International Journal of
Network Dynamics and Intelligence (IJNDI)

Scilight

Check for updates

Article

Adaptive Event-Triggered Fault-Tolerant Control for Networked Multi-Agent Systems under Actuator and Sensor Faults

Wei-Rong Wang [1], Ji Zhang [1,*], Guo-Ping Liu [2], and Hao-Hua Lv [1]

1 School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266525, China
2 School of Automation and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen 518055, China
* Correspondence: zhangji@qut.edu.cn
Received: 30 December 2025; Revised: 19 March 2026; Accepted: 31 March 2026; Published: 15 June 2026

Abstract: This paper proposes a networked active fault-tolerant predictive control strategy based on an adaptive event-triggered mechanism for networked multi-agent systems subject to simultaneous actuator and sensor faults. First, a state and fault estimator is designed to estimate the system states as well as the actuator and sensor faults under the double-fault scenario. Second, an adaptive event-triggered mechanism is developed to reduce the communication frequency and alleviate the communication burden of the system. Then, a fault-tolerant control law is constructed to achieve active fault-tolerant control for multi-agent systems with double faults. Finally, a delay compensator is introduced to address the network-induced delay problem. The proposed strategy effectively compensates for communication delays and reduces the occupation of network resources.

Keywords: event-triggered mechanism; fault-tolerant control; networked delay; multi-agent system; networked control

1. Introduction

Nowadays, with the rapid development of network control [1,2] and distributed control [3] technologies, multi-agent systems (MAS) are increasingly used in fields such as autonomous driving, robot clusters, space satellites, and formation control [4]. Real systems composed of multiple synergistic [5] agents are inevitably subject to problems such as faults, communication delays, and bandwidth constraints, which may lead to degraded or even unstable system performance. In particular, for networked MAS involving frequent information exchange, these adverse effects become more pronounced under simultaneous actuator and sensor [6,7] fault conditions. Therefore, under the networked delay, ensuring the stability and reliability of MAS and reducing the system communication load becomes the core of current research.

Most of the current fault-tolerant control (FTC) [8] methods mainly focus on single-fault scenarios. For example, in [9], the FTC problem of MAS under sensor faults is investigated using a distributed fault-tolerant tracking control approach based on a Kalman filter estimator. In [10], a cooperative FTC scheme based on a dynamic surface control technique is proposed for multi-quadrotor formation systems subject to sensor faults. FTC for hybrid actuator faults under nonlinear systems is investigated in [11], where a distributed coherent cooperative FTC is designed based on robust adaptive fuzzy techniques. Moreover, the FTC of MAS subject to actuator faults and external disturbances is studied in [12], in which an adaptive gain compensation control law is developed to achieve the fault-tolerant consistency problem. In [13], the coherent tracking problem of MAS with input-quantized actuator faults is investigated by introducing Young's inequality and Gaussian function properties. A dynamic control scheme with adaptive neural networks is proposed to ensure consensus among followers. Based on these studies, FTC techniques are further developed to enhance system performance under sensor or actuator fault conditions.

Event-triggered mechanisms (ETM) [14,15] are widely used to reduce communication burden and efficiently conserve network resources. By introducing an auxiliary function to dynamically adjust the triggering threshold and regulate the frequency of information transmission, ETMs effectively improve system operational efficiency [16]. Existing studies mainly investigate three types of triggering mechanisms: static ETM [17], dynamic ETM [18], and adaptive ETM [19]. In addition, an event-triggered fault-tolerant leader-follower control strategy for MAS is proposed in [20], where adaptive control techniques and Lyapunov functions are employed to guarantee system stability. For cooperative output regulation of heterogeneous MAS under switching topology [21,22], designed event-triggered output regulation control mechanisms based on ETM, respectively. Moreover, in [23] studies the ETM consisting of MAS under nonlinear network attacks, which generalizes linear network attacks to a class of nonlinear under which corresponding security control protocols are designed.

The presence of communication delays in a networked control system can also impact the overall system. Based on this situation [24,25], propose a fault-tolerant design method for fuzzy network time-lag system controllers; respectively, the stability is verified with the help of Lyapunov function theory and linear matrix inequality methods. In addition, for the time lag phenomenon under communication networks [26], the problem of estimating the final bounded state of a time lag system in a half-duplex relay communication network, and an adaptive saturation mechanism is used to construct the state estimator.

The above literature mainly focuses on single sensor or actuator faults, not consider synergistic compensation mechanisms when two types of faults coexist or overlap, which limits the adaptability of existing algorithms in complex scenarios; Moreover, the fault estimator design does not take into account bandwidth and delay constraints, in actual network-constrained systems, the performance of state and fault estimation is significantly degraded under inconsistent time delay and sampling rate, and most of the methods are designed under the assumption of ideal communication. Although there are separate fault-tolerant or predictive control [27] in the literature, the triple coupling framework of active fault tolerance and prediction, as well as ETM [28] has not yet been established, as a result, in an inability to balance resource utilization and steady-state accuracy, which limits the algorithms ability to adapt in complex scenarios.

Based on the above description, this paper primarily focuses on the design of an active fault-tolerant predictive control strategy utilizing an ETM for networked MAS in the event of simultaneous actuator and sensor faults. The main contributions of this paper are summarized as follows:

(1) A networked MAS in the presence of random faults in actuators and sensors is constructed, and an observer-based fault estimator is designed to estimate double faults.
(2) Introducing an ETM to filter input signals and utilizing auxiliary functions to dynamically adjust thresholds to reduce communication frequency, thereby reducing the bandwidth burden.
(3) Design a network delay compensator to effectively compensate for the impact of network delay on the system.

The rest of the article consists of Section 2 describes the basics covered in this article and the issues that are the focus of the discussion; Section 3 presents the primary methods of fault estimators, ETM, FTC and delay compensators, and stability analysis of networked MAS; In the Section 4, simulation experiments are conducted to demonstrate the feasibility of the proposed method; finally, the overall research of the paper is summarized in the Section 5.

2. Description of the Problem

MAS are interlinked through networks to exchange information between agents. The communication topology between the agents in the system is denoted by $G = (V, E, A)$, where $V = \{v_1, v_2, \dots, v_N\}$ denotes the agent vertex set of the agent and $E \in V \times V$ denotes the edge set. The expression $G = \text{diag}\left\{g_1, g_2, \dots, g_N\right\}$ denotes the adjacency matrix between leaders and followers in the system; if $g_i > 0$, it means that the follower *i* can receive the signal from the leader; otherwise $g_i = 0$. The adjacency matrix of the follower is denoted as $A = [a_{ij}] \in R^{n\times n}$, if $(v_i, v_j) \in E$, then $a_{ij} = 1$, indicating that the node v_j can receive signals from the node v_i; if $(v_i, v_j) \notin E$, then $a_{ij} = 0$, indicating that the node signal could not be acquired. Define the Laplace matrix of a topological graph G a $L(G) = D(G) - A(G) = \left[l_{ij}\right]$ a diagonal matrix $D = \text{diag}\left(\sum_{j=1}^{N} a_{ij}\right)$, when $i = j, l_{ij} = \sum_{j=1}^{N} a_{ij}$; when $i \neq j, l_{ij} = -a_{ij}$.

Here, consider a MAS consisting of 1 leader and N followers, where random fault situations may occur in the actuators and sensors of each agent, and communication in the networked control system may be delayed or packet lost. In a discrete linear network system, the dynamic equations for each agent are represented as follows:

$$\begin{gathered} x_i(k+1) = Ax_i(k) + Bu_i(k) \\ y_i(k) = Cx_i(k) \end{gathered} \tag{1}$$

where $x_i \in R^n, u_i \in R^m, y_i \in R^p$ denotes the state, control input, and output quantities of agent *i*, respectively; A, B, C are system matrices with corresponding dimensions.

The leader in a MAS is denoted as *r*, and its dynamic equation is as follows:

$$\begin{gathered} x_r(k+1) = Ax_r(k) \\ y_r(k) = Cx_r(k) \end{gathered} \tag{2}$$

where $x_r(k) \in R^n, y_r(k) \in R^p$ denotes the state and output quantities of the leader, respectively.

In the case of a possible random fault of both actuators and sensors, the equations for the dynamics of a MAS can be rewritten as:

$$\begin{gathered} x_i(k+1) = Ax_i(k) + Bu_i(k) + Ef_{a,i}(k) \\ y_i(k) = Cx_i(k) + Ff_{s,i}(k) \end{gathered} \tag{3}$$

where $f_{a,i}(k)$ and $f_{s,i}(k)$ denote actuator and sensor faults, respectively, E and F both are system matrices with corresponding dimensions.

Therefore, the expression of the fault augmentation system can be derived:

$$\begin{gathered} x_{f,i}(k+1) = A_f x_{f,i}(k) + B_f u_i(k) \\ y_i(k) = C_f x_{f,i}(k) \end{gathered} \tag{4}$$

where, $x_{f,i}(k) = \begin{bmatrix} x_i(k) \\ f_{a,i}(k) \\ f_{s,i}(k) \end{bmatrix}, A_f = \begin{bmatrix} A & E & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, B_f = \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}, C_f = [C \quad 0 \quad F].$

Assumption 1. *Matrix (A, B) is state-controllable; matrix (A, C) is state-observable.*

Assumption 2. *Actuators and sensors are time-driven and clock-synchronized.*

Assumption 3. *The random network delay τ_k has an upper bound value $\tau_k \leq \bar{\tau}$.*

Assumption 4. *The communication topology graphs used in the MAS are undirected.*

The flowchart of the network active fault-tolerant predictive control scheme for a MAS as shown in Figure 1:

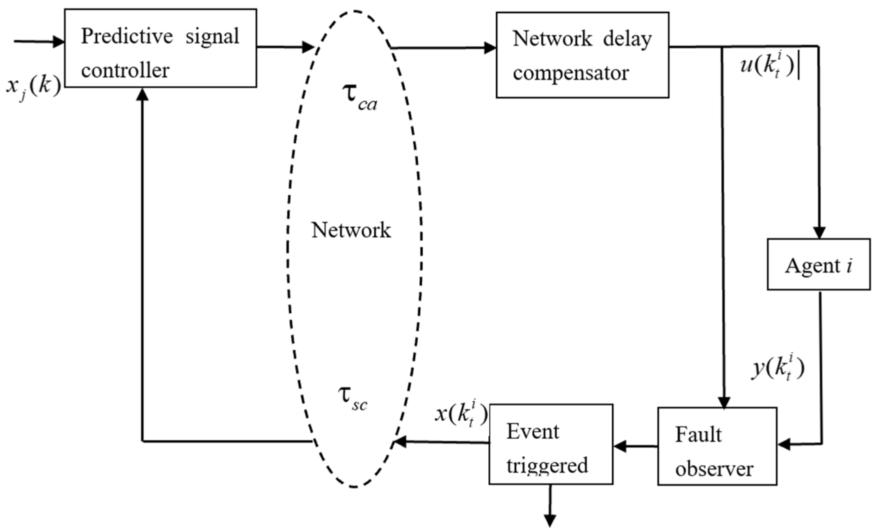

Figure 1. Network active fault tolerance predictive control scheme.

After the signal from Agent *i* reaches the fault observer, used to estimate possible actuator or sensor faults within the system, the fault observer dynamically generates state and fault estimates based on fault errors. When the system state changes exceed the set threshold, the ETM determines whether the current state requires updating and transmits it over the network. If the trigger conditions are met, the status signal is transmitted to the controller via the network channel; If not satisfied, the previous control signal is retained, thereby reducing the communication load. The controller predicts future states based on the latest status signals received and generates control signals. The control input undergoes compensation via the network delay compensator. Agent *i* receives the compensated control signal while simultaneously feeding the output signal back to the fault observer, thereby forming a closed-loop system.

3. Main Achievements

3.1. Fault Estimator Design

For a networked multi-agent discrete linear system under double faults, we design a fault observer, as shown below, to estimate the system's state and the actuator and sensor faults that may randomly occur in each agent in the system.

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + Bu_i(k) + E\hat{f}_{a,i}(k) + L\left(y_i(k) - C\hat{x}_i(k) - F\hat{f}_{s,i}(k)\right) \tag{5}$$

$$V_i(k+1) = M_1C\left(A\hat{x}_i(k) + Bu_i(k) + E\hat{f}_{a,i}(k)\right) + M_2\left(y_i(k) - C\hat{x}_i(k) - F\hat{f}_{s,i}(k)\right) \tag{6}$$

$$\hat{f}_{a,i}(k+1) = \hat{f}_{a,i}(k) + H\left(y_i(k) - C\hat{x}_i(k) - F\hat{f}_{s,i}(k)\right) \tag{7}$$

$$\hat{f}_{s,i}(k+1) = V_i(k+1) - M_1y_i(k+1) \tag{8}$$

To decouple the effects of actuator and sensor faults and to facilitate the algebraic estimation of sensor faults, an intermediate variable $V_i(k)$ is introduced. This variable integrates the model-based predicted output and the measurement residual through properly designed matrices M_1 and M_2. Based on $V_i(k)$, the sensor fault estimate can be obtained in an explicit form, which significantly simplifies the fault estimation structure and enables subsequent stability analysis.

Where $\hat{x}_i(k)$、$\hat{f}_{a,i}(k)$、$\hat{f}_{s.i}(k)$ are the state estimation, actuator fault estimation, and sensor fault estimation values of the MAS, respectively. L, H, M_2 are the observer gains are to be designed later to bring the system estimation error close to zero, as we specify here $M_1 = -(F^{\mathrm{T}}F)^{-1}F^{\mathrm{T}}$, where F is assumed to be of full column rank to ensure the solvability of the fault estimation and compensation matrices.

Collect system state estimates and fault estimates $D_{k_s} = \left[\hat{x}_i(k_s)^{\mathrm{T}},\ \hat{f}_{a,i}(k_s)^{\mathrm{T}}\right]^{\mathrm{T}}$ and send them to the controller through the network channel, where $\hat{x}_i(k_s)$、$\hat{f}_{a,i}(k_s)$ they denote the state estimation and actuator fault estimation at the time stamp, respectively.

The sensor fault estimation equation can be carried out with the following detailed derivation process:

$$\begin{aligned}
\hat{f}_{s,i}(k+1) &= V_i(k+1) - M_1y_i(k+1) \\
&= M_1C\left(A\hat{x}_i(k) + Bu_i(k) + E\hat{f}_{a,i}(k)\right) + M_2\left(y_i(k) - C\hat{x}_i(k) - F\hat{f}_{s,i}(k)\right) \\
&\quad -M_1\left(C\left(Ax_i(k) + Bu_i(k) + Ef_{a,i}(k)\right) + Ff_{s,i}(k)\right) \\
&= (M_2C - M_1CA)e_{x,i}(k) - M_1CEe_{f_{a,i}}(k) + M_2Fe_{f_{s,i}}(k) + f_{s,i}(k)
\end{aligned} \tag{9}$$

Here, it can be introduced that as long as the state and fault estimation errors in the agent system gradually converge to 0, the final sensor fault estimate is equal to the actual value of the fault.

3.2. Design of Event-Triggered Mechanism

To conserve network communication resources for MAS, we designed the system to incorporate an adaptive ETM; the decision to transmit a signal or not is made by dynamically adjusting the trigger condition parameter. The optimal trade-off between resource efficiency and control performance is realized to ensure system stability.

Here, the tracking error of agent *i* is defined as:

$$e_i = \hat{x}_i\left(k_t^i\right) - \hat{x}_i(k) \tag{10}$$

The adaptive event-triggered moment k_{t+1}^i is determined by the following equation:

$$k_{t+1}^i = \inf\left\{k > k_t^i \mid ||e_i(k)|| \geq \sigma_i(k)||x_i(k)||\right\} \tag{11}$$

Adaptive adjustment rate:

$$\dot{\sigma}_i(k) = -\lambda\sigma_i(k) + \mu\|e_i(k)\| \tag{12}$$

where k_t^i denotes the *t-th* trigger moment of the *i-th* agent. The lower bound inf determines the critical point at which an event is triggered, thus forming the event-triggered sequence k_{t+1}^i. $\lambda > 0$ indicates the forgetting factor, which controls the threshold decay rate; $\mu > 0$ denotes the error gain, which determines the effect of the error on the threshold.

When the system is in its initial stages or when a fault occurs, error spikes and threshold increases lead to elevated trigger frequency, enabling rapid synchronization recovery. Once the system stabilizes, the error margin decreases, and the threshold is lobe-end, thereby reducing the trigger frequency and conserving communication resources.

The adaptive event-triggered mechanism essentially implements a communication scheduler based on error energy self-feedback, minimizing network communication load while ensuring performance.

3.3. Fault-Tolerant Controller Design

Once the sensors send the collected data to the controller, the predictive controller of the networked multi-agent system is designed based on this data to compensate for the effect of faults on the system. For the studied multi-agent system, a state feedback controller design scheme is proposed here to design the fault-tolerant controller as:

$$u_i(k) = -K\varphi_i\left(k_t^i\right) - B^{+}Ef_{a,i}\left(k_t^i\right) \quad k \in \left(k_t^i, k_{t+1}^i\right) \tag{13}$$

$$\varphi_i\left(k_t^i\right) = \left(\sum_{\substack{j=1\\ j\neq i}}^{N} a_{ij}\left(x_i\left(k_t^i\right) - x_j\left(k_t^j\right)\right) + \mathrm{g}_i\left(x_i\left(k_t^i\right) - x_r(k)\right)\right) \tag{14}$$

where $K \in R^{m\times n}$ is the controller gain to be designed later. Denote by B^{+} the generalized inverse matrix of the matrix B, where it is specified $(I - BB^{+})E = 0$. k_t^i and k_t^j indicate the latest triggered moments of agent *i* and agent *j*, respectively. a_{ij} is an element of the corresponding position of the adjacency matrix *A* between the followers, indicates the connection relationship between followers, and g_i denotes the link between the leader and the tracker.

3.4. Delay Compensator Design

Since there is some delay in the network control system, the network delay compensator is designed to counteract the effect of network delay on the control system. To ensure that the system is getting the most up-to-date control signals, the following mechanism will be added here:

$$U_{k_t^i}^t = \begin{cases} U_{k_t^i}^t, & \text{if } k_t^i > k_t^{i*} \\ U_{k_t^{i*}}^t, & \text{if } k_t^i < k_t^{i*} \end{cases} \tag{15}$$

$$k_t^{i*} = \begin{cases} k_t^i, & \text{if } k_t^i > k_t^{i*} \\ k_t^{i*}, & \text{if } k_t^i < k_t^{i*} \end{cases} \tag{16}$$

Under the combined effect of timestamping and ETM, the network real-time delay can be expressed as:

$$\tau_k = k - k_t^{i*} \tag{17}$$

where the network delay is an integer value greater than zero. Here, τ_k it can be affected by a combination of event-triggered mechanisms and network delay; thus, the control signal of the network delay compensator can be expressed as:

$$u = \hat{u}(k|k-\tau_k) = \hat{u}\left(k_t^{i*} + \tau_k \middle| k_t^{i*}\right) \tag{18}$$

After the conversion of the above formulas, it can be guaranteed that the signal collected by the actuator is the control signal of the latest moment.

3.5. Fault Estimator Design

In case of possible random faults in both actuators and sensors, the compact form of the system is expressed as follows:

$$x(k+1) = (I_N \otimes A)x(k) + (I_N \otimes B)u(k) + (I_N \otimes E)f_a(k) \tag{19}$$

$$y(k) = (I_N \otimes C)x(k) + (I_N \otimes F)f_s(k) \tag{20}$$

where the symbol $\otimes$ is the Kronecker product, the states, inputs, outputs, actuator faults and sensor faults of each agent are represented by the set $x(k)$、$u(k)$、$y(k)$、$f_a(k)$、$f_s(k)$:

$$x(k) = [\, x_1^{\mathrm{T}}(k) \quad x_2^{\mathrm{T}}(k) \ \dots \ x_N^{\mathrm{T}}(k) \,]^{\mathrm{T}},$$

$$y(k) = [\, y_1^{\mathrm{T}}(k) \quad y_2^{\mathrm{T}}(k) \ \dots \ y_N^{\mathrm{T}}(k) \,]^{\mathrm{T}},$$

$$u(k) = [\, u_1^{\mathrm{T}}(k) \quad u_2^{\mathrm{T}}(k) \ \dots \ u_N^{\mathrm{T}}(k) \,]^{\mathrm{T}},$$

$$f_a(k) = [\, f_{a,1}^{\mathrm{T}}(k) \quad f_{a,2}^{\mathrm{T}}(k) \ \dots \ f_{a,N}^{\mathrm{T}}(k) \,]^{\mathrm{T}},$$

$$f_s(k) = [f_{s,1}^{\mathrm{T}}(k) \quad f_{s,2}^{\mathrm{T}}(k) \ \dots \ f_{s,N}^{\mathrm{T}}(k)]^{\mathrm{T}}.$$

The fault observer of the system is rewritten in compact form:

$$\hat{x}(k+1) = (I_N \otimes A)\hat{x}(k) + (I_N \otimes B)u(k) + (I_N \otimes E)\hat{f}_a(k) + (I_N \otimes L)\delta(k) \tag{21}$$

$$\begin{aligned} V(k+1) = (I_N \otimes M_1 C)\Big((I_N \otimes A)\hat{x}(k) + (I_N \otimes B)u(k) + (I_N \otimes E)\hat{f}_a(k)\Big) \\ +(I_N \otimes M_2)\delta(k) \end{aligned} \tag{22}$$

$$\hat{f}_a(k+1) = \hat{f}_a(k) + (I_N \otimes H)\delta(k) \tag{23}$$

$$\hat{f}_s(k+1) = V(k+1) - (I_N \otimes M_1)y(k+1) \tag{24}$$

where $\delta(k) = y(k) - (I_N \otimes C)\hat{x}(k) - (I_N \otimes F)\hat{f}_s(k)$.

The following dynamic estimation errors of states and faults are used to analyze the stability of the system:

$$e_x(k+1) = x(k+1) - \hat{x}(k+1) \tag{25}$$

$$e_{f_a}(k+1) = f_a(k+1) - \hat{f}_a(k+1) \tag{26}$$

$$e_{f_s}(k+1) = f_s(k+1) - \hat{f}_s(k+1) \tag{27}$$

where $e_x(k+1), e_{f_a}(k+1), e_{f_s}(k+1)$ denotes the system's state estimation error and the actuators' and sensors' fault estimation error, respectively.

Based on the MAS model and the fault estimation formula, the following conclusions can be deduced:

$$\begin{aligned} e_x(k+1) &= x(k+1) - \hat{x}(k+1) \\ &= (I_N \otimes A - I_N \otimes LC)e_x(k) + (I_N \otimes E)e_{f_a}(k) - (I_N \otimes LF)e_{f_s}(k) \end{aligned} \tag{28}$$

$$\begin{aligned} e_{f_a}(k+1) &= f_a(k+1) - \hat{f}_a(k+1) \\ &= -(I_N \otimes HC)e_x(k) + I_N e_{f_a}(k) - (I_N \otimes HF)e_{f_s}(k) \end{aligned} \tag{29}$$

$$\begin{aligned} e_{f_s}(k+1) &= f_s(k+1) - \hat{f}_s(k+1) \\ &= (I_N \otimes M_1 CA - I_N \otimes M_2 C)e_x(k) + (I_N \otimes M_1 CE)e_{f_a}(k) - (I_N \otimes M_2 F)e_{f_s}(k) \end{aligned} \tag{30}$$

Based on the above three equations, the coupled dynamic error equation can be derived:

$$\begin{aligned} \bar{e}(k+1) &= \begin{bmatrix} I_N \otimes A - I_N \otimes LC & I_N \otimes E & -(I_N \otimes LF) \\ -(I_N \otimes HC) & I_N & -(I_N \otimes HF) \\ I_N \otimes M_1 CA - I_N \otimes M_2 C & I_N \otimes M_1 CE & -(I_N \otimes M_2 F) \end{bmatrix} \bar{e}(k) \\ &= (\bar{A} - \bar{L}\bar{C})\bar{e}(k) \end{aligned} \tag{31}$$

where, $\bar{e}(k) = \begin{bmatrix} e_x(k) \\ e_{f_a}(k) \\ e_{f_s}(k) \end{bmatrix}$, $\bar{A} = \begin{bmatrix} I_N \otimes A & I_N \otimes E & 0 \\ 0 & I_N & 0 \\ I_N \otimes M_1 CA & I_N \otimes M_1 CE & 0 \end{bmatrix}$, $\bar{L} = \begin{bmatrix} I_N \otimes L \\ I_N \otimes H \\ I_N \otimes M_2 \end{bmatrix}$, $\bar{C} = [I_N \otimes C \quad 0 \quad I_N \otimes F]$.

Since the matrix $(\bar{A}, \bar{C})$ is fully observable and $\bar{L}$ can be made such that the eigenvalues $(\bar{A} - \bar{L}\bar{C})$ are within the unit circle, the system can be stabilized, and the state and fault estimation error will gradually approach zero.

The formula can be expressed as:

$$\begin{gathered}\lim_{x\to\infty} e_x(k) = 0\\ \lim_{x\to\infty} e_{f_a}(k) = 0\\ \lim_{x\to\infty} e_{f_s}(k) = 0\end{gathered} \tag{32}$$

Lemma 1. *The closed-loop system is asymptotically stable when and only when the eigenvalues of the triangular matrix on the matrix are within the unit circle [29].*

Theorem 1. *A closed-loop networked MAS is stable when and only when all eigenvalues of A and B are within the unit circle.*

Proof. Rewrite the observer fault estimate as:

$$\begin{gathered}\hat{x}(k+1) = (I_N \otimes A)\hat{x}(k) + (I_N \otimes B)u(k) + (I_N \otimes E)\hat{f}_a(k)\\ +(I_N \otimes LC)e_x(k) + (I_N \otimes LF)e_{f_s}(k)\end{gathered} \tag{33}$$

$$\hat{f}_a(k+1) = \hat{f}_a(k) + (I_N \otimes HC)e_x(k) + (I_N \otimes HF)e_{f_s}(k) \tag{34}$$

Rewrite the state prediction equation as follows:

$$\bar{x}(k+1) = (I_N \otimes A)\bar{x}(k) + (I_N \otimes B)u(k) + (I_N \otimes E)\bar{f}_a(k) \tag{35}$$

$$\bar{f}_a(k+1) = \bar{f}_a(k) \tag{36}$$

The result is obtained by subtracting the predicted value from the system estimate:

$$\tilde{e}(k+1) = (I_N \otimes A)\tilde{e}(k) + (I_N \otimes LC)e_x(k) + (I_N \otimes LF)e_{f_s}(k) \tag{37}$$

$$\tilde{f}_a(k+1) = \tilde{f}_a(k) + (I_N \otimes HC)e_x(k) + (I_N \otimes HF)e_{f_s}(k) \tag{38}$$

where, $\tilde{e}(k-\tau) = \hat{x}(k-\tau) - \bar{x}(k-\tau)$, $\tilde{f}_a(k-\tau) = \hat{f}_a(k-\tau) - \bar{f}_a(k-\tau)$.

Further derivations:

$$\begin{gathered}\tilde{e}(k) = (I_N \otimes A)^{\tau_k}\tilde{e}(k-\tau_k)\\ +\sum_{\tau=1}^{\tau_k}(I_N \otimes A)^{\tau-1}\left((I_N \otimes LC)e_x(k-\tau) + (I_N \otimes LF)e_{f_s}(k-\tau)\right)\\ =\sum_{\tau=1}^{\tau_k}(I_N \otimes A)^{\tau-1}\left((I_N \otimes LC)e_x(k-\tau) + (I_N \otimes LF)e_{f_s}(k-\tau)\right)\end{gathered} \tag{39}$$

$$\begin{gathered}\tilde{f}_a(k) = \tilde{f}_a(k-\tau_k) + \sum_{\tau=1}^{\tau_k}\left((I_N \otimes HC)e_x(k-\tau) + (I_N \otimes HF)e_{f_s}(k-\tau)\right)\\ =\sum_{\tau=1}^{\tau_k}\left((I_N \otimes HC)e_x(k-\tau) + (I_N \otimes HF)e_{f_s}(k-\tau)\right)\end{gathered} \tag{40}$$

To ensure the system's stability, the system network delay is coupled with the triggered interval, the triggered interval of the system signal $\Delta k_t^i = k_{t+1}^i - k_t^i$, and the maximum delay of the control input transmission in the network $\bar{\tau}_i$. Then, the control signal in the fault-tolerant controller is executed $k_t^i + \tau_k^i$. Here, the constraints need to be satisfied $\Delta k_t^i \geq \tau_k^i$.

The predicted value control signal of the whole agent system at the moment k_t is:

$$\begin{gathered}u(k_t) = -(I_N \otimes K)(L_s \otimes I_n)\left(x(k_t) - x_r(k_t)\right) - (I_N \otimes B^+E)\hat{f}_a(k_t)\\ = -(L_s \otimes K_n)\hat{\tilde{x}}(k_t) - (I_N \otimes B^+E)\hat{f}_a(k_t)\end{gathered} \tag{41}$$

where $L_s = L + G, L$ are Laplace matrices, $G = \text{diag}\{g_1, g_2, \dots, g_N\}$ is the agent connectivity matrix of leaders and followers, $K_n = K \otimes I_n$, $\tilde{x}(k)$ are the state tracking errors,

$$\tilde{x}(k)=\begin{bmatrix} x_1(k) - x_r(k) \\ x_2(k) - x_r(k) \\ \vdots \\ x_N(k) - x_r(k) \end{bmatrix}.$$

Combine the above expressions to arrive at it:

$$\begin{aligned}
\tilde{x}(k_{t+1}) &= (I_N\otimes A)\tilde{x}(k_t)+(I_N\otimes B)\left(-(L_s\otimes K_n)\hat{\tilde{x}}(k_t)-(I_N\otimes B^{+}E)\hat{f}_a(k)\right) \\
&\quad +(I_N\otimes E)f_a(k) \\
&= (I_N\otimes A-L_s\otimes BK_n)\tilde{x}(k_t)+(L_s\otimes BK_n)e_x(k_t) \\
&\quad +(L_s\otimes BK_n)\sum_{\tau=1}^{\tau_k}(I_N\otimes A)^{\tau-1}\left((I_N\otimes LC)e_x(k_t-\tau)+(I_N\otimes LF)e_{f_s}(k_t-\tau)\right) \\
&\quad +(I_N\otimes E)\left(e_{f_a}(k_t)+\sum_{\tau=1}^{\tau_k}\left((I_N\otimes HC)e_x(k_t-\tau)+(I_N\otimes HF)e_{f_s}(k_t-\tau)\right)\right) \\
&= (I_N\otimes A-L_s\otimes BK_n)\tilde{x}(k_t)+(L_s\otimes BK_n)e_x(k_t)+(I_N\otimes E)e_{f_a}(k_t) \\
&\quad +\sum_{\tau=1}^{\tau_k}\left((L_s\otimes BK_n)(I_N\otimes A)^{\tau-1}(I_N\otimes LC)+(I_N\otimes EHC)\right)e_x(k_t-\tau) \\
&\quad +\sum_{\tau=1}^{\tau_k}\left((L_s\otimes BK_n)(I_N\otimes A)^{\tau-1}(I_N\otimes LF)+(I_N\otimes EHF)\right)e_{f_s}(k_t-\tau)
\end{aligned} \tag{42}$$

The joint dynamic error equation and the multi-agent state equation can be derived:

$$\bar{X}(k_{t+1})=\Lambda\bar{X}(k_t) \tag{43}$$

where, $\bar{X}(k_t)=[\tilde{x}^{\mathrm{T}}(k_t)\quad \bar{e}^{\mathrm{T}}(k_t)\quad \bar{e}^{\mathrm{T}}(k_t-1)\quad \dots\quad \bar{e}^{\mathrm{T}}(k_t-\tau)]^{\mathrm{T}}$, $\Lambda=\begin{bmatrix}\Omega & \Phi\\ 0 & \Gamma\end{bmatrix}$, $\Omega=I_N\otimes A-L_s\otimes BK_n$, $\Gamma=\mathrm{diag}\{(\bar{A}-\bar{L}\bar{C}),\quad (\bar{A}-\bar{L}\bar{C}),\quad \dots,\quad (\bar{A}-\bar{L}\bar{C})\}$.

As long as the diagonal chunking system is satisfied to be asymptotically stable under any switching, the eigenvalues of the upper triangular chunking system are within the unit circle. Therefore, the system is stable, which is the end of the proof. □

4. Numerical Simulation

Here, to validate the effectiveness of the proposed method, we consider a homogeneous MAS network topology consisting of 1 leader and 4 followers. The Laplace matrix L defines the communication topology:

$$L=\begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

The communication topology as shown in Figure 2:

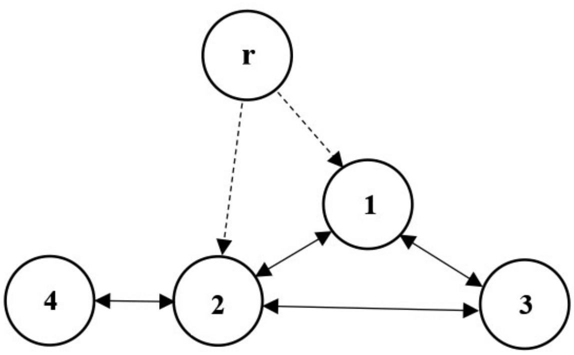

Figure 2. Networked Multi-agent Systems communication topology.

The state-space model is:

$$\begin{aligned} x_i(k+1) &= Ax_i(k) + Bu_i(k) + Ef_{a,i}(k) \\ y_i(k) &= Cx_i(k) + Ff_{s,i}(k) \end{aligned} \tag{44}$$

Control of threshold decay rate in the system: $\lambda = 1$, error gain: $\mu = 1$. The parameters of each matrix are set to:

$$A = \begin{bmatrix} 0.2 & 0.3 \\ 0.6 & 0.5 \end{bmatrix}, C = \begin{bmatrix} 0.1 & 0.8 \\ 0 & 0.2 \end{bmatrix}, B = \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}, E = B, F = \begin{bmatrix} 0 \\ 1 \end{bmatrix};$$

The initial value of each agent is set:

$$x_r(0) = [1;1], x_1(0) = [-1;2], x_2(0) = [-1;3], x_3(0) = [2;-1], x_4(0) = [3;-1]$$

The observer and the closed-loop system want the poles to be set:

$$[0.1 \quad 0.2 \quad 0.3 \quad 0.4], [0.4 + 0.2\mathrm{j} \quad 0.4 - 0.2\mathrm{j}].$$

Depending on the pole configuration, the observer gain and controller gain matrices can be obtained:

$$\bar{L} = \begin{bmatrix} 0.4425 & -0.1571 \\ 1.1037 & -1.6055 \\ 0.6001 & -1.1049 \\ -0.2232 & 0.0939 \end{bmatrix}, K = [1.1429 \quad -0.2381].$$

Here, the leader is set for an actuator fault $f_a(k)$, a sensor fault $f_s(k)$ in follower 1, and a sensor fault $f_w(k)$ in follower 2. The design actuator and sensor fault model are shown below:

$$f_a(k) = \begin{cases} 0, & k < 100 \\ 1 & k \geq 100 \end{cases} \tag{45}$$

$$f_s(k) = \begin{cases} 0, & k < 120 \\ \sin\big(0.2(k-120)\big), & k \geq 120 \end{cases} \tag{46}$$

$$f_w(k) = \begin{cases} 0, & k < 60 \\ 0.005(k-60)), & k \geq 60 \end{cases} \tag{47}$$

The simulation results are shown below.

Figure 3 shows the delay of followers 1–4 in a multi-agent system, and it describes the respective delay times of the agent systems in the network operation.

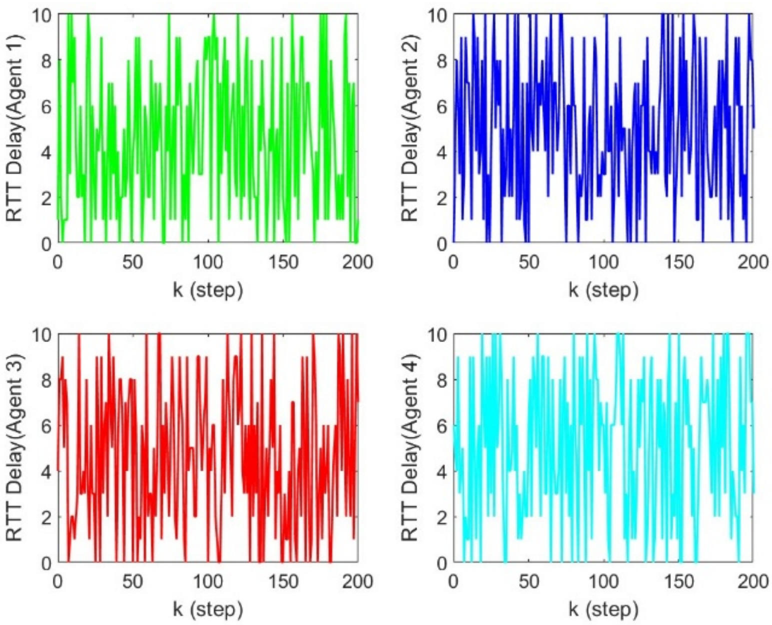

Figure 3. Network delay for each agent.

Figure 4 describes different agents' actual and estimated values when different faults occur. $k > 100$, an actuator fault occurs in the leader of the system, and the red dashed line indicates the fault estimate; $k > 60$, follower 2 has a sensor fault, and the blue dashed line indicates its fault estimate; $k > 120$, follower 1 has a sensor fault, and the green dashed line indicates its fault estimate.

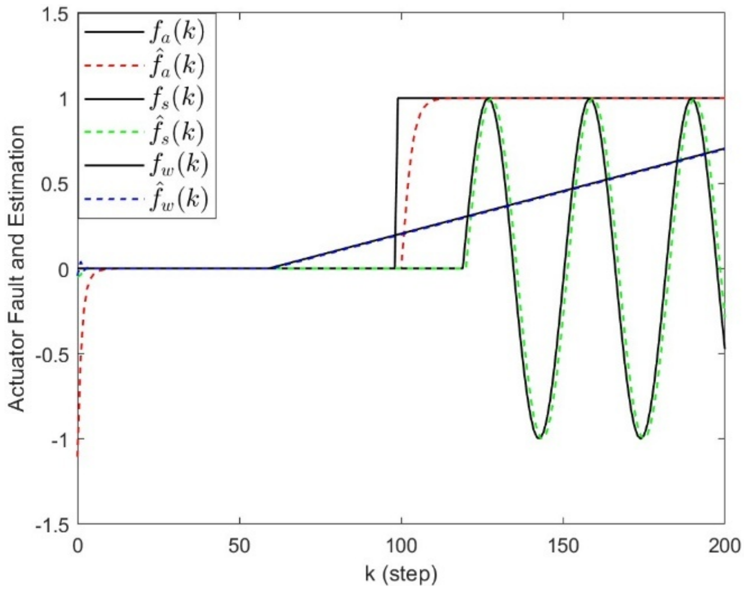

Figure 4. Actual and estimated values for each fault.

Figures 5 and 6 represent the states without and with prediction; Green, blue, red, and sky blue indicate the state values of the agent followers 1–4, respectively. The hand of faults when there is prediction is a feed-forward plus feedback two-layer control utilizing future state prediction, which can actively anticipate the effects of system lag and disturbances.

Figure 7 shows the output signal of the system without predictive compensation without FTC, and it can be seen that the system is destabilized and the output is divergent.

Figure 8 shows that, in the case of system network delay, output without time delay compensation, the output signal can be destabilized by network delay.

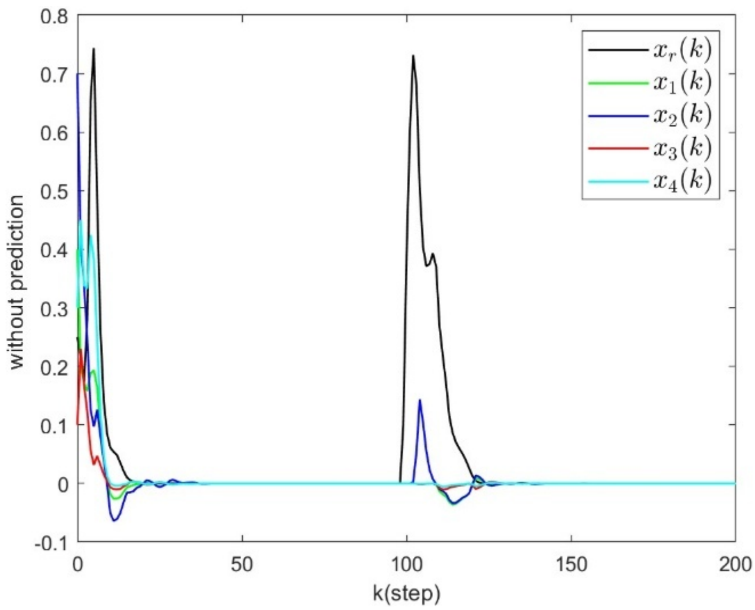

Figure 5. State of the system without prediction.

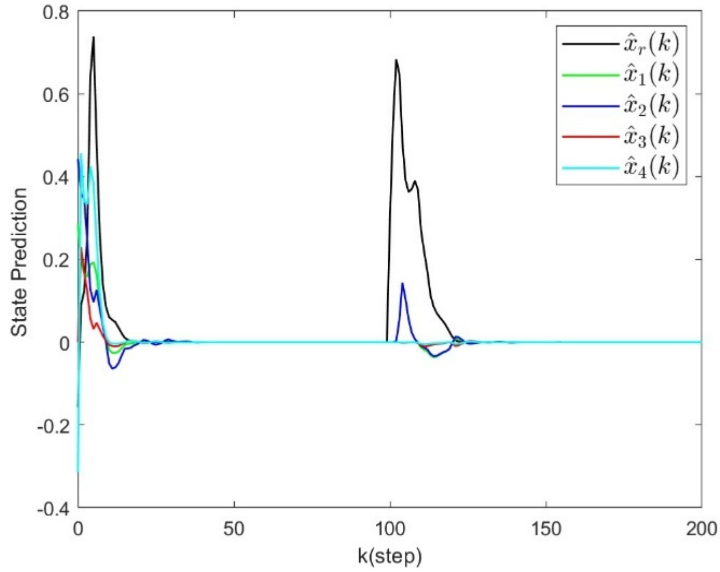

Figure 6. Status prediction.

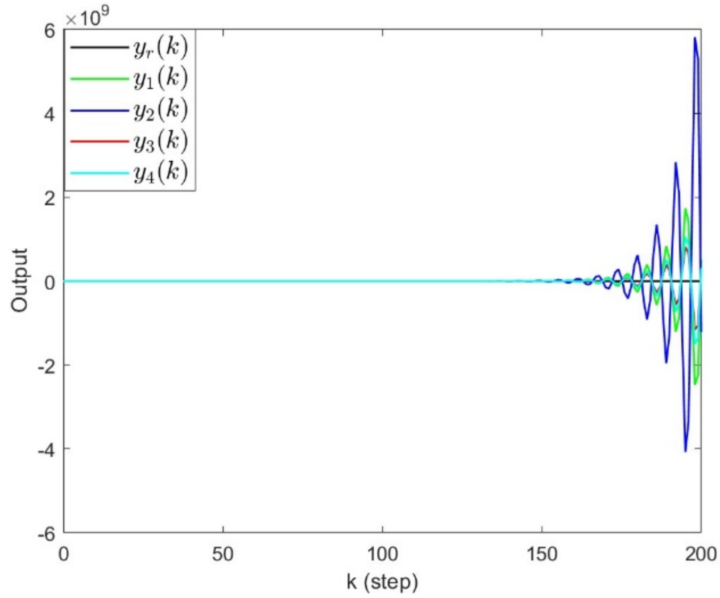

Figure 7. Outputs without predictive compensation without FTC.

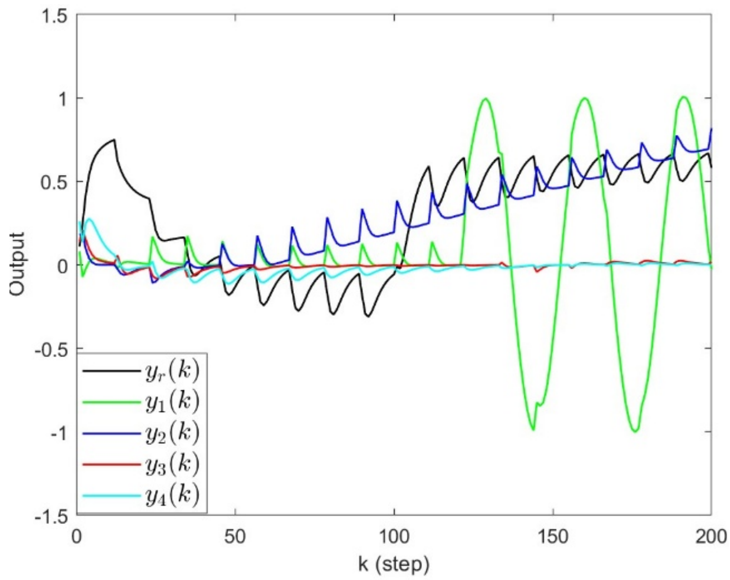

Figure 8. Outputs without delay compensation.

Figure 9 represents the actual outputs of the various agents of the system when different types of faults occur in the absence of FTC, its performance deteriorates dramatically and is unstable in the event of a fault.

Figure 10 illustrates the introduction of fault-tolerant controllers and delay compensators to compensate for system outputs in the event of multi-agent faults, thereby maintaining system stability and maximizing performance.

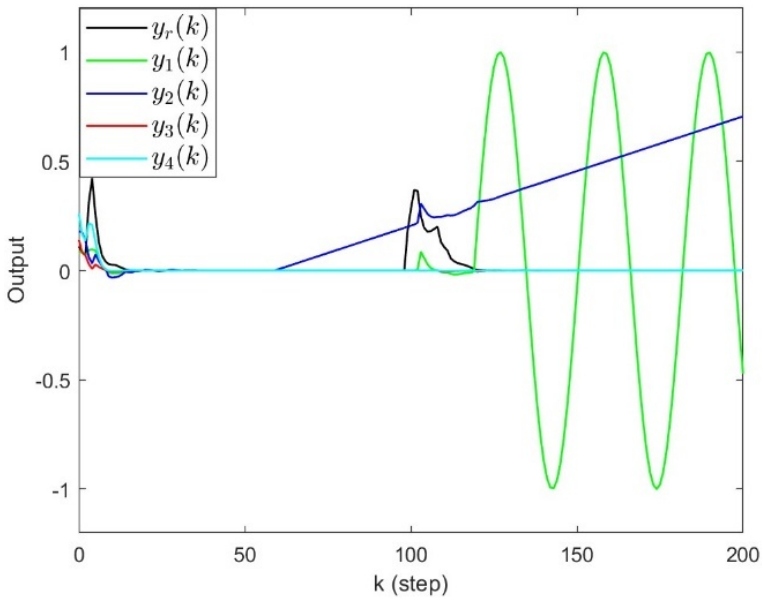

Figure 9. Actual outputs without fault tolerance.

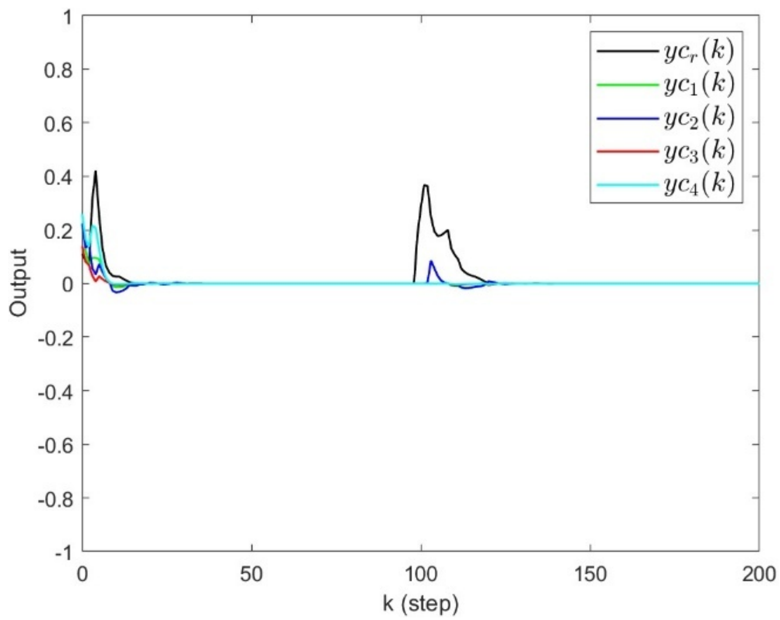

Figure 10. compensation outputs.

Figure 11 represents the tracking errors of the agent followers 1–4, the tracking errors of which are drastically increased by the faults of the agent system.

Figure 12 represents the trigger signals of the agent followers 1–4 under satisfying the adaptive ETM, which reduces unnecessary communication and effectively saves communication resources.

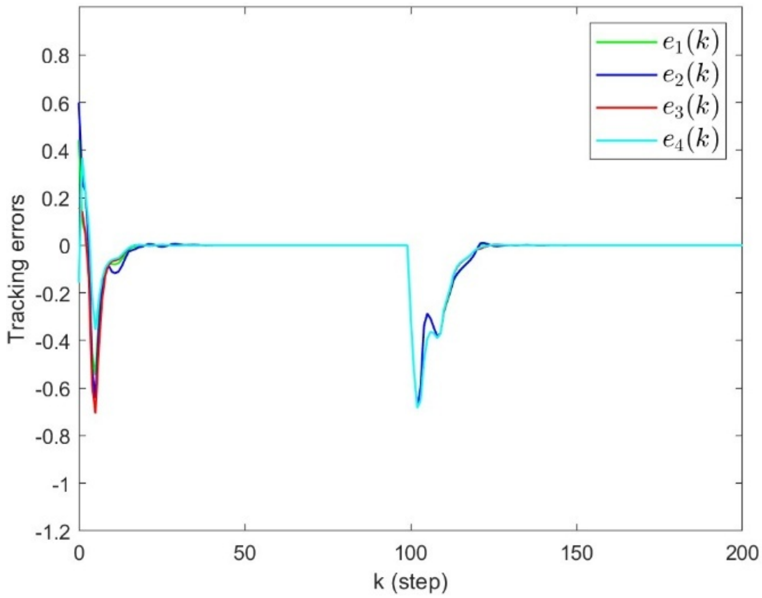

Figure 11. System tracking error.

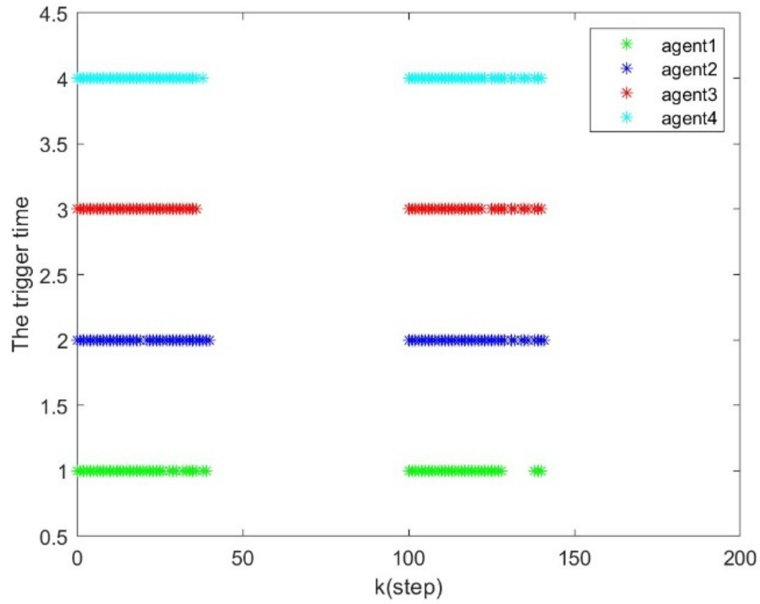

Figure 12. Trigger Time.

5. Conclusions

In the available literature, ETM and fault-tolerant predictive control are typically investigated separately. In this paper, these two approaches are deeply integrated to fully exploit the advantages of ETM in resource utilization and the fault handling capability of fault-tolerant predictive control. For discrete-time linear MAS in the presence of actuator and sensor double faults, the introduction of an ETM breaks the traditional time-driven fixed communication model, and agents communicate and control updates only when changes in the system state satisfy preset trigger conditions, significantly reducing unnecessary communication and lowering the need for communication bandwidth. The output feedback control under the event-triggered mechanism is proposed based on the fault information, which guarantees that the MAS effectively compensates after the occurrence of actuator faults and sensor faults to ensure the system's stability. Finally, multi-agent simulations are conducted to verify the effectiveness of the proposed fault-tolerant strategy in the presence of simultaneous faults.

Author Contributions: J.Z.: conceptualization, methodology; H.-H.L.: data curation; W.-R.W.: software, validation, writing—original draft preparation; G.-P.L.: reviewing.

Funding: This work was partly by the Natural Science Foundation of Shandong Province under Grant ZR2022QF080.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies: No AI tools were utilized for this paper.

References

1. Cai, W.; Zhao, W.; Xiang, X.; et al. Neural network control for mitigating actuator delay in ATR engines using predictive compensation and stability reward. *Chin. J. Aeronaut.* **2026**, *39*, 103720. https://doi.org/10.1016/J.CJA.2025.103720.
2. Yang, C.; Zhang, Q.; Zhang, J.; et al. Incremental multi-subreservoirs echo state network control for uncertain aeration process. *Neural Netw.* **2026**, *196*, 108454. https://doi.org/10.1016/J.NEUNET.2025.108454.
3. Tu, H.; Liu, S.; Fang, X.; et al. Research on Load Balancing for Distributed Control Planes in Multimodal Networks. *Chin. J. Electron.* **2025**, *53*, 3996–4009. Available online: https://link.cnki.net/urlid/11.2087.TN.20251210.0954.046 (accessed on 23 December 2025). (In Chinese)
4. Tang, Y.; Shou, Y.; Xu, B.; et al. Consensus learning based coordinated formation control of multiple UAVs. *Chin. J. Aeronaut.* **2026**, *39*, 103722. https://doi.org/10.1016/J.CJA.2025.103722.
5. Qin, J.H.; Ma, Q.C.; Li, M.; et al. A review of progress in multi-intelligence collaboration research: A game and control intersection perspective. *J. Autom.* **2025**, *51*, 489–509. https://doi.org/10.16383/j.aas.c240508.
6. Guo, H.B.; Pang, Z.H.; Li, C. Side Information-Based Stealthy False Data Injection Attacks Against Multi-Sensor Remote Estimation. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 1054–1056. https://doi.org/10.1109/JAS.2023.124086.
7. Guo, H.B.; Pang, Z.H.; Sun, J.; et al. A Stealthy False Data Injection Attack Scheme Against Sensor Measurements Using Partial System Knowledge. *IEEE Trans. Autom. Control.* **2025**, *70*, 5600–5607.
8. Jiang, B.; Xu, Y.H.; Yang, H. A review of research on fault-tolerant control, optimization and gaming for network systems. *J. Nanjing Univ. Aeronaut. Astronaut.* **2022**, *54*, 789–800. https://doi.org/10.16356/j.1005-2615.2022.05.005.
9. Lu, X.; Yu, C.Y.; Yang, H.L.; et al. Multi-quadrotor formation cooperative fault-tolerant control under sensor faults. *J. Aerosp. Sci. Technol.* **2021**, *9*, 119–132.
10. Zhang, P.; Xue, H.F.; Gao, S.; et al. Adaptive cooperative fault-tolerant control of multi-agent formation system with hybrid actuator faults and multiple unknown control directions. *J. Natl. Def. Univ. Sci. Technol.* **2022**, *44*, 131–140.
11. Zhang, X.; Chen, M.Y.; Wang, L.; et al. Fault-tolerant consistency of multi-intelligent body systems in the presence of actuator faults. *J. Shanghai Jiao Tong Univ.* **2015**, *49*, 806–811. https://doi.org/10.16183/j.cnki.jsjtu.2015.06.012. (In Chinese)
12. Shi, L.; Wu, L.B. Adaptive inclusion control of nonlinear multi-intelligent body systems with input quantization and actuator faults. *J. Sci. Teach. Coll. Univ.* **2024**, *44*, 15–27.(In Chinese)
13. Lin, M.F.; Zhang, T.P. Adaptive dynamic surface control of uncertain multi-intelligent body systems with actuator faults. *Control. Theory Appl.* **2021**, *38*, 1452–1465.(In Chinese)
14. Zhong, M.Y.; Liu, H.; Zhu, X.Q.; et al. A review of fault diagnosis technology based on event triggering mechanism. *J. Shandong Univ. Sci. Technol.* **2021**, *40*, 95–1102. https://doi.org/10.16452/j.cnki.sdkjzk.2021.02.011. (In Chinese)
15. Yang, F.S.; Wang, J.; Pan, Q. A review of research on network control based on event-triggered mechanism. *Control. Decis. Mak.* **2018**, *33*, 969–977. https://doi.org/10.13195/j.kzyjc.2017.0531.
16. Zhang, D.; Ma, S.H.; Lv, S.; et al. A review of event-triggered consistency research in multi-intelligent body systems. *J. Beijing Inst. Technol.* **2022**, *42*, 1059–1072. https://doi.org/10.15918/j.tbit1001-0645.2021.222.
17. Li, K.D. Research on Fault-Tolerant Coherent Control of Multi-Agent Event-Triggered Considering Actuator Faults. Master's Thesis, Institute of Electrical Engineering, Yanshan University, Qinhuangdao, China, 2023.
18. Gao, X.Y.; Fu, D.F.; Meng, H.F. Dynamic event-triggered predictive control for coherent multi-agent systems. *Control. Theory Appl.* **2025**, *42*, 2398–2408. (In Chinese)
19. Wang, J.; Cai, X.Q. Fault-tolerant consistency of heterogeneous multi-intelligent body systems based on adaptive distributed event triggering mechanism. *Syst. Sci. Math.* **2025**, *45*, 3635–3656. Available online: http://kns.cnki.net/kcms/detail/11.2019.O1.20240826.1637.016.html (accessed on 23 December 2025). (In Chinese)
20. Wang, J.; Zhang, J. Adaptive fault-tolerant control of multi-agent system based on event-triggered. *Electron. Des. Eng.* **2022**, *30*, 1–5. https://doi.org/10.14022/j.issn1674-6236.2022.05.001.
21. Yang, R.H.; Zhang, H.; Yan, H.C. Event-triggered event-based cooperative output regulation of topology-switching heterogeneous multi-agent. *Acta Autom. Sin.* **2017**, *43*, 472–477. https://doi.org/10.16383/j.aas.2017.c160304.
22. Yang, R.H.; Zhang, H.; Feng, G.; et al. Distributed event-triggered adaptive control for cooperative output regulation of heterogeneous multi-agent systems under switching topology. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3528.
23. Xiao, J.; Liu, Y. Adaptive neural network dynamic event-triggered consensus control for nonlinear multi-agent systems subject to sensor deception attacks and actuator faults. *Nonlinear Dyn.* **2024**, *112*, 20019–20034. https://doi.org/10.1007/S11071-024-10116-W.
24. Zhang, W.; Tong, S.C. Event-triggered event-based fault-tolerant control of fuzzy networked time-lag systems. *Fuzzy Syst. Math.* **2019**, *33*, 19–28. (In Chinese)

25. Zhang, L.; Yao, H. Exponential Stability Analysis and Observer-based Control of T-S Fuzzy Networked Systems with Time Delay and Communication Delay. *Eng. Lett.* **2022**, *30*, 1689–1695.
26. Le, W.; Shen, B.; Zou, L. Stubborn state estimation for time-delay systems with relay communication network: The half-duplex case. *J. Frankl. Inst.* **2025**, *362*, 107465.
27. Lu, Y.; Xia, J.; Shen, L. Predictive Control Strategy Based Fault-Tolerant Formation Control for Fully Actuated Multi-Agent Systems. *Circuits Syst. Signal Process.* **2025**, *44*, 7092–7107.
28. Li, F.Q.; Dou, G.S.; Zheng, B.Z. Research on average consistency of multi-intelligent body network based on event triggering mechanism. *Comput. Appl. Res.* **2017**, *34*, 665–670.
29. Zhang, J. Research on Active Fault-Tolerant Predictive Control Method of Networked System. Master's Thesis, School of Information and Control Engineering, Qingdao University of Technology, Qingdao, China, 2016. (In Chinese)