International Journal of Research and Technology in Electrical Industry

journal homepage: ijrtei.sbu.ac.ir

Shahid Beheshti University

Data-Driven Distributed Fault Diagnosis in Multi-Agent Systems with Fractional-Order Sliding Mode Estimator

Mohsen Shiri[1] , Hadi Delavari [2,*] , Younes Solgi [1]

[1] *Department of Electrical Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran*
[2] *Electrical Engineering Department, Hamedan University of Technology, Hamedan, Iran*

ARTICLE INFO

Article history:
Received: 29 October 2025
Revised: 18 May 2026
Accepted: 16 June 2026

Keywords:
Distributed sliding mode controller
Distributed sliding mode observer
Discrete nonlinear multi-agent systems
Reinforcement learning
Fractional-order estimator
Distributed fault diagnosis

ABSTRACT

As large-scale systems and industrial processes become more complex and widespread, guaranteeing their safety and reliability is now a primary concern for control and intelligent monitoring. The performance of these Multi-Agent Systems (MAS) is highly vulnerable to faults, leading to significant disruptions. This paper introduces a new Model Free-Distributed Sliding Mode Control (MF-DSMC) method to ensure agents in a homogeneous nonlinear MAS accurately track a reference signal. This approach is combined with a Model Free-Distributed Fractional-Order Sliding Mode Estimator (MF-DFOSME) to provide fault diagnosis. Within the fault detection framework, we develop a Model-Free-Distributed Sliding Mode Observer (MF-DSMO) that utilizes distributed estimation errors to facilitate rapid fault detection. This is done by monitoring the residual signal; if its magnitude crosses a predefined threshold, it indicates a system fault. By integrating the MF-DFOSME, we enhance the performance of the fault diagnosis system (FDS). This combination provides key advantages, including greater robustness, long-term memory, and highly accurate and rapid fault estimation. Finally, we optimize the design parameters using a Reinforcement Learning (RL) algorithm. We use Lyapunov functions to provide a formal stability guarantee for the proposed algorithms. Simulation results confirm that our methods significantly boost MAS performance. The MF-DSMC method yielded a 17% performance improvement, while the proposed FDS method achieved a 26% enhancement over the baseline techniques.

1. Introduction

Over the past few years, with significant advancements in communication technologies and the increasing scalability and intelligent cooperation of systems, the growing interest in MAS highlights their crucial role in shaping and advancing industrial ecosystems [1]. The consensus problem represents a pivotal topic within the study of MAS. It focuses on how agents can optimally use exchanged information, and engage in mutual interactions to collectively accomplish shared objectives [2]. Extensive and integrated connectivity among objects, machines, and humans—which represents a novel approach to the application of intelligent communications and the utilization of emerging network infrastructures—is realized through cyber-physical systems (CPS). Key challenges in this domain include coordinating components optimally, executing remote control, and detecting faults via FDS remotely [3]. Challenges such as distributed fault detection across sensors among agents in MAS [4], the implementation of distributed control across geographically dispersed subsystems [5], and the accurate modeling of subsystems composed of heterogeneous

* Corresponding author
E-mail address: delavari@hut.ac.ir
https://orcid.org/0000-0002-5718-933X
https://doi.org/10.48308/ijrtei.2026.242242.1106

components significantly impact the performance of CPS [6]. Distributed control and fault diagnosis are critical for applications like multi-robot collaboration [7], smart grids [8], intelligent transportation systems [9], and sensor networks [10], where system monitoring and control are paramount. In such MAS, safety and reliability issues are of paramount importance, as a fault or failure in a single agent can have serious and widespread consequences for the overall MAS. Consequently, fault detection has become a central challenge for the intelligent monitoring and control of large-scale systems. Modeling and simulation platforms are crucial for developing and evaluating how well new algorithms can be implemented. For this reason, they remain a significant focus for researchers [11].

The increasing scale of systems and rising performance expectations, combined with inherent complexity, nonlinear behavior, and the continuous evolution of real-world systems, have made accurate modeling a fundamental challenge. Unlike traditional modelling approaches, data-driven methods leverage input-output data to approximate complex system functions, thereby eliminating the need for precise structural modelling [12]. Recently, data-driven approaches have gained prominence in the field of control, monitoring, and fault diagnosis, and this area has grown significantly across various engineering fields [13,14]. Advances in information technology, greater data availability, and better sensors and computing power have made data-driven techniques essential for industrial digitalization [15].

In recent years, the research community has shown considerable interest in model-free adaptive control (MFAC) for achieving distributed consensus in MAS with unknown dynamics [16]. The MFAC method maintains a consistent controller structure and tunes its parameters, yet it has demonstrated highly effective performance across diverse applications involving various cooperative tasks [17]. Accurate and high-quality input–output data are essential for ensuring optimal performance of MFAC controllers. In many current studies on consensus in MAS, it is assumed that agents operate under normal conditions. In practical applications, MAS consist of multiple agents, and the occurrence of faults in these agents is inevitable. Since a single agent's fault can spread across the network, ignoring it risks system instability, performance loss, and corrupted data. A FDS is therefore essential for timely corrective action. This system uses a detection section to identify faults and an estimation section to assess their magnitude and severity. Recent years have seen significant advances in fault diagnosis for MAS. For instance, in study [18], Second-order multi-robot systems employed unknown input observers to achieve effective fault detection and isolation. In study [19], a data-driven strategy for distributed fault diagnosis in linear MAS was proposed using subspace-based techniques. Study [20] developed a data-driven distributed fault diagnosis framework based on the Kalman filter to monitor the safety of large-scale dynamic processes. This framework offers high-speed processing capabilities. Therefore, there is a growing need for data-driven, distributed fault detection and estimation methods that can operate using the sensor network's own structure, without needing an accurate system model. Therefore, fault detection and estimation in nonlinear, unknown MAS—including the optimization of observer and estimator gains—remains a key area for further research.

The motivation for mentioning the methods above is to address challenges such as distributed control, modelling complexity in MAS, and fault detection and estimation within these systems. A novel model-free approach for distributed control and fault diagnosis has been proposed, enabling remote tracking control of agents and fault detection and estimation via a cloud server. The details of the innovations and scientific contributions of the proposed method are described below:

- A novel MF-DSMC is introduced, which can rapidly and effectively track reference changes without requiring precise mathematical modelling.
- An MF-DSMO is introduced, capable of identifying faults with high speed and accuracy without requiring precise mathematical modelling.
- A novel MF-DFOSME is proposed, capable of accurately estimating the magnitude and severity of faults occurring in agents with high precision and minimal error, without requiring precise mathematical modelling.
- The proposed data-driven method is implemented for remote control and fault diagnosis on cloud servers, enabling MAS with terminals distributed across different geographical locations to share cloud computational resources and achieve consensus tracking in both control and fault diagnosis.
- Combining a sliding mode estimator with fractional-order calculus strengthens its robustness, incorporates long-term memory effects, and enhances both estimation accuracy and convergence speed.
- The design parameters have been effectively tuned using an RL algorithm, resulting in reduced tracking errors and enhanced accuracy in fault estimation.

The subsequent sections of the paper are organized as outlined below. The theoretical foundations and required formulations are presented in Section 2. The proposed control strategy is detailed in Section 3. Section 4 covers the distributed fault diagnosis system. Section 5 presents the RL algorithm. The simulation results and their discussion are provided in Section 6. Section 7 outlines the conclusions of the study.

2. Problem Formulation

2.1 Data-Driven Model

The system investigated in this study is considered a MAS with a fixed and homogeneous topology. This work addresses a discrete-time, nonlinear, single-input single-output (SISO) MAS composed of N agents. The dynamics for the i agent are given by:

$$O_i(k+1) = \Omega\big(O_i(k), \dots, O_i(k - \kappa_O), \mathcal{u}_i(k), \dots, \mathcal{u}_i(k - \kappa_u)\big) \quad (1)$$

where the output $O_i(k) \in \mathbb{R}$ and the input $\mathcal{u}_i(k) \in \mathbb{R}$ of the i agent is governed by a nonlinear function $\Omega(\cdot)$.

Assumption 1: The nonlinear function $\Omega(\cdot)$ is continuously differentiable with respect to the control input sequence $u_i(k)$.

The partial derivatives of the nonlinear function $\Omega(\cdot)$ with respect to the control signal sequence $u_i(k)$ are continuous, and the MAS model (1) is generalized Lipschitz for all k.

Compact Form Dynamic Linearization (CFDL) is one of the most prevalent techniques in data-driven control design, and is applied to the system described in (1) [21]:

$$\Delta O_i(k+1) = \Theta_i(k)\Delta u_i(k) \tag{2}$$

where $\Delta O_i(k+1) = O_i(k+1) - O_i(k)$, $\Theta_i(k)$ denotes the pseudo partial derivative (PPD) of the i agent, and $\Delta u_i(k) = u_i(k) - u_i(k-1)$ is defined accordingly. The PPD serves as a data-driven approximation of the system's dynamic gradient with respect to the input.

At each sampling instant, the PPD must be identified to apply the control signal to the data-driven system. The PPD is estimated as follows [21]:

$$\hat{\Theta}_i(k) = \hat{\Theta}_i(k-1) + \frac{\nu \Delta u_i(k-1)}{\rho + \Delta u_i(k-1)^2} \times \left[\Delta O_i(k) - \hat{\Theta}_i(k-1)\Delta u_i(k-1)\right] \tag{3}$$

the parameters $\nu > 0$ and $\rho > 0$ denote the degrees of freedom in the design, and the detailed proof of equation (3) is presented in [21]. By utilizing the estimated PPD, (2) can be expressed as follows:

$$\Delta O_i(k+1) = \hat{\Theta}_i(k)\Delta u_i(k) \tag{4}$$

If a fault occurs in the sensor of the i agent, the measured output will be affected. Considering the sensor fault in (4), we obtain:

$$\Delta O_i(k+1) = \hat{\Theta}_i(k)\Delta u_i(k) + \Delta \mathcal{F}_i(k) \tag{5}$$

where $\mathcal{F}_i(k)$ represents the sensor fault in the agent i, and $\Delta \mathcal{F}_i(k) = \mathcal{F}_i(k) - \mathcal{F}_i(k-1)$ is considered.

Assumption 2: It is assumed that the sensor fault affecting each agent remains within known bounds.

Assumption 3: It is worth noting that the PPD estimator (3) relies on measured input–output data. In the presence of an incipient sensor fault, the measured output may be temporarily corrupted before the fault detection mechanism becomes active. In this work, it is assumed that sensor faults introduce bounded deviations rather than catastrophic failures. Under this practical assumption, the estimated PPD remains bounded and preserves its sign, which guarantees the closed-loop stability during the short transient interval prior to fault detection.

Remark 1: Assumption 1 highlights an inherent design procedure and the output's reliance on the system input. Given the practical constraints of sensors and communication devices, Assumption 2 is introduced, Assumption 3 guarantees the stability of the system in the presence of a sensor fault.

Ultimately, the output of the i agent under sensor fault can be expressed in the following form:

$$O_i(k+1) = O_i(k) + \hat{\Theta}_i(k)\Delta u_i(k) + \mathcal{F}_i(k) \tag{6}$$

model (6) will be employed in the design section.

2.2 Fault detection threshold

In control and monitoring systems, the fault detection threshold (FDT) is a numerical boundary or range established to distinguish between the "healthy" and "faulty" conditions of the system. The purpose of establishing this threshold is to reduce the false alarm rate while improving the sensitivity of fault detection. The discrete FDT is defined as follows [22]:

$$\delta(k+1) = (1-\gamma)\delta(k) + \gamma\delta(\infty) \tag{7}$$

$$0 < \delta(\infty) < \delta(0) \tag{8}$$

where $0 < \gamma < 1$ represents the convergence rate, and the initial value of $\delta(k)$ is given by $\delta(0)$. The final value at convergence is expressed as follows:

$$\lim_{k\to\infty} \delta(k) = \delta(\infty) \tag{9}$$

The output error between the reference model and the actual system is represented by the residual signal $r(k)$. The detection threshold is given by the following expression:

$$-\delta(k) < r(k) < \delta(k) \tag{10}$$

if (10) holds for an initial residual $r(0)$, then its realization is deemed satisfactory.

2.3 Discrete Fractional Calculus

Integer-order calculus can be regarded as a particular case of fractional-order calculus. Discrete fractional calculus provides a powerful mathematical framework for modelling and control of dynamical systems exhibiting memory and hereditary properties. Among the available formulations, the Grünwald–Letnikov (GL) definition represents one of the most fundamental and widely used approaches for the discrete implementation of fractional operators.

Let $\alpha \in (0,1]$ denote the fractional order and define the shifted natural number set as:

$$\mathbb{N}_{1-\alpha} = \{1-\alpha, 2-\alpha, 3-\alpha, \dots\}$$

Consider a discrete-time function $G(k)$ defined on $k \in \mathbb{N}$. The GL fractional difference operator of order α denoted by ${}^{GL}\Delta_0^{\alpha}\mathcal{G}(k)$, is defined as [23]:

$${}^{GL}\Delta_0^{\alpha}\mathcal{G}(k) = \frac{1}{d^{\alpha}} \sum_{m=0}^{k} (-1)^m \binom{\alpha}{m} \mathcal{G}(k-m) \tag{11}$$

where $d > 0$ denotes the discrete sampling interval (time step). The coefficients appearing in (12) are the GL weighting coefficients, obtained from the generalized binomial coefficients. For a non-integer order α, the binomial coefficient is defined using the Gamma function as:

$$\binom{\alpha}{m} = \frac{\Gamma(\alpha+1)}{\Gamma(m+1)\Gamma(\alpha-m+1)} \quad m = 0,1,2,\dots \tag{12}$$

where $\Gamma(\cdot)$ denotes the Gamma function.

For computational purposes, the coefficients can equivalently be expressed in recursive form as:

$$\begin{cases} C_0^{(\alpha)} = 1 & m = 0 \\ C_m^{(\alpha)} = \left(\dfrac{\alpha(\alpha-1)\cdots(\alpha-m+1)}{m!}\right) & m > 0 \end{cases} \tag{13}$$

where $C_m^{(\alpha)} = (-1)^m \binom{\alpha}{m}$. These coefficients act as convolution weights in (11) and explicitly capture the memory effect characteristic of fractional-order systems. The use of fractional-order difference operators enhances controller performance by improving modelling accuracy and providing additional flexibility in the design of discrete-time control algorithms, particularly for systems exhibiting long-term memory and nonlocal dynamics [24].

Assumption 4: In theory, the GL fractional difference in (11) requires the use of the complete past history of the signal, which may lead to a high computational burden in long-time and real-time implementations. In this work, the GL operator is implemented based on the short-memory principle, where the infinite summation is truncated to a finite memory window of length L. Accordingly, the fractional difference is computed using only the most recent L samples. Because the binomial coefficients of the GL operator decay rapidly, the contribution of distant past samples becomes negligible. Therefore, truncating the memory introduces only a bounded approximation error and does not affect the stability of the proposed MF-DFOSME scheme, provided that the memory length is chosen sufficiently large.

3. Proposed MF-DSMC

In this work, an ideal communication network between the agents and the cloud server is assumed, and effects such as transmission delays, jitter, and packet losses are not explicitly considered in the mathematical modelling. This assumption allows us to focus on the proposed MF-DSMC design and its fault-tolerant capabilities. Nevertheless, due to the robustness properties of sliding mode-based control, the proposed scheme can tolerate small bounded communication imperfections. The extension to explicit networked control settings with delays and packet dropouts is an interesting direction for future research. The details related to the proposed MF-DSMC controller are discussed in this section. The error between agents takes the following form:

$$\varrho_{c_{ij}}(k) = O_j(k) - O_i(k) \tag{14}$$

The discrepancy between the agents' outputs and the reference signal is mathematically characterized as:

$$\varrho_{c_i}(k) = O_d(k) - O_i(k) \tag{15}$$

Taking into account the mutual interactions among agents, the distributed control strategy is formulated based on a weighted combination of the output tracking errors between the agents and the reference signal, along with the error between the agents' outputs themselves, as follows:

$$\mathcal{E}_{c_i}(k) = \sum_{j\in N} a_{ij}\left(O_j(k) - O_i(k)\right) + \mathscr{b}_i\left(O_d(k) - O_i(k)\right) \tag{16}$$

the element a_{ij} the connectivity matrix defines the communication topology among agents, such that $a_{ij} = 1$ implies that agent i receives information from agent j, whereas $a_{ij} = 0$ indicates no information exchange between them. Furthermore, the variable $\mathscr{b}_i$ specifies whether agent i can access the reference signal; $\mathscr{b}_i = 1$ denotes access to the reference, while $\mathscr{b}_i = 0$ indicates the absence of such information. By defining $\mathcal{A} = diag\{a_1, \dots, a_N\}$ and introducing the variable transformation $d_i = \sum_{j=1}^{N} a_{ij}$ with $\mathcal{D} = diag\{d_1, \dots, d_N\}$, the Laplacian matrix can be expressed as $\mathcal{L} = \mathcal{D} - \mathcal{A}$.

The distributed algorithm can equivalently be expressed using the Laplacian matrix as below:

$$\mathcal{E}_c(k) = \left((\mathcal{L} + \mathcal{B})\otimes I_n\right)\varrho_c(k) = \mathcal{L}_{\mathcal{B}}\varrho_c(k) \tag{17}$$

in this formulation $\varrho_c(k) = O_d(k) - O(k) = [\varrho_1^T(k), \dots, \varrho_N^T(k)]^T, O_d(k) = [O_d^T(k), \dots, O_d^T(k)]^T \in \mathbb{R}^{n\times N}, O(k) = [O_1^T(k), \dots, O_N^T(k)]^T, \mathcal{B} = diag\{\mathscr{b}_1, \dots, \mathscr{b}_N\}$, and $\mathcal{L}_{\mathcal{B}} = \left((\mathcal{L} + \mathcal{B})\otimes I_n\right)$, The Kronecker product is denoted by the symbol $\otimes$, and I_n represents the $N \times N$ identity matrix.

The variations in the error of the distributed algorithm are expressed in the following manner:

$$\begin{aligned}\Delta\mathcal{E}_c(k) &= \mathcal{E}_c(k) - \mathcal{E}_c(k-1) \\ &= \mathcal{L}_{\mathcal{B}}\varrho_c(k) - \mathcal{L}_{\mathcal{B}}\varrho_c(k-1) \\ &= \mathcal{L}_{\mathcal{B}}\Delta\varrho_c(k)\end{aligned} \tag{18}$$

where $\Delta\varrho_c(k) = \varrho_c(k) - \varrho_c(k-1)$.

The derivative sliding surface for the discrete sliding mode controller is defined as follows:

$$\sigma_c(k) = \lambda_1\mathcal{E}_c(k) + \lambda_2\Delta\mathcal{E}_c(k-1) \tag{19}$$

where $\lambda_1, \lambda_2 > 0$ are the controller design parameters, which are determined through the optimization and tuning process using RL, and $\sigma_c(k) = \left[\sigma_{c_1}^T(k), \dots, \sigma_{c_N}^T(k)\right]^T$. $\Delta\mathcal{E}_c(k)$ represents the discrete-time derivative form.

In the control law design, the $k+1$ step can be utilized, such that by reformulating (19), we obtain:

$$\begin{aligned}\sigma_c(k+1) &= \lambda_1\mathcal{E}_c(k+1) + \lambda_2\Delta\mathcal{E}_c(k) \\ &= \lambda_1\mathcal{L}_{\mathcal{B}}\varrho_c(k+1) + \lambda_2\mathcal{L}_{\mathcal{B}}\Delta\varrho_c(k) \\ &= \lambda_1\mathcal{L}_{\mathcal{B}}\left(O_d(k+1) - O(k+1)\right) \\ &+\lambda_2\mathcal{L}_{\mathcal{B}}\Delta\varrho_c(k) \\ &= \lambda_1\mathcal{L}_{\mathcal{B}}(O_d(k+1) - O(k) - \widehat{\Theta}(k)\Delta u(k) \\ &+\lambda_2\mathcal{L}_{\mathcal{B}}\Delta\varrho_c(k)\end{aligned} \tag{20}$$

where $\widehat{\Theta}(k) = diag\left(\widehat{\Theta}_1(k), \dots, \widehat{\Theta}_N(k)\right)$, and $\Delta u(k) = [\Delta u_1^T(k), \dots, \Delta u_N^T(k)]^T$.

In the sliding mode control reaching law, the reaching condition must be satisfied, which is ensured by the following equation:

$$\Delta\sigma_c(k+1) = \sigma_c(k+1) - \sigma_c(k) = 0 \tag{21}$$

By combining and simplifying equations (20) and (21), we obtain:

$$\begin{aligned}\sigma_c(k) &= \lambda_1\mathcal{L}_{\mathcal{B}}(O_d(k+1) - O(k) \\ &-\hat{\Theta}(k)\Delta u(k)) + \lambda_2\mathcal{L}_{\mathcal{B}}\Delta\varrho_c(k)\end{aligned} \tag{22}$$

Lemma 1 [25]: Provided that the MAS topology includes at least one spanning tree, the matrix $\mathcal{L}_{\mathcal{B}}$ possesses the invertibility property.

By employing Lemma 1 and (22), the equivalent control law for the MAS can be determined as follows:

$$\Delta u_{eq}(k) = \hat{\theta}^{-1}(k)[O_d(k+1) - O(k) \quad (23)$$
$$-\varrho_c(k) - \lambda_1^{-1}\lambda_2\big(\Delta\varrho_c(k) - \Delta\varrho_c(k-1)\big)]$$

The switching function can be designed by utilizing the variations of the sliding surface with respect to the previous step. The sliding mode control switching function, designed to maintain the system on the sliding surface and ensure stability, is defined as follows:

$$\Delta u_{sw}(k) = \Delta\sigma_c(k) \quad (24)$$
$$= -\hat{\theta}^{-1}(k)\lambda_3 sign\big(\sigma_c(k)\big)$$

where, $\lambda_3 > 0$ is a design parameter that is optimized and determined using the RL algorithm. $sign(\cdot)$ denotes the sign function, and $sign\big(\sigma_c(k)\big) = \left[sign\left(\sigma_{c_1}^T(k)\right), \dots, sign\left(\sigma_{c_N}^T(k)\right)\right]^T$.

By combining the equivalent control function (23) with the switching function (24), the following sliding mode control law is obtained:

$$\Delta u(k) = \Delta u_{eq}(k) + \Delta u_{sw}(k) \quad (25)$$
$$= \hat{\theta}^{-1}(k)[O_d(k+1) - O(k) - \varrho_c(k)$$
$$-\lambda_1^{-1}\lambda_2\big(\Delta\varrho_c(k) - \Delta\varrho_c(k-1)\big)$$
$$-\lambda_3 sign\big(\sigma_c(k)\big)]$$

Since the reference signal is predetermined, $O_d(k+1)$ is known, making (25) implementable.

Stability Proof: A formal stability analysis of the proposed MF-DSMC is conducted employing a Lyapunov function. The Lyapunov function employed in this analysis is positive semi-definite, and is defined by:

$$V_c(k) = \frac{1}{2}\sigma_c(k)^2 \quad (26)$$

the first-order difference is taken of both sides of (26), and is determined:

$$\Delta V_c(k) = \sigma_c(k)\Delta\sigma_c(k) \quad (27)$$

by substituting (24) into (27) and simplifying, the following result is obtained:

$$\Delta V_c(k) \le -\hat{\theta}^{-1}(k)\lambda_3|\sigma_c(k)| \quad (28)$$

in this work, the sign of $\hat{\theta}(k)$ is determined by its initial condition, which is chosen to be positive based on the system characteristics. The stability of the closed-loop system is ensured by maintaining a constant sign of $\hat{\theta}(k)$ throughout the learning process. Since the main focus of this study is not the estimation design itself, further theoretical details regarding the sign preservation property can be found in [26]. If $\hat{\theta}(0) > 0$, then the inverse of $\hat{\theta}(k)$ is also positive. The first-order difference of (26) yields a negative semi-definite function, indicating the Lyapunov stability of the proposed controller.

4. Proposed Fault Diagnosis System

This section presents the details regarding the design of the fault detector and fault estimator.

4.1 Fault Detection Based on MF-DSMO

The fault detection component is presented in this section. Its operation is as follows: first, the output of each agent is estimated using the MF-DSMO; following this, the deviation between the estimated output and the corresponding measured output for each agent is computed, which is referred to as the residual. If the residual is below the FDT, the system is considered to be operating under normal conditions; otherwise, a fault is indicated, prompting the necessary corrective actions.

Similar to the distributed control design section, the error between the agents' estimates is given by:

$$\varrho_{p_{ij}}(k) = \hat{O}_j(k) - \hat{O}_i(k) \quad (29)$$

where, $\hat{O}$ represents the measured output estimate of the agent i. The error between the target agent and the agents' estimates is given by:

$$\varrho_{p_i}(k) = O(k) - \hat{O}_i(k) \quad (30)$$

Considering the mutual interactions among agents, the distributed estimation scheme is determined based on the weighted sum of the error between each agent's output estimate and the leader's output estimate, and the mistake among the agents' estimates themselves, as defined below:

$$\mathcal{E}_{p_i}(k) = \sum_{j\in N} a_{ij}\left(\hat{O}_j(k) - \hat{O}_i(k)\right) + \mathcal{b}_i\left(O(k) - \hat{O}_i(k)\right) \quad (31)$$

as in the controller design section, the Laplacian matrix $\mathcal{L}_{\mathcal{B}}$ is calculated and determined.

The distributed estimation algorithm is derived using the Laplacian matrix as follows:

$$\mathcal{E}_p(k) = \big((\mathcal{L} + \mathcal{B})\otimes I_n\big)\varrho_p(k) = \mathcal{L}_{\mathcal{B}}\varrho_p(k) \quad (32)$$

the error vector $\varrho_p(k) = O(k) - \hat{O}(k) = [\varrho_1^T(k), \dots, \varrho_N^T(k)]^T$, where $, O(k) = [O^T(k), \dots, O^T(k)]^T \in \mathbb{R}^{n\times N}$ represents the actual output, and $\hat{O}(k) = \left[\hat{O}_1^T(k), \dots, \hat{O}_N^T(k)\right]^T$ denotes the estimated output.

The variation in the error of the distributed estimation algorithm with respect to the previous step is given by:

$$\Delta\mathcal{E}_p(k) = \mathcal{E}_p(k) - \mathcal{E}_p(k-1) \quad (33)$$
$$= \mathcal{L}_{\mathcal{B}}\varrho_p(k) - \mathcal{L}_{\mathcal{B}}\varrho_p(k-1) = \mathcal{L}_{\mathcal{B}}\Delta\varrho_p(k)$$

where $\Delta\varrho_p(k) = \varrho_p(k) - \varrho_p(k-1)$. The sliding surface for the MF-DSMO is given by:

$$\sigma_p(k) = \lambda_4\mathcal{E}_p(k-1) + \lambda_5\Delta\mathcal{E}_p(k) \quad (34)$$

where, $\lambda_4, \lambda_5 > 0$ represent the controller design parameters, which are determined and adjusted through the RL optimization algorithm. Additionally, $\sigma_p(k) = \left[\sigma_{p_1}^T(k), \dots, \sigma_{p_N}^T(k)\right]^T$.

To implement the control strategy, the sliding surface for the next step must be computed as follows:

$$\sigma_p(k+1) = \lambda_4\mathcal{E}_p(k) + \lambda_5\Delta\mathcal{E}_p(k+1) \quad (35)$$

$$= \lambda_4 \mathcal{L}_{\mathcal{B}} \varrho_p(k) + \lambda_5 \mathcal{L}_{\mathcal{B}} \Delta \varrho_p(k+1)$$
$$= \lambda_4 \mathcal{L}_{\mathcal{B}} \varrho_p(k) + \lambda_5 \mathcal{L}_{\mathcal{B}} \left(\varrho_p(k+1) - \varrho_p(k) \right)$$
$$= \lambda_4 \mathcal{L}_{\mathcal{B}} \varrho_p(k) + \lambda_5 \mathcal{L}_{\mathcal{B}} \left(O(k+1) - \hat{O}(k+1) - \varrho_p(k) \right)$$
$$= \lambda_4 \mathcal{L}_{\mathcal{B}} \varrho_p(k) + \lambda_5 \mathcal{L}_{\mathcal{B}} (O(k) + \hat{\Theta}(k) \Delta u(k) - \hat{O}(k+1)) - \varrho_p(k)$$

The reaching law used in designing the MF-DSMO can be expressed as:

$$\sigma_p(k+1) - \sigma_p(k) = 0 \tag{36}$$

by combining (35) and (36), it follows that:

$$\sigma_p(k) = \lambda_4 \mathcal{L}_{\mathcal{B}} \varrho_p(k) + \lambda_5 \mathcal{L}_{\mathcal{B}} (O(k) + \hat{\Theta}(k) \Delta u(k) - \hat{O}(k+1)) - \varrho_p(k) \tag{37}$$

by simplifying (37) and applying Lemma 1 under the invertibility condition of $\mathcal{L}_{\mathcal{B}}$, this yields:

$$\hat{O}(k+1) = O(k) + \hat{\Theta}(k) \Delta u(k) + (\lambda_5^{-1} \lambda_4 - 1) \varrho_p(k) - \lambda_5^{-1} \mathcal{L}_{\mathcal{B}}^{-1} \left(\sigma_p(k) \right) \tag{38}$$
$$= O(k) + \hat{\Theta}(k) \Delta u(k) + (\lambda_5^{-1} \lambda_4 - 1) \varrho_p(k) - \lambda_5^{-1} \mathcal{L}_{\mathcal{B}}^{-1} \left(\lambda_4 \mathcal{L}_{\mathcal{B}} \varrho_p(k-1) + \lambda_5 \mathcal{L}_{\mathcal{B}} \Delta \varrho_p(k) \right)$$
$$= O(k) + \hat{\Theta}(k) \Delta u(k) + \lambda_5^{-1} \lambda_4 \Delta \varrho_p(k) - \varrho_p(k-1)$$

To improve implementation accuracy, the variable transformation is performed as follows:

$$\hat{O}(k+1) = \hat{O}(k) + \hat{\Theta}(k) \Delta u(k) + \lambda_5^{-1} \lambda_4 \Delta \varrho_p(k) - \varrho_p(k-1) \tag{39}$$

The switching function used to maintain the sliding surface in the proposed MF-DSMO observer is formulated as follows:

$$\Delta \sigma_{p_{sw}}(k) = \Delta \sigma_p(k) = -\lambda_6 sign \left(\sigma_p(k) \right) \tag{40}$$

where $sign \left(\sigma_p(k) \right) = \left[sign \left({\sigma_p}_1^T(k) \right), \dots, sign \left({\sigma_p}_N^T(k) \right) \right]^T$. Finally, the desired estimation is obtained from the output of the corresponding agent using the following relation:

$$\hat{O}(k+1) = \hat{O}(k) + \hat{\Theta}(k) \Delta u(k) + \lambda_5^{-1} \lambda_4 \Delta \varrho_p(k) - \varrho_p(k-1) - \lambda_6 sign \left(\sigma_p(k) \right) \tag{41}$$

The residual signal, which reflects the presence of a fault in the controlled system, performs as follows:

$$r(k+1) = O(k+1) - \hat{O}(k+1) \tag{42}$$

where $r(k+1) = [r_1^T(k+1), \dots, r_N^T(k+1)]^T$. The fault detection mechanism identifies fault emergence in the system by comparing equation (42) with equation (7) as follows:

$$\begin{cases} r(k+1) \le \delta(k+1) \ \ fault\ occurrence \\ r(k+1) > \delta(k+1) \ \ fault\ absence \end{cases} \tag{43}$$

The stability proof of the MF-DSMO observer is presented as follows.

Stability Analysis of MF-DSMO: The proposed Lyapunov function is assumed to be positive semi-definite, defined as follows:

$$V_p(k) = \frac{1}{2} \sigma_p(k)^2 \tag{44}$$

by computing the first-order difference of both sides of equation (44), it follows that:

$$\Delta V_p(k) = \sigma_p(k) \Delta \sigma_p(k) \tag{45}$$

by substituting equation (40) into (45) and simplifying, it follows that:

$$\Delta V_p(k) \le -\lambda_6 \left| \sigma_p(k) \right| \tag{46}$$

a negative semi-definite function is obtained, demonstrating that the proposed MF-DSMO observer is Lyapunov stable.

4.2 Novel Fault Estimation Based on MF-DFOSME

A fault estimator based on an MF-DFOSME is presented in this section. A fault is described as an event that, when it occurs, leads to a deviation or disruption in the system's optimal performance. Therefore, promptly and accurately determining the extent and severity of a fault can help avoid severe adverse effects and enable the application of suitable corrective actions.

The estimation of the measured output with a sensor fault is defined as follows:

$$\hat{O}_{\mathcal{F}_i}(k+1) = \hat{O}_{\mathcal{F}_i}(k) + \hat{\Theta}_i(k) \Delta u_i(k) + \hat{\mathcal{F}}_i(k) \tag{47}$$

where $\hat{O}_{\mathcal{F}}(k) = \left[\hat{O}_{\mathcal{F}_1}^T(k), \dots, \hat{O}_{\mathcal{F}_N}^T(k) \right]^T$. Analogous to the previous section, the difference between the estimates of the faulty components is formulated as:

$$\varrho_{\mathcal{F}_{ij}}(k) = \hat{O}_{\mathcal{F}_j}(k) - \hat{O}_{\mathcal{F}_i}(k) \tag{48}$$

and the error between the faulty factor and its estimate is formulated as:

$$\varrho_{\mathcal{F}_i}(k) = O(k) - \hat{O}_{\mathcal{F}_i}(k) \tag{49}$$

Taking into account the mutual interactions among the factors, the distributed estimation scheme is formulated based on a weighted combination of the error between the estimated outputs of the faulty factors and that of the leader factor, along with the output error among the estimates of the faulty factors themselves, as follows:

$$\mathcal{E}_{\mathcal{F}_i}(k) = \sum_{j \in N} a_{ij} \left(\hat{O}_{\mathcal{F}_j}(k) - \hat{O}_{\mathcal{F}_i}(k) \right) + \mathcal{b}_i \left(O(k) - \hat{O}_{\mathcal{F}_i}(k) \right) \tag{50}$$

where, $O(k)$ denotes the leader factor, and, as in the controller design section, the Laplacian matrix $\mathcal{L}_{\mathcal{B}}$ is calculated and specified.

The distributed estimation algorithm is derived using the Laplacian matrix as follows:

$$\mathcal{E}_{\mathcal{F}_i}(k) = \big((\mathcal{L}+\mathcal{B})\otimes I_n\big)\varrho_{\mathcal{F}}(k) = \mathcal{L}_{\mathcal{B}}\varrho_{\mathcal{F}}(k) \quad (51)$$

Given the absence of information about the sensor fault function, the estimation error is obtained using equations (4) and (47) as follows:

$$\begin{aligned}\varrho_{\mathcal{F}}(k+1) &= O(k+1) - \hat{O}_{\mathcal{F}}(k+1) \quad (52)\\ &= O(k) + \hat{\theta}(k)\Delta u(k)\\ &-\Big(\hat{O}_{\mathcal{F}}(k) + \hat{\theta}(k)\Delta u(k) + \hat{\mathcal{F}}(k)\Big)\end{aligned}$$

since the actual measured output is identical to the estimation output $\big(O(k) = \hat{O}_{\mathcal{F}}(k)\big)$, (52) can be simplified as follows:

$$\varrho_{\mathcal{F}}(k+1) = -\hat{\mathcal{F}}(k) \quad (53)$$

where $e_{\mathcal{F}}(k) = -\hat{\mathcal{F}}(k) = \big[\varrho_{\mathcal{F}_1}^T(k), \dots, \varrho_{\mathcal{F}_N}^T(k)\big]^T$, $\hat{\mathcal{F}}(k) = \big[\hat{\mathcal{F}}_1^T(k), \dots, \hat{\mathcal{F}}_N^T(k)\big]^T$.

The change in the error of the distributed estimation algorithm with respect to the previous iteration is described as follows:

$$\begin{aligned}\Delta\mathcal{E}_{\mathcal{F}}(k) &= \mathcal{E}_{\mathcal{F}}(k) - \mathcal{E}_{\mathcal{F}}(k-1) \quad (54)\\ &= \mathcal{L}_{\mathcal{B}}\varrho_{\mathcal{F}}(k) - \mathcal{L}_{\mathcal{B}}\varrho_{\mathcal{F}}(k-1) = \mathcal{L}_{\mathcal{B}}\Delta\varrho_{\mathcal{F}}(k)\end{aligned}$$

where $\Delta\varrho_{\mathcal{F}}(k) = \varrho_{\mathcal{F}}(k) - \varrho_{\mathcal{F}}(k-1)$.

The sliding surface designed for MF-DFOSME, utilizing the GL fractional calculus, is expressed as follows:

$$\begin{aligned}\sigma_{\mathcal{F}}(k) &= \lambda_7\mathcal{E}_{\mathcal{F}}(k) + \lambda_8\Delta^{\alpha}\mathcal{E}_{\mathcal{F}}(k-1) \quad (55)\\ &= \lambda_7\mathcal{L}_{\mathcal{B}}\varrho_{\mathcal{F}}(k) + \lambda_8\mathcal{L}_{\mathcal{B}}\Delta^{\alpha}\varrho_{\mathcal{F}}(k-1)\end{aligned}$$

where $\lambda_7, \lambda_8 > 0$ are the design parameters of the fault estimator, computed and tuned by the RL optimization algorithm, and $\sigma_{\mathcal{F}}(k) = \big[\sigma_{\mathcal{F}_1}^T(k), \dots, \sigma_{\mathcal{F}_N}^T(k)\big]^T$.

In designing the estimator, it is also necessary to calculate the sliding surface for the next time step, which yields the following:

$$\sigma_{\mathcal{F}}(k+1) = \lambda_7\mathcal{L}_{\mathcal{B}}\varrho_{\mathcal{F}}(k+1) + \lambda_8\mathcal{L}_{\mathcal{B}}\Delta^{\alpha}\varrho_{\mathcal{F}}(k) \quad (56)$$

The reaching law used in the design of the MF-DFOSME is defined as follows:

$$\sigma_{\mathcal{F}}(k+1) - \sigma_{\mathcal{F}}(k) = 0 \quad (57)$$

by combining (56) and (57), the following is obtained:

$$\sigma_{\mathcal{F}}(k) = \lambda_7\mathcal{L}_{\mathcal{B}}\Big(-\hat{\mathcal{F}}(k)\Big) + \lambda_8\mathcal{L}_{\mathcal{B}}\Delta^{\alpha}\varrho_{\mathcal{F}}(k) \quad (58)$$

by simplifying equation (58) and assuming that $\mathcal{L}_{\mathcal{B}}$ is invertible, we obtain:

$$\begin{aligned}\hat{\mathcal{F}}(k) &= \lambda_7^{-1}\lambda_8\Delta^{\alpha}\varrho_{\mathcal{F}}(k) - \lambda_7^{-1}\mathcal{L}_{\mathcal{B}}^{-1}\big(\sigma_{\mathcal{F}}(k)\big) \quad (59)\\ &= \lambda_7^{-1}\lambda_8\Delta^{\alpha}\varrho_{\mathcal{F}}(k) - \lambda_7^{-1}\mathcal{L}_{\mathcal{B}}^{-1}(\lambda_7\mathcal{L}_{\mathcal{B}}\varrho_{\mathcal{F}}(k)\\ &+\lambda_8\mathcal{L}_{\mathcal{B}}\Delta^{\alpha}\varrho_{\mathcal{F}}(k-1))\\ &= \lambda_7^{-1}\lambda_8\Delta^{\alpha}\varrho_{\mathcal{F}}(k) - \lambda_7^{-1}\lambda_8\Delta^{\alpha}\varrho_{\mathcal{F}}(k-1)\\ &-\varrho_{\mathcal{F}}(k)\end{aligned}$$

The switching function designed to keep the system on the sliding surface is defined as follows:

$$\Delta\sigma_{\mathcal{F}_{sw}}(k) = -\lambda_9 sign\big(\sigma_{\mathcal{F}}(k)\big) \quad (60)$$

where, $\lambda_9 > 0$ is the fault estimator design parameter, which is tuned and optimized through RL, and $sign\big(\sigma_{\mathcal{F}}(k)\big) = \Big[sign\Big(\sigma_{\mathcal{F}_1}^T(k)\Big), \dots, sign\Big(\sigma_{\mathcal{F}_N}^T(k)\Big)\Big]^T$.

Finally, the sensor fault occurring in the agents is estimated as follows:

$$\begin{aligned}\hat{\mathcal{F}}(k) &= \lambda_7^{-1}\lambda_8\big(\Delta^{\alpha}\varrho_{\mathcal{F}}(k) - \Delta^{\alpha}\varrho_{\mathcal{F}}(k-1)\big) \quad (61)\\ &-\varrho_{\mathcal{F}}(k) - \lambda_9 sign\big(\sigma_{\mathcal{F}}(k)\big)\end{aligned}$$

ultimately, the fault in each agent is determined.

The above derivation shows that the fault estimation law is obtained directly from the sliding surface dynamics and reaching condition. The key assumption in (53) corresponds to zero initial estimation error, which is commonly adopted in sliding-mode estimation frameworks to simplify analysis without loss of generality.

5. The RL optimization algorithm

In this study, an RL toolbox in Matlab 2023b is employed to optimize the tuning parameters of the fault detector, controller, and fault estimator. The objective of the controller is to reduce the tracking error. In the fault detector, the aim is to minimize the error between the measured output and its estimated counterpart; the objective is to minimize the estimation error of the sensor fault signal.

The agent is based on the Deep Deterministic Policy Gradient (DDPG) algorithm. The observation space consists of system error signals, while the action space includes the tunable parameters λ_1 to λ_9. The reward function is defined as $reward = -\sum|error|$, aiming to minimize tracking, estimation, and fault errors. The main hyperparameters include 156 neurons in the hidden layer, a discount factor of 0.98, a learning rate of 0.014, and a maximum of 288 episodes. The final optimized parameter values are reported in the simulation section. the condition is determined by the termination flag, activated if the system output (O) is outside the bounds $[\mathcal{H}, \mathcal{U}]$ [27]. The adopted approach ensures error reduction.

The parameters $\lambda_1, \dots, \lambda_9$are obtained through RL-based optimization with respect to the specific reference trajectory and fixed MAS topology considered in this study. While the learned parameters ensure satisfactory performance under the trained scenario, significant changes in the reference signal or system topology may affect optimality. In such cases, re-training or online adaptation of the RL agent may be required to preserve optimal performance. Nevertheless, due to the stability of the underlying control framework, the obtained parameters may still provide acceptable performance under moderate variations in operating conditions.

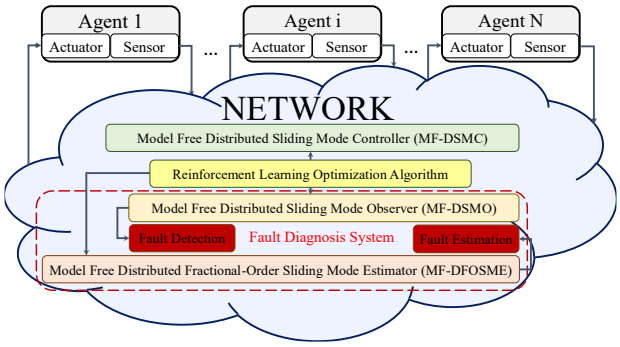

Fig. 1. The overall block diagram of the proposed methods.

6. Simulation Results and Discussion

This section evaluates the effectiveness of the proposed algorithms via numerical simulations. The comparative analysis is structured as follows: Section 6.1 contrasts the proposed MF-DSMC algorithm with the distributed MFAC technique, and Section 6.2 compares the proposed FDS with methods [28].

The simulation utilizes the MAS configuration illustrated in Fig. 1, which presents the overall architecture of the proposed control and fault diagnosis method. The MAS adopted from [29] is employed here, and the data-driven model of each agent is described by:

$$O_i(k+1) = \vartheta_i u_i(k) + \frac{O_i(k)u_i(k)}{1 - O_i^{\beta_i}(k)} + \mathcal{F}(k) \tag{62}$$

in this setting, $\mathcal{F}(K)$ corresponds to the fault affecting each agent, and the parameter values are selected as:

$$\vartheta_i = 1, \quad \beta_i = 2, \qquad i = 1,2,3 \tag{63}$$

This setup ensures that the agents share the same nominal dynamics.

The agents are initialized with the following output values $O(0) = [-1,1,2]^T$, the initial PPD value is $\hat{\Theta}_i(0) = 1.5$, and the initial control signal is $u_i(0) = 0$.

The reference output is given by:

$$O_d(k) = 0.5 + 0.25sin(0.0314k) + 0.3cos(0.0628k) \tag{64}$$

The Laplacian matrix below depicts the communication topology among the agents:

$$\mathcal{L}_{\mathcal{B}} = \begin{bmatrix} 3 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \tag{65}$$

To ensure a fair and unbiased comparison, the benchmark methods adopted from [28] and [29] were not implemented using their nominal parameter settings. Instead, the tunable parameters of these methods were first identified according to their original formulations. Subsequently, the same RL–based optimization procedure employed for the proposed controller, observer, and fault detector was applied to these benchmark approaches under identical simulation conditions. This procedure allowed the extraction of optimized parameter sets for all competing methods using a unified optimization framework. Therefore, all controllers, estimators, and fault detection schemes were evaluated using RL-tuned parameters, ensuring that the comparative results reflect the intrinsic performance of the methods rather than differences in parameter tuning.

In the simulations, the memory length of the GL fractional difference was set to $L = 100$, which provides a suitable trade-off between computational efficiency and estimation accuracy for the considered data length ($k = 400$).

6.1 Analysis of the Proposed MF-DSMC Controller

In this section, we evaluate how the MF-DSMC controller, tuned via the RL algorithm, compares to the MFAC algorithm [29]. It is well known that conventional SMC suffers from the chattering phenomenon due to the discontinuous nature of the control law. Since the proposed RL-based optimization framework does not include an explicit penalty on control effort or switching activity, a certain level of chattering may appear in the control signal. Nevertheless, the RL algorithm ensures bounded and stable system behaviour by appropriately tuning the control parameters $\lambda_1, \dots, \lambda_3$ under the defined performance index. Chattering reduction techniques can be incorporated in future extensions of this work for industrial-scale implementations. The set of parameters chosen for the proposed controller is given by: $\upsilon = 1$, $\rho = 1$, $\lambda_1 = 1$, $\lambda_2 = .13$, $\lambda_3 = 0.002$. For agent i, the MF-DAC control law is formulated as follows:

$$\Delta u_i(k) = \frac{\xi \hat{\Theta}_i(k)}{1 + \hat{\Theta}_i(k)^2} \mathcal{E}_{c_i}(k) \tag{66}$$

where $\xi = 0.5$.

Fig. 2 presents a comparison of the tracking performance between the proposed MF-DSMC and the MFAC methods. It can be observed that the agents' outputs converge to the reference signal quickly. The tracking error between the agents and the reference signal gradually approaches zero over time. Moreover, the error between agents remains negligibly small. As shown in Fig. 3, the MFAC exhibits a larger tracking error, highlighting its comparatively lower performance relative to the MF-DSMC proposed controller. Fig. 4 illustrates the control signals generated by both the proposed controller and the MFAC. As illustrated in the figure, the proposed MF-DSMC controller produces control signals with lower amplitude than the MFAC, suggesting reduced energy consumption during tracking of the desired output. To facilitate a quantitative comparison among the methods, the monitoring Root Mean Square Error (RMSE) of each approach is presented in Table I.

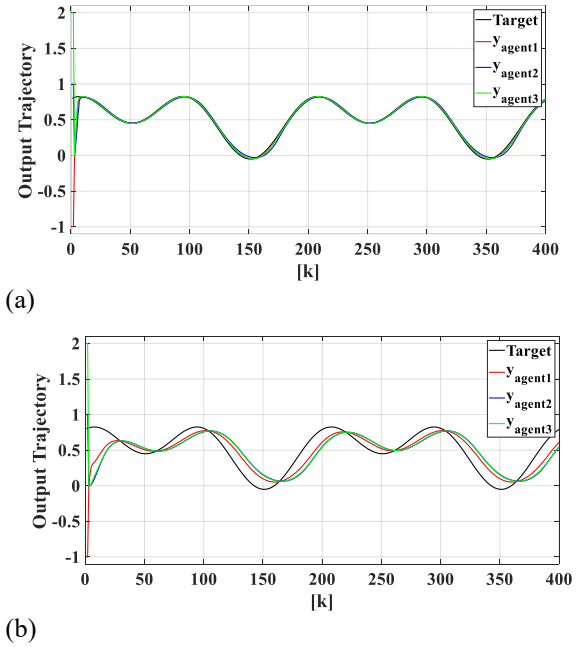

Fig. 2. The trajectory tracking results. (a) MF-DSMC method, (b) MFAC method.

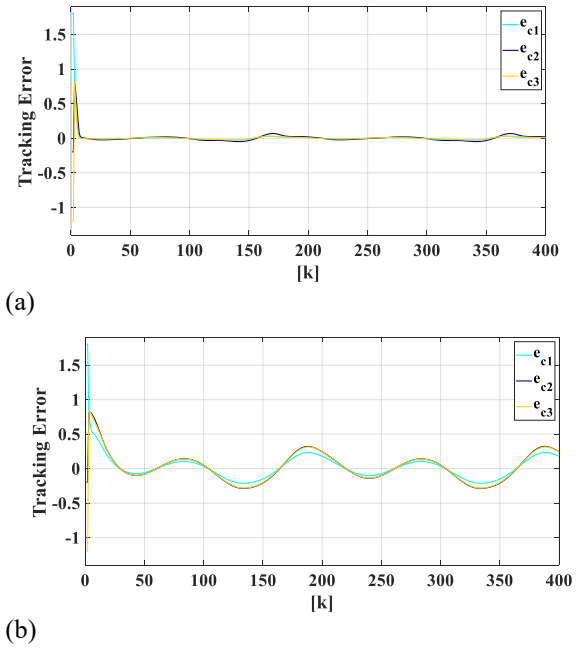

Fig. 3. The agents' tracking error results. (a) MF-DSMC method, (b) MFAC method.

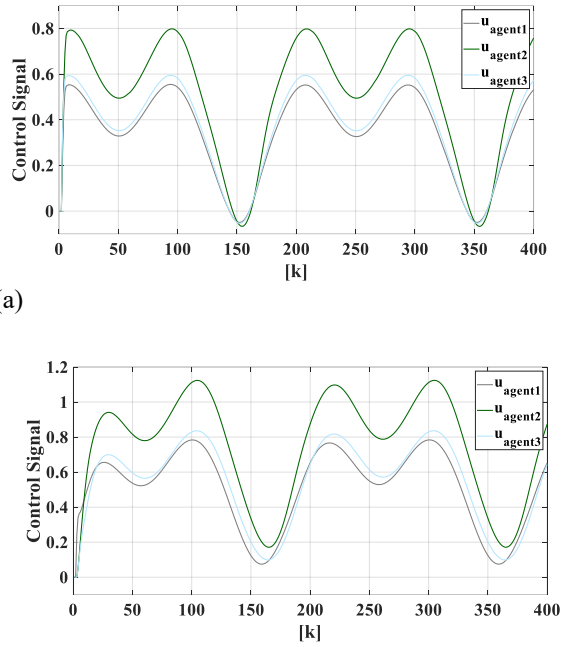

Fig. 4. The control signal results. (a) MF-DSMC method, (b) MFAC method.

6.2 Analysis of the Proposed FDS

The architecture of the proposed FDS is introduced in this section, encompassing both a fault estimator and a fault detector. The sensor fault scenario implemented in the MAS is defined as follows:

$$\mathcal{F}_i(k) = \begin{cases} 0 & k \le 150 \\ \dfrac{\mathcal{F}_i(k)}{1+\mathcal{F}_i(k)^2} + 0.08cos(0.6k) & k > 150 \end{cases} \quad (67)$$

In this study, the FDT is configured using the parameters $\lambda_4 = 0.84, \lambda_5 = 0.47$, and $\lambda_6 = 0.005$. The final value of FDT is formulated as follows:

$$\delta_\infty(k+1) = 0.05 + exp(-0.1k) \quad (68)$$

the parameters for the FDT are, $\delta(0) = 0.9$ and $\gamma = 0.2$, considered.

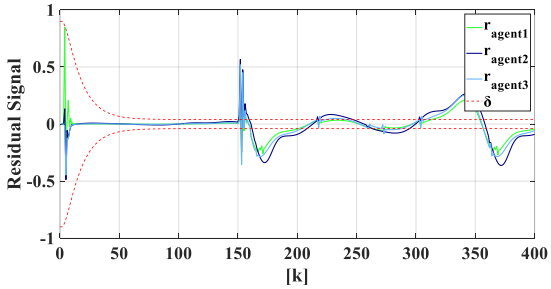

Fig. 5. The Fault detection in agents results.

Fig. 5 illustrates the detection of the fault that occurred in the control system using the MF-DSMO. As shown Fig. 5, faults within the agents are promptly identified with negligible delay, allowing the efficient dissemination of essential diagnostic information from the FDS concurrently.

The fault estimator proposed MF-DSME in this study is compared and evaluated against the [28] method. The parameters of the proposed fault estimator are set to $\lambda_7 = 0.98, \lambda_8 = 0.34, \lambda_9 = 0.001$, and $\alpha = 0.14$ using the RL optimization algorithm. The formulation of the estimator for the i agent is formulated as in [28]:

$$\hat{\mathcal{F}}(k) = \frac{\ell \hat{\Theta}_i^2(k+1)}{\varphi + \hat{\Theta}_i(k+1)} \mathcal{E}_{\mathcal{F}_i}(k) \tag{69}$$

where $\ell = 0.25$, and $\varphi = 1$.

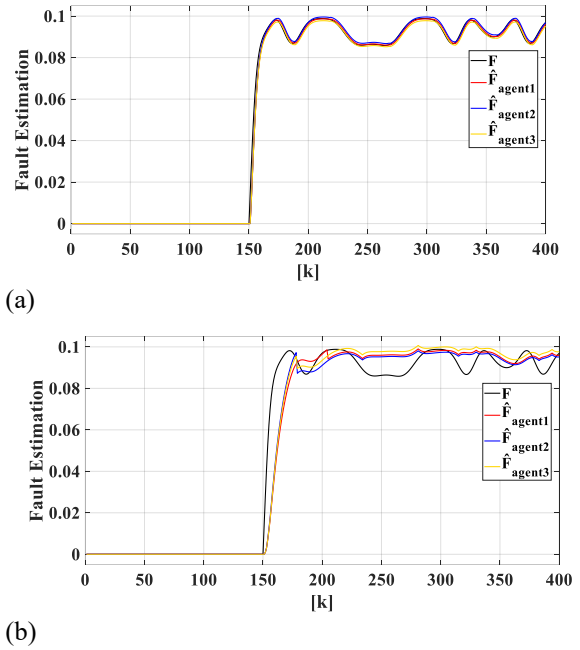

Fig. 6 The Fault estimation in agents results. (a) MF-DSME method, (b) method in reference [28].

The estimation of the sensor fault occurring in the system, obtained using the proposed MF-FODSME, and method in reference [28] is illustrated in Fig. 6.

As illustrated, the proposed MF-FODSME estimation algorithm identifies the sensor fault much faster than the method in reference [28]. The consistently high estimation accuracy across all agents highlights a key advantage of the proposed MF-FODSME. To facilitate a quantitative comparison among the methods, the estimation RMSE of each approach is presented in Table I.

Table I. Quantitative RMSE comparison.

Section6.1	Proposed MF-DSMC	0.1272
	MFAC	0.1517
Section6.2	Proposed MF-DFOSME	0.0039
	Method in [28]	0.0053

7. Conclusion

This study focuses on the challenges of achieving distributed tracking control and performing distributed fault diagnosis in networked, nonlinear MAS when sensor faults are present. A MF-DSMC approach is proposed to address the combined challenges of distributed tracking and fault tolerance under sensor fault conditions. This study further introduces a MF-DSMO for detecting sensor faults, together with a MF-DFOSME to assess the magnitude and severity of faults, providing an integrated approach for both fault diagnosis and fault modelling. According to the simulation results and Table I, the proposed approach demonstrates a 17% reduction in tracking error relative to the MFAC method and a 26% improvement in fault estimation accuracy compared with the method described in reference [28]. Future research will focus on more complex and realistic operational scenarios.

8. Acknowledgments

This work was supported by the Information and Communication Technology Research Institute, Ministry of Information and Communication Technology (grant number 500/5581/P).

9. References

[1] S. J. Oks *et al.*, "Cyber-physical systems in the context of industry 4.0: A review, categorization and outlook," (2024), *Information Systems Frontiers,* vol. 26, no. 5, pp. 1731-1772.

[2] R. Chi, H. Zhang, H. Li, B. Huang, and Z. Hou, "Data-driven dynamic internal model control," (2024), *IEEE Transactions on Cybernetics,* vol. 54, no. 9, pp. 5347-5359.

[3] D. Ding, Q.-L. Han, Z. Wang, and X. Ge, "A survey on model-based distributed control and filtering for industrial cyber-physical systems," (2019), *IEEE Transactions on Industrial Informatics,* vol. 15, no. 5, pp. 2483-2499.

[4] C. Gao, X. He, H. Dong, H. Liu, and G. Lyu, "A survey on fault-tolerant consensus control of multi-agent systems: trends, methodologies and prospects," (2022), *International Journal of Systems Science,* vol. 53, no. 13, pp. 2800-2813.

[5] S. Karnouskos, P. Leitao, L. Ribeiro, and A. W. Colombo, "Industrial agents as a key enabler for realizing industrial cyber-physical systems: Multiagent systems entering industry 4.0," (2020), *IEEE Industrial Electronics Magazine,* vol. 14, no. 3, pp. 18-32.

[6] J. Cederbladh, R. Eramo, V. Muttillo, and P. E. Strandberg, "Experiences and challenges from developing cyber‐physical systems in industry‐academia collaboration," (2024), *Software: Practice and Experience,* vol. 54, no. 6, pp. 1193-1212.

[7] Y.-S. Ma and W.-W. Che, "A hierarchical distributed data-driven adaptive learning control for nonaffine nonlinear MASs," (2024), *IEEE Transactions on Neural Networks and Learning Systems*.

[8] A. Y. R. González *et al.*, "A competitive and profitable multi-agent autonomous broker for energy markets," (2019), *Sustainable Cities and Society,* vol. 49, p. 101590.

[9] J. Zhao, Y. Wang, X. Xi, and G. Wu, "Simulation of steel production logistics system based on multi-agents," (2017), *Int. J. Simul. Model,* vol. 16, no. 1, pp. 167-175.

[10] B. Wang, S. Li, G. Battistelli, L. Chisci, and W. Yi, "Multi-agent fusion with different limited fields-of-view," (2022), *IEEE Transactions on Signal Processing,* vol. 70, pp. 1560-1575.

[11] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," (2017), *Journal of industrial information integration,* vol. 6, pp. 1-10.

[12] W. Liu, M. Zhang, Q. Xu, and L. Xie, "Survey on data-driven control and its application in cyber-physical energy systems," (2025), *Cyber-Physical Energy Systems*.

[13] A. Veisi and H. Delavari, *Applied Data Driven Nonlinear Control*. Hamedan University of Technology Press, (2025).

[14] A. Veisi, M. Shiri, and H. Delavari, "Fractional Order Sliding Mode Observer-Based Control in the Presence of Faults," (2024), *Contributions of Science and Technology for Engineering,* vol. 1, no. 1, pp. 20-25.

[15] A. Villalonga, G. Beruvides, F. Castano, and R. E. Haber, "Cloud-based industrial cyber–physical system for data-driven reasoning: A review and use case on an industry 4.0 pilot line," (2020), *IEEE Transactions on Industrial Informatics,* vol. 16, no. 9, pp. 5975-5984.

[16] J. Zhang, S.-C. Chai, B.-H. Zhang, and G.-P. Liu, "Distributed data-driven tracking control for networked nonlinear MIMO multi-agent systems subject to communication delays," (2021), *Neurocomputing,* vol. 425, pp. 62-70.

[17] F. Celani, M. Heydari, and A. B. Novinzadeh, "Model-Free Adaptive Control for Attitude Stabilization of Earth-Pointing Spacecraft Using Magnetorquers," (2025), *Aerospace,* vol. 12, no. 3, p. 219.

[18] I. Shames, A. M. Teixeira, H. Sandberg, and K. H. Johansson, "Distributed fault detection for interconnected second-order systems," (2011), *Automatica,* vol. 47, no. 12, pp. 2757-2764.

[19] N. Yazdanpanah, M. M. Farsangi, and S. R. Seydnejad, "A data-driven subspace distributed fault detection strategy for linear heterogeneous multi-agent systems," (2024), *ISA transactions,* vol. 146, pp. 186-194.

[20] L. Li, S. X. Ding, C. Wang, M. Zhong, and K. Peng, "A Distributed Semi-Consensus-Based Data-Driven Fault Detection Approach for Dynamic Systems," (2024), *IEEE Transactions on Industrial Informatics*.

[21] Z. Hou and S. Jin, "A novel data-driven control approach for a class of discrete-time nonlinear systems," (2010), *IEEE Transactions on Control Systems Technology,* vol. 19, no. 6, pp. 1549-1558.

[22] Y. Han and M. Hou, "Data-driven second-order sliding mode fault-tolerant control for a class of discrete-time nonlinear systems," (2024), *European Journal of Control,* vol. 76, p. 100954.

[23] M. Soofi, A. Jalilvand, H. Delavari, and S. Mobayen, "H∞ prescribed tracking performance-based fractional-order sliding mode control for disturbed discrete-time systems: an LMI approach," (2025), *International Journal of Systems Science,* vol. 56, no. 5, pp. 1043-1059.

[24] A. Veisi and H. Delavari, "Deep reinforcement learning optimizer based novel Caputo fractional order sliding mode data driven controller," (2025), *Engineering Applications of Artificial Intelligence,* vol. 140, p. 109725.

[25] S. Khoo, L. Xie, and Z. Man, "Robust finite-time consensus tracking algorithm for multirobot systems," (2009), *IEEE/ASME transactions on mechatronics,* vol. 14, no. 2, pp. 219-228.

[26] A. Khaki-Sedigh, *An Introduction to Data-Driven Control Systems*. John Wiley & Sons, (2023).

[27] A. K. Shakya, G. Pillai, and S. Chakrabarty, "Reinforcement learning algorithms: A brief survey," (2023), *Expert Systems with Applications,* vol. 231, p. 120495.

[28] Q. Zhu and C.-X. Shi, "Data-driven fault-tolerant consensus control for nonlinear multi-agent systems under sensor and actuator faults," (2025), *International Journal of Systems Science,* pp. 1-18.

[29] S. S. A. Sahafi and M. M. Farsangi, "Fully distributed data-driven model-free adaptive control for consensus tracking in multi-agent systems," (2025), *ISA transactions,* vol. 158, pp. 122-129.