 = |I - ((L+D) \otimes \Gamma_{D1})(\mathbf{I}_n \otimes C)(\mathbf{I}_n \otimes B)| + \beta_1$, $\rho_2 = \|I + ((L+D) \otimes \Gamma_{D2})(\mathbf{I}_n \otimes C)(\mathbf{I}_n \otimes B)\| - \beta_2$, $\beta_i = \frac{\|((L+D) \otimes \Gamma_{pi})(\mathbf{I}_n \otimes C)\| + \|((L+D) \otimes \Gamma_{Di})(\mathbf{I}_n \otimes C)(\mathbf{I}_n \otimes A)\|}{\Gamma(\alpha) - \|(\cdot)^{\alpha-1}(\mathbf{I}_n \otimes A)\|_1} \|(\cdot)^{\alpha-1}(\mathbf{I}_n \otimes B)\|_1, i = 1, 2$; I is the unit matrix; ρ is a constant; $H=L+D$ (L is the Laplacian matrix; $D = \text{diag}\{d_i, i = 1, \dots, N\}$), then, with the growing number of iterations, the control $u_{i,k}(t)$ and output $y_{i,k}(t)$ of each agent will converge to the desired values $u_d(t)$ and $y_d(t)$.

Proof. Suppose

$$\begin{cases} \delta \mathbf{x}_{i,j}(t) = \mathbf{x}_d(t) - \mathbf{x}_{i,j}(t) \\ \delta \mathbf{u}_{i,j}(t) = \mathbf{u}_d(t) - \mathbf{u}_{i,j}(t) \end{cases} \quad (14)$$

where, $\delta x_i(t)$ and $\delta u_i(t)$ are the vector form of $\delta x_{i,j}(t)$ and $\delta u_{i,j}(t)$, respectively. If $t \in [-\tau, 0]$ ($\tau > 0$), then:

$$\delta \mathbf{x}_i(t) = 0 \quad (15)$$

Hence,

$$\begin{aligned} \|\delta \mathbf{x}_i(t - \tau)\|_p &= \left[\int_0^t (\max_{1 \leq j \leq n} |\delta x_{i,j}(s - \tau)|)^p ds \right]^{1/p} \\ &= \left[\int_{-\tau}^{t-\tau} (\max_{1 \leq j \leq n} |\delta x_{i,j}(s)|)^p ds \right]^{1/p} \\ &= \left[\int_0^{t-\tau} (\max_{1 \leq j \leq n} |\delta x_{i,j}(s)|)^p ds \right]^{1/p} \\ &\leq \left[\int_0^t (\max_{1 \leq j \leq n} |\delta x_{i,j}(s)|)^p ds \right]^{1/p} \\ &= \|\delta \mathbf{x}_i(t)\|_p \end{aligned} \quad (16)$$

Following the definition of error and Assumption 1:

$$\begin{aligned} \mathbf{e}_i^{(\alpha)}(t) &= \mathbf{y}_d^{(\alpha)}(t) - \mathbf{y}_i^{(\alpha)}(t) \\ &= (\mathbf{I}_n \otimes C)(\mathbf{x}_d^{(\alpha)}(t) - \mathbf{x}_i^{(\alpha)}(t)) \\ &= (\mathbf{I}_n \otimes C) \delta \mathbf{x}_i^{(\alpha)}(t) \\ &= (\mathbf{I}_n \otimes C)(\mathbf{I}_n \otimes A) \delta \mathbf{x}_i(t - \tau) \\ &+ (\mathbf{I}_n \otimes C)(\mathbf{I}_n \otimes B) \delta \mathbf{u}_i(t) \end{aligned} \quad (17)$$

From Lemma 1, we have:

$$\begin{aligned} \mathbf{x}_i(t) &= \mathbf{x}_i(0) + \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} ((\mathbf{I}_n \otimes A) \mathbf{x}_i(s - \tau) \\ &+ (\mathbf{I}_n \otimes B) \mathbf{u}_i(s)) ds \\ &= \mathbf{x}_i(0) + \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} (\mathbf{I}_n \otimes A) \mathbf{x}_i(s - \tau) ds \\ &+ \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} (\mathbf{I}_n \otimes B) \mathbf{u}_i(s) ds \end{aligned} \quad (18)$$

Thus, we have:

$$\begin{aligned}
\delta \mathbf{x}_i(t) &= \mathbf{x}_d(t) - \mathbf{x}_i(t) \\
&= \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} ((\mathbf{I}_n \otimes \mathbf{A}) \delta \mathbf{x}_i(s-\tau) \\
&\quad + (\mathbf{I}_n \otimes \mathbf{B}) \delta \mathbf{u}_i(s)) ds \\
&= \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} (\mathbf{I}_n \otimes \mathbf{A}) \delta \mathbf{x}_i(s-\tau) ds \\
&\quad + \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} (\mathbf{I}_n \otimes \mathbf{B}) \delta \mathbf{u}_i(s) ds
\end{aligned} \tag{19}$$

Taking the norm on both sides of formula (19), the following can be derived from Definition 1:

$$\begin{aligned}
&\| (t)^{\alpha-1} (\mathbf{I}_n \otimes \mathbf{A}) \|_1 \| \delta \mathbf{x}_i(t) \|_p \\
\| \delta \mathbf{x}_i(t) \|_p &\leq \frac{+ \| (t)^{\alpha-1} (\mathbf{I}_n \otimes \mathbf{B}) \|_1 \| \delta \mathbf{u}_i(t) \|_p}{\Gamma(\alpha)}
\end{aligned} \tag{20}$$

If $\Gamma(\alpha) - \| (\cdot)^{\alpha-1} (\mathbf{I}_n \otimes \mathbf{A}) \|_1 > 0$ holds, we have:

$$\| \delta \mathbf{x}_i(t) \|_p \leq \frac{\| (t)^{\alpha-1} (\mathbf{I}_n \otimes \mathbf{B}) \|_1}{\Gamma(\alpha) - \| (t)^{\alpha-1} (\mathbf{I}_n \otimes \mathbf{A}) \|_1} \| \delta \mathbf{u}_i(t) \|_p \tag{21}$$

Then, the following can be derived from formulas (7) and (14):

$$\begin{aligned}
\delta \mathbf{u}_{i+1}(t) &= \delta \mathbf{u}_i(t) - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{p1}) \mathbf{e}_i(t) \\
&\quad - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D1}) \mathbf{e}_i^{(\alpha)}(t) \\
&\quad - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{p2}) \mathbf{e}_{i+1}(t) - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D2}) \mathbf{e}_{i+1}^{(\alpha)}(t)
\end{aligned} \tag{22}$$

Taking formulas (17) and (18) into formula (22):

$$\begin{aligned}
\delta \mathbf{u}_{i+1}(t) &= \delta \mathbf{u}_i(t) - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{p1}) (\mathbf{I}_n \otimes \mathbf{C}) \delta \mathbf{x}_i(t) \\
&\quad - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D1}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{A}) \delta \mathbf{x}_i(t-\tau) \\
&\quad - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D1}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{B}) \delta \mathbf{u}_i(t) \\
&\quad - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{p2}) (\mathbf{I}_n \otimes \mathbf{C}) \delta \mathbf{x}_{i+1}(t) \\
&\quad - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D2}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{A}) \delta \mathbf{x}_{i+1}(t-\tau) \\
&\quad - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D2}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{B}) \delta \mathbf{u}_{i+1}(t)
\end{aligned} \tag{23}$$

Sorting formula (23):

$$\begin{aligned}
&(\mathbf{I} + ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D2}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{B})) \delta \mathbf{u}_{i+1}(t) \\
&= (\mathbf{I} - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D1}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{B})) \delta \mathbf{u}_i(t) \\
&\quad - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{p1}) (\mathbf{I}_n \otimes \mathbf{C}) \delta \mathbf{x}_i(t) \\
&\quad - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D1}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{A}) \delta \mathbf{x}_i(t-\tau) \\
&\quad - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{p2}) (\mathbf{I}_n \otimes \mathbf{C}) \delta \mathbf{x}_{i+1}(t) \\
&\quad - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D2}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{A}) \delta \mathbf{x}_{i+1}(t-\tau)
\end{aligned} \tag{24}$$

Taking the norm on both sides of formula (24):

$$\begin{aligned}
&\| \mathbf{I} + ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D2}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{B}) \| \\
\| \delta \mathbf{u}_{i+1}(t) \|_p &\leq \left(\begin{array}{l} \| \mathbf{I} - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D1}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{B}) \| \\ \| (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{B}) \| + \beta_1 \end{array} \right) \\
&\| \delta \mathbf{u}_i(t) \|_p + \beta_2 \| \delta \mathbf{u}_{i+1}(t) \|_p
\end{aligned} \tag{25}$$

where,

$$\begin{aligned}
&\| ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{p1}) (\mathbf{I}_n \otimes \mathbf{C}) \| \\
&\quad + \| ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D1}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{A}) \| \\
\beta_1 &= \frac{\| (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{B}) \|}{\Gamma(\alpha) - \| (t)^{\alpha-1} (\mathbf{I}_n \otimes \mathbf{A}) \|_1} \| (t)^{\alpha-1} (\mathbf{I}_n \otimes \mathbf{B}) \|_1; \\
&\| ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{p2}) (\mathbf{I}_n \otimes \mathbf{C}) \| \\
\beta_2 &= \frac{+ \| ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D2}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{A}) \|}{\Gamma(\alpha) - \| (t)^{\alpha-1} (\mathbf{I}_n \otimes \mathbf{A}) \|_1} \\
&\| (t)^{\alpha-1} (\mathbf{I}_n \otimes \mathbf{B}) \|_1
\end{aligned}$$

Thus, the following can be derived from formula (25):

$$\begin{aligned}
&\| \mathbf{I} - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D1}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{B}) \| + \beta_1 \\
\| \delta \mathbf{u}_{i+1}(t) \|_p &\leq \frac{\| \mathbf{I} + ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D2}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{B}) \| - \beta_2}{\| \mathbf{I} - ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D1}) (\mathbf{I}_n \otimes \mathbf{C}) (\mathbf{I}_n \otimes \mathbf{B}) \| + \beta_1} \\
&\| \delta \mathbf{u}_i(t) \|_p = \frac{\rho_1}{\rho_2} \| \delta \mathbf{u}_i(t) \|_p
\end{aligned} \tag{26}$$

According to the Theorem, $\frac{\rho_1}{\rho_2} = \tilde{\rho} < 1$ holds. Then, the following can be derived from formula (26):

$$\| \delta \mathbf{u}_{i+1}(t) \|_p \leq \tilde{\rho} \| \delta \mathbf{u}_i(t) \|_p \leq \tilde{\rho}^k \| \delta \mathbf{u}_1(t) \|_p \tag{27}$$

Obviously, with the growing number of iterations, if $i \rightarrow \infty$, then:

$$\lim_{i \rightarrow \infty} \| \delta \mathbf{u}_{i+1}(t) \|_p = 0 \tag{28}$$

According to formulas (20) and (28), we have:

$$\lim_{i \rightarrow \infty} \| \delta \mathbf{x}_{i+1}(t) \|_p = 0 \tag{29}$$

That is:

$$\| \mathbf{e}_{i+1}(t) \|_p \leq \| \mathbf{I}_n \otimes \mathbf{C} \| \| \delta \mathbf{x}_{i+1}(t) \|_p \tag{30}$$

Hence,

$$\lim_{i \rightarrow \infty} \| \mathbf{e}_{i+1}(t) \|_p = 0 \tag{31}$$

Formula (31) indicates that the FOMAS error converges to zero.

Q.E.D.

For the controller (13), the open-closed-loop PD^α FOILC algorithm will degenerate into an open-loop PD^α FOILC algorithm, if learning gains $\Gamma_{p2}=0$ and $\Gamma_{D2}=0$:

$$\begin{aligned}
\mathbf{u}_{i+1}(t) &= \mathbf{u}_i(t) + ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{p1}) \mathbf{e}_i(t) \\
&\quad + ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D1}) \mathbf{e}_i^{(\alpha)}(t)
\end{aligned} \tag{32}$$

Corollary 1. For the controller (13), if Assumptions 1-2 hold, if distributed open-loop PD^α updating rule (32) is applied to the controller, and if matrices A, B, and C, as well as the learning gains Γ_{p1} and Γ_{D1} satisfy:

$$\rho = \| \mathbf{I} - ((\mathbf{L} + \mathbf{S}) \otimes \Gamma_{D1} \mathbf{CB}) \| + \beta < 1 \quad (33)$$

where, $\beta = \frac{\| \mathbf{L} + \mathbf{S} \| (\| \Gamma_{P1} \mathbf{C} \| + \| \Gamma_{D1} \mathbf{CA} \|)}{\Gamma(\alpha) - \| (t)^{\alpha-1} (\mathbf{I}_N \otimes \mathbf{A}) \|_1} \| (t)^{\alpha-1} (\mathbf{I}_N \otimes \mathbf{B}) \|_1$, then $\lim_{i \rightarrow \infty} \| e_{i+1}(t) \|_p = 0$. That is, the output $y_i(t)$ converges uniformly to the desired trajectory $y_d(t)$ as $i \rightarrow \infty$.

Similarly, for the controller (13), the open-closed-loop PD $^\alpha$ FOILC algorithm will degenerate into a closed-loop PD $^\alpha$ FOILC algorithm, if learning gains $\Gamma_{P1}=0$ and $\Gamma_{D1}=0$:

$$\begin{aligned} \mathbf{u}_{i+1}(t) &= \mathbf{u}_i(t) + ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{P2}) \mathbf{e}_i(t) \\ &+ ((\mathbf{L} + \mathbf{D}) \otimes \Gamma_{D2}) \mathbf{e}_i^{(\alpha)}(t) \end{aligned} \quad (34)$$

Corollary 2. For the controller (13), if Assumptions 1-2 hold, if distributed open-loop PD $^\alpha$ updating rule (32) is applied to the controller, and if matrices A, B, and C, as well as the learning gains Γ_{P1} and Γ_{D1} satisfy:

$$\rho = \frac{1}{\| \mathbf{I} - ((\mathbf{L} + \mathbf{S}) \otimes \Gamma_{D1} \mathbf{CB}) \| - \rho} < 1 \quad (35)$$

where, $\rho = \frac{\| \mathbf{L} + \mathbf{S} \| (\| \Gamma_{P1} \mathbf{C} \| + \| \Gamma_{D1} \mathbf{CA} \|)}{\Gamma(\alpha) - \| (t)^{\alpha-1} (\mathbf{I}_N \otimes \mathbf{A}) \|_1} \| (t)^{\alpha-1} (\mathbf{I}_N \otimes \mathbf{B}) \|_1$, then $\lim_{i \rightarrow \infty} \| e_{i+1}(t) \|_p = 0$. That is, the output $y_i(t)$ converges uniformly to the desired trajectory $y_d(t)$ as $i \rightarrow \infty$.

Remark 3. Under the conditions of the Theorem, and Corollaries 1 and 2, the convergence condition of the control law in the sense of Lebesgue- p norm depends on the learning gains and the intrinsic properties of the FOMAS.

Remark 5. Since the directed graph \bar{G} is a connected graph, and the matrix $\mathbf{L} + \mathbf{S}$ is positive definite, matrix $-(\mathbf{L} + \mathbf{S})$ is a Hurwitz stable matrix. Thus, the gain matrices Γ_{P1} , Γ_{D1} , Γ_{P2} , Γ_{D2} to satisfy the conditions in formula (13), (32) or (34).

4. SIMULATION

To verify the effectiveness of the proposed algorithms, an FOMAS with state delay was set up as Figure 1, where 0 is the leader, 1-4 are the followers, and dotted lines are the links between the agents.

From Figure 1, the adjacency matrix \mathbf{M} and \mathbf{D}_m can be respectively defined as:

$$\mathbf{M} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{D}_m = \text{diag}[1 \quad 1 \quad 2 \quad 1]$$

Then, the Laplacian matrix can be obtained:

$$\mathbf{L} = \mathbf{D} - \mathbf{M} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & 0 & 1 \end{bmatrix}, \quad \mathbf{S} = \text{diag}[1, 0, 1, 0]$$

Here, each FOMAS agent with state delay can be described as:

$$\begin{cases} D^\alpha \mathbf{x}_j(t) = \begin{bmatrix} 0.4 & 2 \\ 5 & -6 \end{bmatrix} \mathbf{x}_j(t-h) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{u}_j(t), \\ \mathbf{y}_j(t) = [0 \quad 1.2] \mathbf{x}_j(t), \end{cases} \quad (36)$$

The desired trajectory of the agent can be described as $y_d(t) = 2t^2 + 3t^3 + \sin(2\pi t)$, $t \in [0, 1]$. During the simulation, the fractional derivative was set as $\alpha=0.85$. For each agent, the initial state and initial input in each iteration were set to zero: $u_{0,j}(t)=0$, $x_{i,j}(0)=0$, and $j=1, 2, 3, 4$.

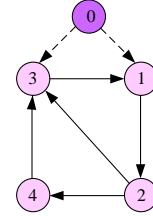
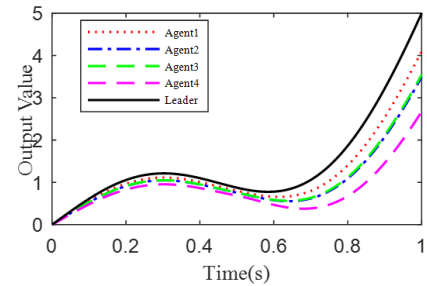


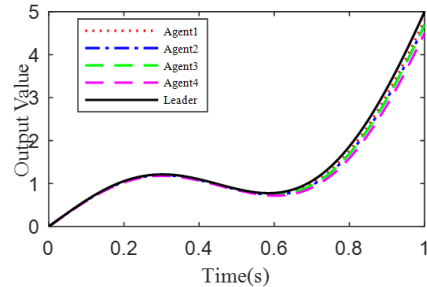
Figure 1. The structure of FOMAS

4.1 Case 1: Open-closed-loop PD $^\alpha$ FOILC algorithm

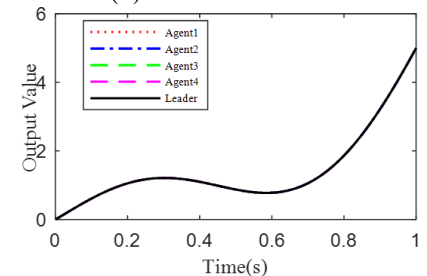
According to the Theorem, the gain matrices were selected as $\Gamma_{P1}=1.2$, $\Gamma_{D1}=0.25$, $\Gamma_{P2}=1.6$, and $\Gamma_{D2}=1.1$. Then, we have $\rho_1=0.876$ and $\rho_2=0.963$, which satisfy the conditions in the Theorem. To verify the robustness of open-closed-loop PD $^\alpha$ FOILC algorithm to state delay, the delay time was set to 0.1 and 0.5, that is, $h=0.1$ and $h=0.5$. The simulation results with $h=0.1$ and $h=0.5$ are recorded in Figures 2 and 3, respectively. Figures 2(a)-(c) and 3(a)-(c) present the tracking results in the 5th, 10th, and 50th iterations, respectively; Figure 2(d) and Figure 3(d) show the absolute maximum error of each iteration.



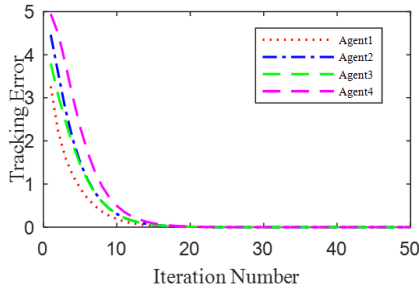
(a) The 5th iteration



(b) The 10th iteration

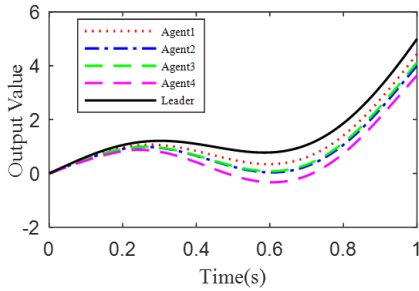


(c) The 50th iteration

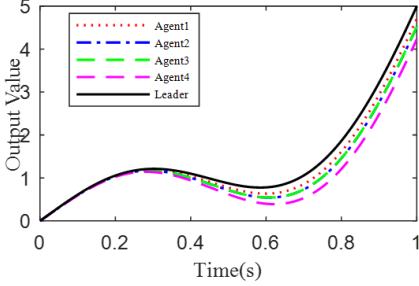


(d) The variation in tracking error

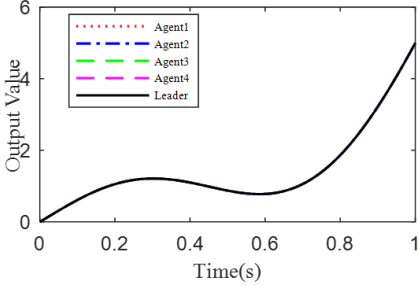
Figure 2. The simulation results with $h=0.1s$



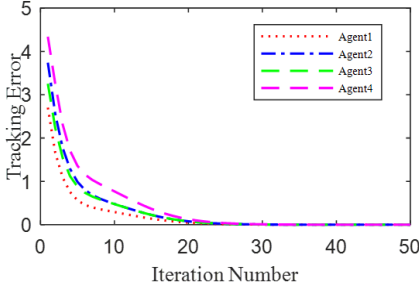
(a) The 5th iteration



(b) The 10th iteration



(c) The 50th iteration



(d) The variation in tracking error

Figure 3. The simulation results with $h=0.5s$

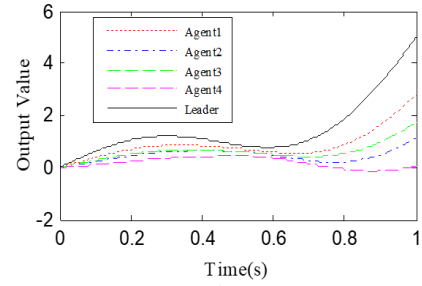
It can be seen that, with the growing number of iterations, each agent gradually tracked the desired trajectory. When $h=0.1$, the maximum errors of the four followers were 0.0009, 0.0012, 0.0011, and 0.0015 at the 50th iteration, respectively. When $h=0.5$, the convergence speed was slower than that under $h=0.1$. Concerning the convergence speed of each agent, agents 1 and 3 converged faster than agents 2 and 4, because

the former two can directly receive the information from the leader.

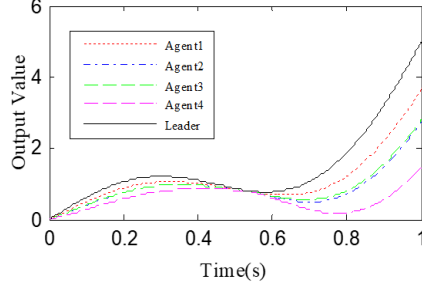
4.2 Case 2: Open-loop PD ^{α} FOILC algorithm

According to Corollary 1 and controller (32), the gain matrices were defined as $\Gamma_{P1}=1.2$ and $\Gamma_{D1}=0.25$. Similar to Case 1, the delay time was set to 0.1 and 0.5, namely $h=0.1$ and $h=0.5$. The simulation results with $h=0.1$ and $h=0.5$ are recorded in Figures 4 and 5, respectively. Figures 4(a)-(c) and 4(a)-(c) present the tracking results in the 5th, 10th, and 50th iterations, respectively; Figure 5(d) and Figure 5(d) show the absolute maximum error of each iteration.

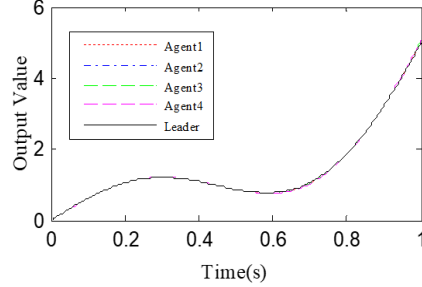
It can be seen that, with the growing number of iterations, the multiple agents gradually tracked the desired trajectory. With the increase in delay time, the convergence speed slowed down. In the same iteration, the open-loop PD ^{α} FOILC algorithm performed poorer than the open-closed-loop PD ^{α} FOILC algorithm. This means the inclusion of the closed-loop control in the latter algorithm improves the control effect and speeds up the convergence.



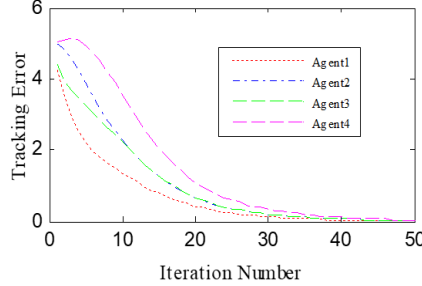
(a) The 5th iteration



(b) The 10th iteration

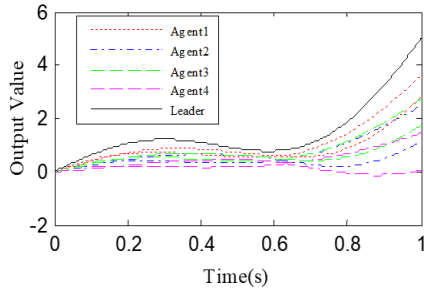


(c) The 50th iteration

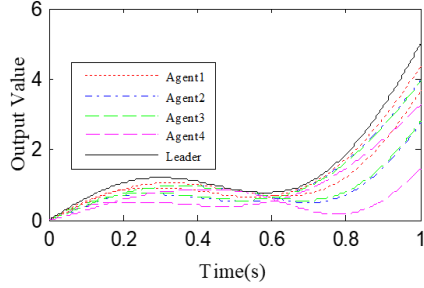


(d) The variation in tracking error

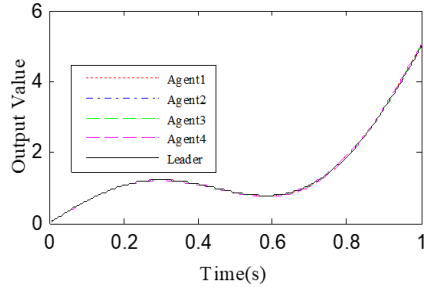
Figure 4. The simulation results with $h=0.1s$



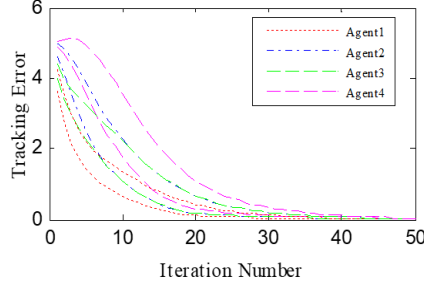
(a) The 5th iteration



(b) The 10th iteration



(c) The 50th iteration

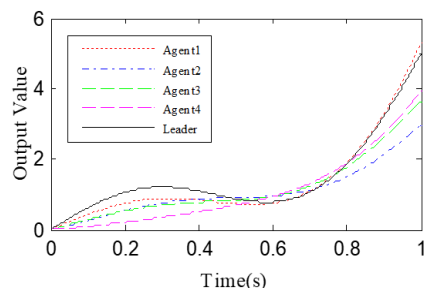


(d) The variation in tracking error

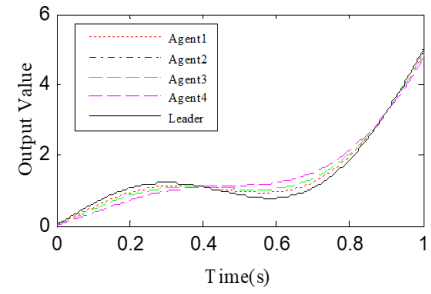
Figure 5. The simulation results with $h=0.5s$

4.3 Case 3: Closed loop PD ^{α} FOILC algorithm

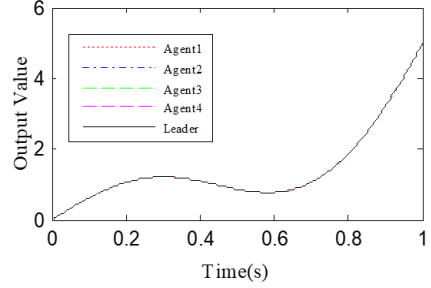
According to Corollary 1 and controller (32), the gain matrices were defined as $\Gamma_{P2}=1.6$ and $\Gamma_{D2}=1.1$. The initial conditions and delay time were configured the same as Cases 1 and 2. The simulation results with $h=0.1$ and $h=0.5$ are recorded in Figures 6 and 7, respectively.



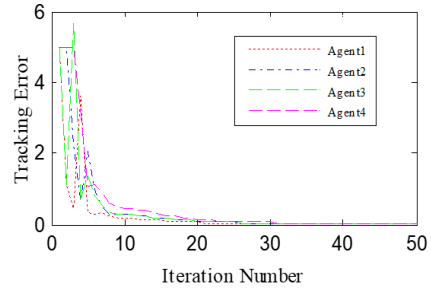
(a) The 5th iteration



(b) The 10th iteration

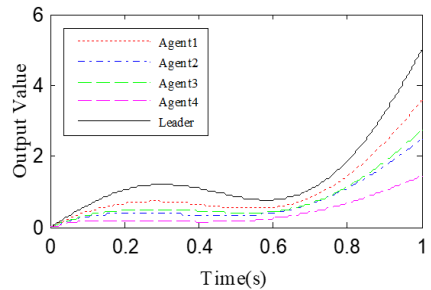


(c) The 50th iteration

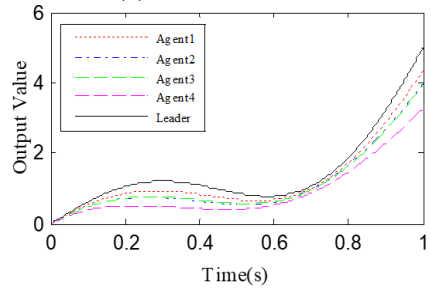


(d) The variation in tracking error

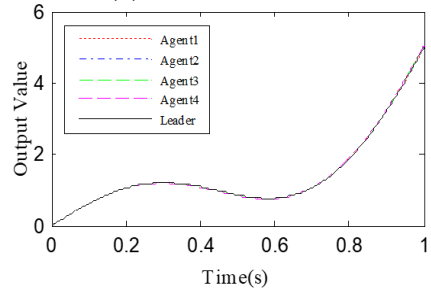
Figure 6. The simulation results with $h=0.1s$



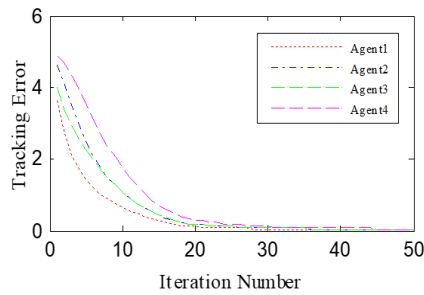
(a) The 5th iteration



(b) The 10th iteration



(c) The 50th iteration



(d) The variation in tracking error

Figure 7. The simulation results with $h=0.5s$

Figures 6(a)-(c) and 7(a)-(c) present the tracking results in the 5th, 10th, and 50th iterations, respectively; Figure 6(d) and Figure 7(d) show the absolute maximum error of each iteration.

The simulation results show that the closed-loop PD ^{α} FOILC algorithm converged slower than the open-closed-loop PD ^{α} FOILC algorithm and open-loop PD ^{α} FOILC algorithm.

In nature, many physical systems have fractional-order dynamic features. But the integer calculus theory in this research is merely a special case of fractional-order calculus theory. It can only approximate the actual fractional-order system. However, the theory of fractional calculus can reveal the nature of the object truthfully and accurately. The proposed fractional-order controller can achieve a good control performance, which highlights the significance of the research results.

5. CONCLUSIONS

This paper proposes an open-closed-loop PD ^{α} FOILC algorithm for FOMAS with state delay. Based on norm theory, graph theory and fractional calculus, the convergence of the proposed algorithm was analyzed in details, and the convergence conditions were summarized. Through theoretical analysis, the proposed algorithm was found to iteratively minimize the error of each agent in FOMAS with state delay. Finally, the algorithm was proved effective through numerical simulation.

REFERENCES

- [1] Liu, X. (2020). Research on decision-making strategy of soccer robot based on multi-agent reinforcement learning. *International Journal of Advanced Robotic Systems*, 17(3): 1729881420916960. <https://doi.org/10.1177/1729881420916960>
- [2] Ota, J. (2006). Multi-agent robot systems as distributed autonomous systems. *Advanced Engineering Informatics*, 20(1): 59-70. <https://doi.org/10.1016/j.aei.2005.06.002>
- [3] Perrusquía, A., Yu, W., Li, X. (2020). Multi-agent reinforcement learning for redundant robot control in task-space. *International Journal of Machine Learning and Cybernetics*, 1-11. <https://doi.org/10.1007/s13042-020-01167-7>
- [4] Osherenko, A. (2001). Plan Representation and Plan Execution in Multi-agent Systems for Robot Control. In *PuK*.
- [5] Naserian, M., Ramazani, A., Khaki-Sedigh, A., Moarefianpour, A. (2020). Fast terminal sliding mode control for a nonlinear multi-agent robot system with disturbance. *Systems Science & Control Engineering*, 8(1): 328-338. <https://doi.org/10.1080/21642583.2020.1764408>
- [6] Gu, P., Tian, S. (2019). Consensus tracking control via iterative learning for singular multi-agent systems. *IET Control Theory & Applications*, 13(11): 1603-1611. <https://doi.org/10.1049/iet-cta.2018.5901>
- [7] Demir, O., Lunze, J. (2014). Optimal and event-based networked control of physically interconnected systems and multi-agent systems. *International Journal of Control*, 87(1): 169-185. <https://doi.org/10.1080/00207179.2013.825816>
- [8] Liu, X., Zhang, Z., Liu, H. (2017). Consensus control of fractional-order systems based on delayed state fractional order derivative. *Asian Journal of Control*, 19(6): 2199-2210. <https://doi.org/10.1002/asjc.1493>
- [9] Liu, J., Chen, W., Qin, K., Li, P. (2018). Consensus of fractional-order multiagent systems with double integral and time delay. *Mathematical Problems in Engineering*, 2018. <https://doi.org/10.1155/2018/6059574>
- [10] Bensafia, Y., Ladaci, S., Khettab, K., Chemori, A. (2018). Fractional order model reference adaptive control for SCARA robot trajectory tracking. *International Journal of Industrial and Systems Engineering*, 30(2): 138-156. <https://doi.org/10.1504/IJISE.2018.094839>
- [11] Yu, Z., Jiang, H., Hu, C. (2015). Leader-following consensus of fractional-order multi-agent systems under fixed topology. *Neurocomputing*, 149: 613-620. <https://doi.org/10.1016/j.neucom.2014.08.013>
- [12] Yu, Z., Jiang, H., Hu, C., Yu, J. (2015). Leader-following consensus of fractional-order multi-agent systems via adaptive pinning control. *International Journal of Control*, 88(9): 1746-1756. <https://doi.org/10.1080/00207179.2015.1015807>
- [13] Bai, J., Wen, G., Rahmani, A., Chu, X., Yu, Y. (2016). Consensus with a reference state for fractional-order multi-agent systems. *International Journal of Systems Science*, 47(1): 222-234. <https://doi.org/10.1080/00207721.2015.1056273>
- [14] Ma, X., Sun, F., Li, H., He, B. (2017). The consensus region design and analysis of fractional-order multi-agent systems. *International Journal of Systems Science*, 48(3): 629-636. <https://doi.org/10.1080/00207721.2016.1218570>
- [15] Yu, W., Li, Y., Wen, G., Yu, X., Cao, J. (2016). Observer design for tracking consensus in second-order multi-agent systems: Fractional order less than two. *IEEE Transactions on Automatic Control*, 62(2): 894-900. <https://doi.org/10.1109/TAC.2016.2560145>
- [16] Chai, Y., Chen, L., Wu, R., Sun, J. (2012). Adaptive pinning synchronization in fractional-order complex dynamical networks. *Physica A: Statistical Mechanics and Its Applications*, 391(22): 5746-5758. <https://doi.org/10.1016/j.physa.2012.06.050>
- [17] Sun, M., Wang, D. (2002). Iterative learning control with initial rectifying action. *Automatica*, 38(7): 1177-1182. [https://doi.org/10.1016/S0005-1098\(02\)00003-1](https://doi.org/10.1016/S0005-1098(02)00003-1)
- [18] Sun, M., Ge, S.S., Mareels, I.M. (2006). Adaptive repetitive learning control of robotic manipulators without the requirement for initial repositioning. *IEEE Transactions on Robotics*, 22(3): 563-568. <https://doi.org/10.1109/TRO.2006.870650>
- [19] Park, K.H. (2005). An average operator-based PD-type iterative learning control for variable initial state error.

- IEEE Transactions on Automatic Control, 50(6): 865-869.
<https://doi.org/10.1109/TAC.2005.849249>
- [20] Meng, D., Jia, Y. (2011). Finite-time consensus for multi-agent systems via terminal feedback iterative learning.
- IET Control Theory & Applications, 5(18): 2098-2110.
<https://doi.org/10.1049/iet-cta.2011.0047>
- [21] Yang, S.P. (2014). On iterative learning in multi-agent systems coordination and control (Doctoral dissertation).