

Event-Triggered Fault Detection and Control in Nonlinear Affine Multi-Agent Systems with Affine Parameter Variations

Mohammad Zangouei¹ , Naser Pariz¹ , Reihaneh Kardehi Moghaddam² 

¹ Engineering Faculty, Electrical Department, Ferdowsi University of Mashhad, Mashhad, Iran.

² Department of Electrical Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran.

✉ Correspondence:

Naser Pariz

E-mail:

n-pariz@um.ac.ir

How to Cite

Zangouei, M., Pariz, N., Kardehi Moghaddam, R. (2025). "Event-triggered fault detection and control in nonlinear affine multi-agent systems with affine parameter variations", Control and Optimization in Applied Mathematics, 10(2): 1-50, doi: [10.30473/coam.2025.75259.1324](https://doi.org/10.30473/coam.2025.75259.1324).

Abstract. In this paper, we present an event-triggered fault-tolerant control framework for nonlinear affine multi-agent systems, together with a state-observer-based fault detection scheme. The proposed approach integrates an event-triggered controller that reduces communication and computation while guaranteeing closed-loop stability, with a robust fault-detection mechanism capable of identifying sensor faults, including current-sensor faults, under bus and load disturbances, and leveraging sensor redundancy to enable rapid recovery. A rigorous stability and robustness assessment based on eigenvalue analysis of the observer matrix is complemented by extensive MATLAB simulations that demonstrate resilience to parameter variations and external disturbances. Open-loop analyses under unconventional inputs reveal high sensitivity to fault types while exhibiting insensitivity to load disturbances, underscoring the detector's discriminative capability. To mitigate startup and transient effects, a low-pass filter is implemented at the detector output, reducing transients and improving fault-detection accuracy for real-time identification of current sensor faults. The overall results show reliable fault detection, rapid recovery, and maintained performance in the presence of sensor faults and load disturbances, thereby enhancing the robustness of nonlinear affine multi-agent systems.

Keywords. Nonlinear multi-agent systems, Event-triggered control, Stability analysis, Sensor faults, Low-pass filtering.

MSC. 34H05; 94C12; 62F35.

1 Introduction

The rapid advancement of industry and technology continually emphasizes the importance of accurately modeling, analyzing, and controlling dynamic systems. Dynamic systems are those whose states evolve over time [6], and underpin a wide range of applications across mechanical, electrical, thermal, and biological domains. In control theory [1], the focus is on understanding how inputs and disturbances shape system behavior and performance, with differential equations providing a common modeling framework for many physical phenomena. Over the past decades, fractional calculus theory has emerged as a powerful tool for modeling complex dynamics [21]. By extending differentiation to non-integer orders [20], fractional-order models capture memory effects and hereditary properties, often yielding more accurate representations than traditional integer-order models [1, 16, 28]. The non-local nature of fractional derivatives [20, 29] implies that the future evolution of a system depends on its entire history, which is advantageous for describing viscoelasticity, diffusion processes, and various engineering systems [29]. Consequently, a broad class of interdisciplinary systems, such as viscoelastic materials, butterfly axis systems, thermoelectric devices, reaction-diffusion processes, viral spread models, investment systems, electrical circuits, respiratory mechanisms, biological networks, and vehicle battery systems, are effectively modeled and analyzed using fractional-order derivatives [9]. However, fractional-order systems can also exhibit chaotic and highly sensitive behavior under certain conditions [25], motivating ongoing research into their stability, control, and synchronization. Chaos refers to irregular, highly sensitive, and non-repetitive dynamics observed in numerous nonlinear systems. The hallmark of chaotic systems is their sensitivity to initial conditions: small differences in the starting states can lead to vastly divergent outcomes. Unlike non-chaotic systems, where deviations grow linearly and are often considered measurement errors, chaos-induced divergence increases exponentially, making long-term prediction inherently infeasible [12]. This unique property has prompted extensive research into chaotic systems, which have found applications in various fields including chemical reactions, power systems, secure communications, information processing, biological systems, and mechanical devices.

For a dynamic system to exhibit chaos, certain necessary conditions must be satisfied [12]: (i). The system must comprise at least three variables. (ii). Its governing equations must include at least one nonlinear term that couples two or more variables.

Notable fractional-order chaotic systems extensively studied for their applications include the fractional-order Chen, Lü, Rössler, Lorenz, Chua, Arneodo, Duffing, and Genesio–Tesi systems [30]. Among these, fractional-order systems are particularly valuable for analyzing complex nonlinear behaviors due to their flexibility and accuracy. The rapid growth of autonomous and networked systems has heightened the need for reliable operation under faults and disturbances. In this work, we address fault diagnosis and fault-tolerant control (FTC) for

a nonlinear affine multi-agent system (MAS) governed by several interacting subsystems. Our focus is on achieving timely fault detection and robust stabilization when sensor faults occur, particularly in current sensors, and when the system is subjected to external load disturbances.

Key motivations and scope

- The challenge of maintaining stability and performance in the presence of sensor faults and disturbances in nonlinear MAS.
- The benefits of reducing communication and computation while preserving stability through an *event-triggered control* (ETC) paradigm.
- The importance of integrating fault detection with the system model to enable rapid identification and isolation of faults, followed by resilient recovery.

Our proposed approach consists of three building blocks:

1. **Model-based fault detection:** a state observer is designed for the affine nonlinear MAS, yielding residual signals that are monitored to detect faults in the current sensors. The observer gains are chosen to guarantee robust performance and to suppress spurious detections due to disturbances.
2. **Event-triggered control (ETC):** the controller uses event-triggered updates, reducing unnecessary communication while ensuring closed-loop stability. The triggering rule is derived to balance detection speed, communication load, and robustness.
3. **Fault-resilient recovery:** upon detection of a faulty sensor, a redundancy-based recovery strategy substitutes the faulty measurement with a backup sensor, enabling the system to reconstruct nominal operation and maintain stability.

Contributions and organization

- We formulate the problem within a state-space, affine nonlinear MAS framework and design an ETC scheme compatible with fault detection.
- We develop a state-observer-based fault-detection mechanism capable of distinguishing current-sensor faults under varying load conditions.
- We provide a stability and robustness analysis by examining the eigenvalues of the observer system matrix and by exploring parameter variations.

- We validate the approach through MATLAB simulations under diverse fault scenarios and disturbances, demonstrating accurate fault detection, rapid isolation, and successful fault recovery.

The remainder of the paper is organized as follows: Section 2 presents the state-space model of the affine nonlinear MAS, the underlying assumptions, and the problem formulation for fault detection and control. Section 3 details the fault-detection framework based on a state observer, including the residual dynamics and fault isolation capabilities. Section 4 introduces the event-triggered controller and the associated triggering mechanism, along with the design choices to reduce communication while preserving performance. Section 5 discusses stability and robustness analyses, followed by simulation results. Section 6 extends the discussion with additional robustness results, adaptive/learning aspects, and practical considerations for real-time implementation. Finally, Section 7 concludes the work and outlines future directions.

2 Theoretical Foundations

Multi-agent systems offer numerous advantages over single-agent systems, including enhanced parallel processing capabilities, increased resilience to failures, and robust operation under diverse conditions due to the absence of a centralized controller. These benefits make multi-agent systems particularly attractive for complex engineering applications. However, the ongoing advancements in computer, communication, and control technologies have compounded system complexity, escalating the likelihood of faults affecting system performance. Consequently, there is an imperative need for control strategies that can detect, mitigate, or tolerate such faults, ensuring continued and reliable operation [9]. This necessity has driven substantial research interest in the fault-tolerant control (FTC), which aims to maintain system stability and functionality in the presence of component faults [25]. Fault tolerance allows systems to operate correctly despite the occurrence of faults within their subsystems, thus significantly enhancing robustness and dependability.

The concept of incorporating redundancy to enhance system reliability was pioneered by John von Neumann in the 1950s. Over the decades, the field has evolved considerably; in 1991, foundational principles of FTC were formalized in a seminal publication [12]. A comprehensive review published in 1997 further clarified critical issues and methodologies within FTC systems [26].

Multiple studies have applied FTC concepts specifically to multi-agent systems. For instance, reference [31] proposed a fault tolerance strategy ensuring all agents stabilize at a common target despite individual agent faults. Similarly, reference [23] introduced an FTC scheme tailored for multi-agent systems experiencing actuator bias faults. Additionally, reference [11]

surveyed model-based FTC methodologies, including observer-based designs and parameter estimation techniques. Historical development of model-based approaches can be traced back to Willsky's pioneering work in 1976, which introduced key concepts in FTC reconfiguration systems [7]. The evolution of FTC also encompasses results related to nonlinear systems, broadening the scope and applicability of fault-tolerant strategies [24].

3 Literature Review

This section provides an overview of recent advancements in the FTC for nonlinear systems, with a particular focus on multi-agent systems. Multi-agent systems comprise multiple interacting agents that communicate with each other, often operating within a shared environment and possessing individual spheres of influence, which can overlap. These agents are interconnected through various organizational communication protocols. The versatility of multi-agent systems has led to their widespread adoption across fields such as defense, manufacturing, and robotics. A review of the literature published over the past two decades, as summarized in [2], highlights the rapid growth and increasing significance of research in this domain. In terms of dynamic structures, much of the existing controller design work addresses systems with linear dynamics. For example, Liu et al. [14] explored the development of controllers aimed at achieving consensus among agents within a linear multi-agent system framework. When it comes to fault detection and isolation in such systems, various approaches have been proposed. In [8], an online least squares method is employed to estimate faults, complemented by a Luenberger observer for fault identification. Similarly, Han et al. utilized observer-based techniques for fault detection and diagnosis [5]. The fault hiding approach, based on adaptive control, is detailed in [27], while Farzanegan et al. also employed adaptive control for fault-tolerant design [4]. Additionally, Chen and Herrmann introduced a residual generator using a filter bank combined with a Descriptor Estimator System (DES) unit for fault detection [3].

For nonlinear affine multi-agent systems, the literature on FTC design remains relatively sparse. Notably, recent works such as [10] utilize neural networks for fault detection and diagnosis, demonstrating the potential of machine learning techniques in fault management. Other approaches, such as [15], employ Kalman filters to identify and diagnose faults, leveraging their effectiveness in stochastic estimation. Additionally, Lyapunov-based methodologies are discussed in [22], offering systematic frameworks to ensure system stability under fault conditions. Despite these advancements, research targeting non-affine nonlinear multi-agent systems is still limited, with only recent contributions like [13] exploring neural network- and adaptive control-based fault tolerance in this context. A critical observation is that approximately 85% of current research efforts are concentrated on fault detection and diagnosis, while only around

15% investigate fault hiding strategies. Fault hiding offers significant advantages because it enables the primary controller to operate without modification during faults, simplifying the reconfiguration process. Unlike fault detection methods that rely heavily on fault diagnosis units, which can be prone to errors or may inadvertently introduce additional faults, the fault hiding approach maintains system stability and performance without necessitating architectural changes. To date, a fault hiding-based FTC design specifically for linear multi-agent systems has been proposed in [19]. However, to our knowledge, such methodologies have yet to be extended to affine nonlinear systems with nonlinear input gains. This represents a promising area for further research. Moreover, integrating event-triggered mechanisms within the FTC architectures presents a compelling avenue for enhancing efficiency and resilience. The ETC reduces communication bandwidth and energy consumption by ensuring information exchanges occur only when necessary. Additionally, this approach inherently offers protection against unexpected errors and disturbances. Consequently, developing event-triggered, fault-tolerant schemes for multi-agent systems constitutes a timely and impactful research direction, promising improvements in both resource management and system robustness.

Recent theoretical advances have established foundations for predefined-time notions in consensus and fault-tolerant operation. In particular, Mei and Wen (2024) provide foundational insights into predefined-time event-triggered consensus for second-order MAS under actuator faults and saturation, informing robust fault-tolerant design in nonlinear MAS with similar disturbance structures [18]. Building on this line, Ma, Tan, and Mei (2023) offer foundational results on predefined-time consensus for nonlinear MAS subject to input delays and switching topology, employing a dynamic event-triggered framework to achieve convergence within a prescribed time while reducing communication demands [17].

4 Research Methodology

4.1 Event-triggered Controller Design for Nonlinear Multi-Agent Systems

In this section, we focus on the development of an event-triggered controller for nonlinear multi-agent systems. The ETC rule determines when to update control actions to balance performance and communication. We derive the triggering condition to guarantee asymptotic stability under the designed estimator dynamics and to tolerate bounded disturbances. The primary objective is to minimize unnecessary communication between agents by ensuring that information exchanges occur only when required, enabling follower agents to effectively track the leader agent under these event-based conditions. The dynamics of each agent within this multi-agent framework are modeled as follows.

$$\begin{cases} \dot{x}_i(t) &= Ax_i(t) + b[f(x_i(t)) + Bu_i(t)], \\ y_i(t) &= Cx_i(t), \\ x_i(0) &= x_{i0}. \end{cases} \quad (1)$$

where $x(t) \in \mathbb{R}^n$ represents the state variable vector, $u(t) \in \mathbb{R}^m$ denotes the control input, and $y(t) \in \mathbb{R}^l$ implies the output vector. $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^{n \times 1}$, $B \in \mathbb{R}^{1 \times m}$ and $C \in \mathbb{R}^{l \times n}$ are constant matrices. The function $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a Lipschitz continuous nonlinear function.

In Kronecker product form, the entire multi-agent system (1) can be expressed as

$$\dot{X}(t) = (I_N \otimes A)X(t) + (I_N \otimes b)[F(t) + (I_N \otimes B)U(t)], \quad (2)$$

where

$$\begin{aligned} X(t) &= [x_1(t), x_2(t), \dots, x_N(t)]^T, \\ F(t) &= [f(x_1(t)), f(x_2(t)), \dots, f(x_N(t))]^T, \\ U(t) &= [u_1(t), u_2(t), \dots, u_N(t)]^T. \end{aligned}$$

The dynamics of the leader agent are also modeled as Equation (3):

$$\dot{x}_r(t) = Ax_r(t) + B'r(t), \quad (3)$$

where $x_r(t) \in \mathbb{R}^n$ denotes the leader's state, and $r(t)$ is the reference input. The pair (A, B') is assumed controllable. The observer uses available outputs to produce an estimate $\hat{x}(t)$ of the state. The residual

$$r(t) = y(t) - \hat{y}(t),$$

serves as the fault indicator, defined as the discrepancy between measured outputs and observer predictions. In healthy operation, $r(t) \rightarrow 0$; deviations in $r(t)$ indicate faults, with robustness to bounded disturbances ensured by the observer gain design.

The Kronecker form for the leader's dynamics is written as

$$\dot{X}_r(t) = (I_N \otimes A)X_r(t) + (I_N \otimes B')R(t), \quad (4)$$

with

$$X_r(t) = [x_{r1}(t), x_{r2}(t), \dots, x_{rN}(t)]^T, \quad R(t) = [r_1(t), r_2(t), \dots, r_N(t)]^T.$$

State Estimation via Observer

Since not all state variables are accessible at all times, each agent employs an observer for estimation. The observer dynamics used for each agent are given as (5):

$$\dot{\hat{x}}_i(t) = A\hat{x}_i(t) + b[f(\hat{x}_i(t)) + Bu_i(t)] + \ell_i c_i (\hat{x}_i(t) - x_i(t_k)), \quad (5)$$

where $\hat{x}_i(t)$ is the estimated state variable, $x_i(t_k)$ denotes the available state measurement at the previous sampling time t_k , and ℓ_i is the observer gain. Defining the estimation error $\bar{x}_i(t) = x_i(t) - \hat{x}_i(t)$ and the measurement discrepancy $\Delta x_i = x_i(t) - x_i(t_k)$, and expressing in Kronecker form, the collective observer Equation (5) is rewritten as follows:

$$\dot{\hat{X}}(t) = (I_N \otimes A)\hat{X}(t) + (I_N \otimes b)[F(\hat{X}(t)) + (I_N \otimes B)U(t)](I_N \otimes lc)(\Delta X(t) - \bar{X}(t)), \quad (6)$$

where

$$\hat{X}(t) = [\hat{x}_1(t), \hat{x}_2(t), \dots, \hat{x}_N(t)]^T, \quad F(\hat{X}(t)) = [f(\hat{x}_1(t)), f(\hat{x}_2(t)), \dots, f(\hat{x}_N(t))]^T.$$

Leader-Follower Consensus and Error Dynamics

The control design aims at achieving consensus with the leader. Define the local error for each agent as:

$$e_i(t) = \sum_{j=1, j \neq i}^N a_{ij} (\hat{x}_j(t) - \hat{x}_i(t)) + g_i (x_r(t) - \hat{x}_i(t)) \quad (7)$$

where $A = [a_{ij}] \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix with $a_{ii} = 0$. In this matrix, if $(v_j, v_i) \in \epsilon$, then $a_{ij} = 1$; otherwise, $a_{ij} = 0$. Moreover, if the follower agents are directly connected to the leader agent, $g_i^* > 0$; otherwise, $g_i^* = 0$.

The collective error vector is:

$$E(t) = [e_1(t), e_2(t), \dots, e_N(t)]^T.$$

Event-Triggering Mechanism

The control system incorporates an event-based triggering condition to determine when communication occurs. This is defined by the relationship

$$W(E(t), X(t)) = 0, \quad (8)$$

where

$$E(t) = [e_1(t), e_2(t), \dots, e_N(t)]^T, \quad X(t) = [x_1(t), x_2(t), \dots, x_N(t)]^T,$$

and $W(\cdot, \cdot)$ is the event-trigger function that governs the timing of information exchange. The sequence of event times $t_0, t_1, \dots$ is determined by the condition

$$W(E(t_k), X(t_k)) = 0$$

for $k = 0, 1, 2, \dots$

Event-triggered Controller Design

Figure 1 illustrates the structure of the proposed controller for each individual agent.

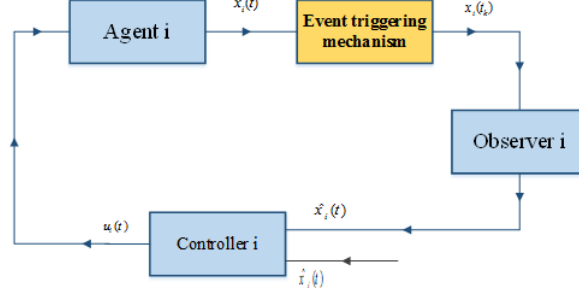


Figure 1: Structure of the event-triggered controller for each agent.

Error term re-expression

To clarify the design of this controller, we begin by re-expressing the error term introduced in Equation (7) as follows:

$$\begin{aligned}
 E(t) = & \begin{bmatrix} a_{i1}, \dots, a_{iN} \end{bmatrix} \otimes I_n \hat{X}(t) - (D \otimes I_n) \hat{X}(t) + \begin{bmatrix} g_1 & \dots & 0 \\ \cdot & \dots & \cdot \\ 0 & \dots & g_N \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} x_r(t) \\
 & - \begin{bmatrix} g_1 \otimes I_n & \dots & 0 \\ \vdots & & \\ 0 & \dots & g_N \otimes I_n \end{bmatrix} \hat{X}(t). \tag{9}
 \end{aligned}$$

Next, define the matrices:

$$\begin{aligned}
 A^* &= [a_{i1}, a_{i2}, \dots, a_{iN}], \\
 G &= \text{diag}(g_1, g_2, \dots, g_N), \\
 D &= \text{diag}(d_1, d_2, \dots, d_N), \quad \text{where } d_i = \sum_j a_{ij},
 \end{aligned}$$

and utilize the Laplacian $L = D - A^*$. With these definitions, the error term simplifies to:

$$\begin{aligned}
 E(t) &= (A^* \otimes I_n) \hat{X}(t) - (D \otimes I_n) \hat{X}(t) + ((L + G) \otimes I_n) x_r(t) - (G \otimes I_n) \hat{X}(t) \\
 &= -[(L + G) \otimes I_n] (\hat{X}(t) - X_r(t)). \tag{10}
 \end{aligned}$$

Differentiating Equation (10) with respect to time yields

$$\dot{E}(t) = -[(L + G) \otimes I_n](\dot{\hat{X}}(t) - \dot{X}_r(t)). \quad (11)$$

Applying Equations (4) and (6), we obtain:

$$\begin{aligned} \dot{E}(t) = & -[(L + G) \otimes I_n]((I_N \otimes A)\hat{X}(t)) - [(L + G) \otimes I_n]((I_N \otimes b)F(\hat{x}(t))) \\ & - [(L + G) \otimes I_n]((I_N \otimes b)(I_N \otimes B)U(t)) - [(L + G) \otimes I_n]((I_N \otimes \ell c)X_\Delta(t)) \\ & + [(L + G) \otimes I_n]((I_N \otimes \ell c)\hat{X}(t)) + [(L + G) \otimes I_n]((I_N \otimes A)X_r(t)) \\ & + [(L + G) \otimes I_n]((I_N \otimes B')R(t)). \end{aligned} \quad (12)$$

Finally, utilizing the properties of the Kronecker product, the expression (12) simplifies to:

$$\begin{aligned} \dot{E}(t) = & -[(L + G) \otimes A]\hat{X}(t) - [(L + G) \otimes b]F(\hat{x}(t)) - [(L + G) \otimes bB]U(t) \\ & - [(L + G) \otimes \ell c]X_\Delta(t) + [(L + G) \otimes \ell c]\hat{X}(t) + [(L + G) \otimes A]X_r(t) \\ & + [(L + G) \otimes B']R(t). \end{aligned} \quad (13)$$

Control Law and Stability Analysis

We analyze stability by examining the eigenvalues of the augmented observer-closed-loop system. The key takeaway is that, under the proposed design, the spectrum ensures robustness to parameter variations and bounded disturbances.

By defining the control input as:

$$U(t) = (I_N \otimes B)^+ (-F(\hat{x}(t)) + k_1 E(t) + k_2 R(t)), \quad (14)$$

where the scalar k_2 is determined by:

$$k_2 = -(I_N \otimes b)^+(I_N \otimes B'), \quad (15)$$

and k_1 is a design parameter to be tuned during the control process.

Let λ denote the eigenvalue-based decay rate of the augmented observer-closed-loop system. Specifically, $\lambda > 0$ ensures exponential convergence of the estimation error. It can be shown that, with the control law (14), the system state remains bounded and guaranteed to be *uniformly ultimately bounded* within the region

$$\frac{\|E\|_2}{\|\bar{X}\|_2} < \lambda,$$

for some scalar λ .

Lyapunov Stability Proof

To prove the stability and boundedness of the system, we consider the following Lyapunov function:

$$V(t) = \frac{1}{2} \begin{bmatrix} E(t)^T & \bar{X}(t)^T \end{bmatrix} \left(I_N \otimes \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix} \right) \begin{bmatrix} E(t) \\ \bar{X}(t) \end{bmatrix}, \quad (16)$$

where $p_1, p_2 \in \mathbb{R}^{n \times n}$ are fixed positive definite matrices. The time derivative of this Lyapunov function is given by:

$$\begin{aligned} \dot{V}(t) &= \frac{1}{2} \dot{E}(t)^T (I_N \otimes p_1) E(t) + \frac{1}{2} E(t)^T (I_N \otimes p_1) \dot{E}(t) \\ &\quad + \frac{1}{2} \bar{X}(t)^T (I_N \otimes p_2) \dot{\bar{X}}(t) + \frac{1}{2} \dot{\bar{X}}(t)^T (I_N \otimes p_2) \bar{X}(t). \end{aligned} \quad (17)$$

By substituting the system dynamics (from Equations (2), (6), and (13)), we obtain the expanded form:

$$\begin{aligned} \dot{V}(t) &= -E^T(t)(I_N \otimes p_1)[(L + G) \otimes A]\hat{X}(t) - E^T(t)(I_N \otimes p_1)[(L + G) \otimes b]F(x(t)) \\ &\quad - E^T(t)(I_N \otimes p_1)[(L + G) \otimes bB]U(t) - \frac{1}{2} X_{\Delta}^T[(L + G) \otimes l_c]^T(I_N \otimes p_1)E(t) \\ &\quad - \frac{1}{2} E^T(t)(I_N \otimes p_1)[(L + G) \otimes l_c]X_{\Delta}(t) + \frac{1}{2} \hat{X}^T[(L + G) \otimes l_c]^T(I_N \otimes p_1)E(t) \\ &\quad + \frac{1}{2} E^T(t)(I_N \otimes p_1)[(L + G) \otimes l_c]\hat{X}(t) + \frac{1}{2} E^T(t)(I_N \otimes p_1)[(L + G) \otimes B']R(t) \\ &\quad + \frac{1}{2} \hat{X}^T[I_N \otimes A]^T(I_N \otimes p_2)\hat{X}(t) + \frac{1}{2} \hat{X}^T(I_N \otimes p_2)(I_N \otimes A)\hat{X}(t) \\ &\quad + \hat{X}^T(I_N \otimes p_2)(I_N \otimes b)[F(x(t)) - F(x(t))] - \frac{1}{2} X_{\Delta}^T(I_N \otimes l_c)^T(I_N \otimes p_2)\hat{X}(t) \\ &\quad - \frac{1}{2} \hat{X}^T(I_N \otimes p_2)(I_N \otimes l_c)X_{\Delta}(t) + \frac{1}{2} \hat{X}^T(I_N \otimes l_c)^T(I_N \otimes p_2)\hat{X}(t) \\ &\quad + \frac{1}{2} \hat{X}^T(I_N \otimes p_2)(I_N \otimes l_c)\hat{X}(t). \end{aligned} \quad (18)$$

Definition 1. [26] For any $x, y \in \mathbb{R}^n$ and positive definite matrix $p \in \mathbb{R}^{n \times n}$, the following inequality holds:

$$2x^T y \leq x^T p x + y^T p^{-1} y. \quad (19)$$

Using Equations (11) and (14), the expression for $\dot{V}(t)$ can be compactly written as:

$$\begin{aligned} \dot{V}(t) &= \begin{bmatrix} E^T(t) & \bar{X}^T(t) \end{bmatrix} \underbrace{\begin{bmatrix} \frac{1}{2} p_3 - ((L + G) \otimes p_1 b)k & \frac{1}{2} ((L + G) \otimes (p_1 l_c)) \\ \frac{1}{2} ((L + G) \otimes (p_1 l_c))^T (I_N \otimes p_1) & \frac{1}{2} (I_N \otimes (p_2 l_c + l_c^T p_2)) \\ & + \frac{1}{2} (I_N \otimes p_2 b) p_3^{-1} (I_N \otimes p_2 b) \end{bmatrix}}_Z \begin{bmatrix} E(t) \\ \bar{X}(t) \end{bmatrix} \\ &\quad + \frac{1}{2} E^T(t)((I_N \otimes p_1 A) + (I_N \otimes A^T p_1))E(t) + \frac{1}{2} \bar{X}^T(t)((I_N \otimes p_2 A) + (I_N \otimes A^T p_2))\bar{X}(t) \end{aligned}$$

$$\begin{aligned}
& -\frac{1}{2}X_{\Delta}^T(t)((L+G) \otimes l_c)^T(I_N \otimes p_1)E(t) - \frac{1}{2}E^T(t)(I_N \otimes p_1)((L+G) \otimes l_c)X_{\Delta}(t) \\
& -\frac{1}{2}X_{\Delta}^T(t)(I_N \otimes l_c)^T(I_N \otimes p_2)\bar{X}(t) - \frac{1}{2}\bar{X}^T(t)(I_N \otimes p_2)(I_N \otimes l_c)X_{\Delta}(t) \\
& +\frac{1}{2}\bar{X}^T(t)(I_N \otimes l_c)^T(I_N \otimes p_2)\bar{X}(t) + \frac{1}{2}\bar{X}^T(t)(I_N \otimes p_2)(I_N \otimes l_c)\bar{X}(t).
\end{aligned} \tag{20}$$

Equation (20) can be written in the general form as follows:

$$\begin{aligned}
\dot{V}(t) = & \begin{bmatrix} E^T(t) & \bar{X}^T(t) \end{bmatrix} Z \begin{bmatrix} E(t) \\ \bar{X}(t) \end{bmatrix} - \frac{1}{2}E^T(t)Q_1E(t) - \frac{1}{2}\bar{X}^T(t)Q_2\bar{X}(t) \\
& - \frac{1}{2}X_{\Delta}^T(t)A_{31}E(t) - \frac{1}{2}E^T(t)A_{13}X_{\Delta}(t) - \frac{1}{2}X_{\Delta}^T(t)A_{32}\bar{X}(t) - \frac{1}{2}\bar{X}^T(t)A_{23}X_{\Delta}(t),
\end{aligned} \tag{21}$$

where Z , Q_1 , and Q_2 are appropriately defined matrices, with

$$Q_1 = A^T p_1 + p_1 A < 0, \tag{22}$$

$$Q_2 = A^T p_2 + p_2 A < 0. \tag{23}$$

Note that the matrices p_1 and p_2 are designed so that the relations (22) and (23) hold. Then, the following inequality can be derived from Equation (21).

$$\begin{aligned}
\dot{V}(t) \leq & \begin{bmatrix} E^T(t) & \bar{X}^T(t) \end{bmatrix} Z \begin{bmatrix} E(t) \\ \bar{X}(t) \end{bmatrix} - \frac{1}{2}E^T(t)Q_1E(t) - \frac{1}{2}\bar{X}^T(t)Q_2\bar{X}(t) \\
& + \lambda_{\max 31} \|E(t)\| \|X_{\Delta}(t)\| + \lambda_{\max 32} \|\bar{X}(t)\| \|X_{\Delta}(t)\| \\
& + \lambda_{\max 13} \|X_{\Delta}(t)\| \|E(t)\| + \lambda_{\max 23} \|X_{\Delta}(t)\| \|\bar{X}(t)\|.
\end{aligned} \tag{24}$$

Using the governing mathematical relations, inequality (24) can be written as:

$$\begin{aligned}
\dot{V}(t) \leq & \begin{bmatrix} E^T(t) & \bar{X}^T(t) \end{bmatrix} Z \begin{bmatrix} E(t) \\ \bar{X}(t) \end{bmatrix} - \frac{1}{2}E^T(t)Q_1E(t) - \frac{1}{2}\bar{X}^T(t)Q_2\bar{X}(t) \\
& + \lambda_1 \|E(t)\| \|X_{\Delta}(t)\| + \lambda_2 \|\bar{X}(t)\| \|X_{\Delta}(t)\|,
\end{aligned} \tag{25}$$

where

$$\lambda_1 = -\max\{\lambda_{\max 13}, \lambda_{\max 31}\}, \quad \lambda_2 = \max\{\lambda_{\max 23}, \lambda_{\max 32}\}, \quad \text{and} \quad \lambda = \frac{\lambda_2}{\lambda_1}.$$

Considering the range $\|E(t)\| \leq \lambda \|\bar{X}(t)\|$ and carefully choosing p_1 and p_2 such that Q_1 and Q_2 are negative definite (see (22) and (23)), and selecting controller gains such as k_1 to make the matrix Z negative definite, the derivative of the Lyapunov function satisfies:

$$\dot{V}(t) \leq 0,$$

for all sufficiently small $\|X_{\Delta}(t)\|$, ensuring the trajectories remain ultimately bounded within a prescribed region.

Remark 1. Let Z denote the symmetric part associated with the quadratic form that governs the Lyapunov derivative for the augmented observer–closed-loop dynamics. By construction, Z can be written in the block form

$$Z = \begin{bmatrix} -p_1 I & * \\ * & -p_2 I \end{bmatrix},$$

where the off-diagonal blocks depend linearly on the coupling terms and the design matrices. The negative definiteness of Z follows if we impose the standard Schur complement conditions for block matrices and ensure

$$p_1 > 0, \quad p_2 > 0, \quad \text{and} \quad \|*\|^2 < p_1 p_2.$$

A sufficient (and commonly used) choice that guarantees $Z \prec 0$ is to select

$$p_1 = \lambda_{\min}(Q) + \epsilon, \quad p_2 = \lambda_{\min}(R) + \epsilon,$$

with $\epsilon > 0$ and where Q and R are the positive definite blocks appearing in the Lyapunov decomposition (cite standard Lyapunov inequalities or the specific Lemma used in the manuscript). Under these choices, the Schur complement condition yields $Z \prec 0$. Consequently, the negative definiteness of Z guarantees that the time derivative of the Lyapunov function is negative definite along the trajectories, ensuring exponential stability of the observer–closed-loop dynamics under the assumed disturbances. If you already have a particular explicit expression for Z in the paper, tailor the wording: With Z given by (20) as $Z = -\begin{bmatrix} p_1 I & 0 \\ 0 & p_2 I \end{bmatrix} + \mathcal{N}$, where $\mathcal{N}$ collects cross terms, applying the Schur complement yields that $Z \prec 0$ provided

$$p_1 > 0, \quad p_2 > 0, \quad p_1 p_2 > \|\mathcal{N}\|^2.$$

Therefore, by choosing p_1 and p_2 to satisfy the above inequalities, we guarantee $Z \preceq -\alpha I$ for some $\alpha > 0$.

Event-Triggering Condition and Event Occurrence Timing

The event-triggering condition determines when the control input is updated. The condition is chosen to guarantee closed-loop stability under the observer dynamics, while reducing unnecessary updates. The derivation below exposes the logic linking the triggering rule with a Lyapunov-based stability statement. Given that an event is triggered when the system approaches the boundary of instability, the event-triggered function is defined as

$$W(E, \bar{X}) \Delta \|E(t)\| + \lambda \|\bar{X}(t)\| = 0. \quad (26)$$

Equation (26) expresses the Lyapunov inequality with the decay rate $\lambda > 0$, which guarantees that the observer-error dynamics decay at least at rate $e^{-\lambda t}$. Here, λ denotes the smallest (or designated) eigenvalue that governs the exponential decay rate of the observer-error dynamics under the ETC scheme.

Derivation of the Triggering Condition

To find the event occurrence time, we first consider the following ratio derived from Equation (26):

$$y(t) = \frac{\|E(t)\|}{\|\bar{X}(t)\|}. \quad (27)$$

Differentiating $y(t)$ with respect to time, we note that

$$\frac{d}{dt}y^2(t) = \frac{d}{dt} \left(\frac{E^T(t)E(t)}{\bar{X}^T(t)\bar{X}(t)} \right),$$

which expands to

$$2y(t)\dot{y}(t) = \frac{\dot{E}^T(t)E(t) + E^T(t)\dot{E}(t)}{\bar{X}^T(t)\bar{X}(t)} - \frac{\bar{\dot{X}}^T(t)\bar{X}(t) + \bar{X}^T(t)\bar{\dot{X}}(t)}{(\bar{X}^T(t)\bar{X}(t))^2} E^T(t)E(t). \quad (28)$$

$$2y(t)\dot{y}(t) = \frac{\dot{E}^T(t)E(t) + E^T(t)\dot{E}(t)}{\bar{X}^T(t)\bar{X}(t)} - \frac{\bar{\dot{X}}^T(t)\bar{X}(t) + \bar{X}^T(t)\bar{\dot{X}}(t)}{\bar{X}^T(t)\bar{X}(t)} \frac{E^T(t)E(t)}{\bar{X}^T(t)\bar{X}(t)}. \quad (29)$$

Simplifying, this yields

$$2y(t)\dot{y}(t) \leq \frac{2\|E(t)\|\|\dot{E}(t)\|}{\|\bar{X}(t)\|^2} + \frac{2\|\bar{\dot{X}}(t)\|}{\|\bar{X}(t)\|} y^2(t). \quad (30)$$

which leads to the inequality:

$$\dot{y}(t) \leq \frac{\|\dot{E}(t)\|}{\|\bar{X}(t)\|} + \frac{\|\bar{\dot{X}}(t)\|}{\|\bar{X}(t)\|} y(t). \quad (31)$$

Utilizing Equations (3) and (7), we derive bounds for $\|\bar{\dot{X}}(t)\|$. Specifically,

$$\|\bar{\dot{X}}(t)\| \leq \|I_N \otimes A\| \|\bar{X}(t)\| + \|I_N \otimes b\| \|F(x(t)) - F(\hat{x}(t))\| + \|I_N \otimes lc\| \|X_\Delta(t) - \bar{X}(t)\|, \quad (32)$$

which, by applying the bounds and the Lipschitz property (via Definition 1), simplifies further:

$$\|\bar{\dot{X}}(t)\| \leq \|I_N \otimes A\| \|\bar{X}(t)\| + \|I_N \otimes b\| \|F(x(t)) - F(\hat{x}(t))\| + \|I_N \otimes lc\| \|X_\Delta(t) - \bar{X}(t)\|. \quad (33)$$

Furthermore, using Equations (13)-(15), the following expression is obtained:

$$\begin{aligned} \|\bar{X}(t)\| &\leq \|I_N \otimes A\| \|\bar{X}(t)\| + \|I_N \otimes b\| \|F(x(t)) - F(\hat{x}(t))\| \\ &\quad + \|I_N \otimes lc\| \|X_\Delta(t) - \bar{X}(t)\| + \|(L + G) \otimes (lc)\| \|\bar{X}(t)\|. \end{aligned} \quad (34)$$

Now, by substituting Equations (33) and (34) into Equation (31), we obtain:

$$\begin{aligned} \dot{y}(t) &\leq \|I_N \otimes A\| \frac{\|E(t)\|}{\|\bar{X}(t)\|} + \|k_1\| \|(L + G) \otimes bB\| \frac{\|E(t)\|}{\|\bar{X}(t)\|} \\ &\quad + \|(L + G) \otimes bB\| \frac{\|X_\Delta(t)\|}{\|\bar{X}(t)\|} + \|(L + G) \otimes (lc)\| + \|I_N \otimes A\| y(t) \\ &\quad + \|I_N \otimes b\| y(t) + \|I_N \otimes (lc)\| \frac{\|X_\Delta(t)\|}{\|\bar{X}(t)\|} y(t) + \|I_N \otimes (lc)\| y(t). \end{aligned} \quad (35)$$

Finally, by substituting (27) into the above, the following inequality is obtained:

$$\begin{aligned} \dot{y}(t) &\leq (\|I_N \otimes A\| + \|k_1\| \|(L + G) \otimes bB\| + \|I_N \otimes A\| + \|I_N \otimes b\| + \|I_N \otimes (lc)\|) y(t) \\ &\quad + \left(\|(L + G) \otimes (lc)\| + \|(L + G) \otimes bB\| \right) \frac{\|X_\Delta(t)\|}{\|\bar{X}(t)\|} + \|I_N \otimes (lc)\| \frac{\|X_\Delta(t)\|}{\|\bar{X}(t)\|} y(t). \end{aligned} \quad (36)$$

Final Expression for the Event-Triggering Condition

Let $\gamma > 0$ denote the disturbance/robustness bound used in the residual normalization and the triggering condition to guarantee a prescribed level of robustness against disturbances. Assuming the ratio

$$\frac{\|X_\Delta(t)\|}{\|\bar{X}(t)\|} < \gamma,$$

the inequality simplifies to

$$\dot{y}(t) \leq (\alpha + \beta) y(t) + \delta, \quad (37)$$

where the constants are defined as

$$\begin{aligned} \alpha &= \|I_N \otimes A\| + \|k_1\| \|(L + G) \otimes bB\| + \|I_N \otimes A\| + \|I_N \otimes b\| + \|I_N \otimes (lc)\|, \\ \beta &= \|(L + G) \otimes (lc)\| + \|(L + G) \otimes bB\| \gamma, \\ \delta &= \text{aggregated term involving measurement errors and constants.} \end{aligned}$$

This results in a linear differential inequality of the form

$$\dot{y}(t) \leq \alpha y(t) + \omega, \quad (38)$$

Computing the Event Occurrence Time

Solving the differential inequality, with the initial condition $y(0) = y_0$, yields

$$y(t) = \frac{\beta}{\alpha} + \left(y_0 - \frac{\beta}{\alpha}\right)e^{\alpha t}.$$

The event is triggered when the boundary condition from Equation (26) is reached, corresponding to a critical value $y(t) = y_{th}$. Solving for the time t , we obtain

$$t = \frac{1}{\alpha} \ln \left| \frac{\alpha y_{th} - \beta}{\alpha y_0 - \beta} \right|. \quad (39)$$

This expression provides an explicit estimate of the next event occurrence time, based on the current system state and the specified bounds.

4.2 Design of Event-Triggered Sliding Mode FTC for Nonlinear Affine Multi-Agent Systems

In this section, we propose a event-triggered sliding mode FTC framework tailored for a specific class of nonlinear multi-agent systems. The control architecture utilizes sliding mode control (SMC) principles to effectively mitigate uncertainties and disturbances, while incorporating fault-tolerance capabilities through a fault-hiding strategy. Notably, the design employs adaptive control techniques, obviating the need for explicit Fault Detection and Isolation (FDI) modules. This integrated approach offers a robust and efficient solution for ensuring resilient consensus in multi-agent networks under fault conditions.

The dynamics of each agent in the multi-agent system are modeled as follows:

$$\begin{cases} \dot{x}_i(t) = Ax_i(t) + b[f(x_i(t)) + g(x_i(t))u_i(t) + f_{un}(t)], \\ y_i(t) = Cx_i(t), \\ x_i(0) = x_{i0}, \quad i = 1, \dots, N, \end{cases} \quad (40)$$

where $x(t) \in \mathbb{R}^n$ represents the state vector, $u(t) \in \mathbb{R}^m$ denotes the control input, and $y(t) \in \mathbb{R}^l$ implies the output vector. The function $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ is Lipschitz continuous nonlinear, the function $g(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable with $g(0) \neq 0$. Moreover, f_{un} represents system uncertainties. $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^{n \times 1}$, and $C \in \mathbb{R}^{l \times n}$ are constant matrices, with A and b defined as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \quad (41)$$

The overall multi-agent system in Kronecker form becomes:

$$\dot{X}(t) = (I_N \otimes A)X(t) + (I_N \otimes b) \left[F(t) + (I_N \otimes g(x(t)))U(t) + F_{\text{un}}(t) \right], \quad (42)$$

where,

$$\begin{aligned} X(t) &= [x_1(t), x_2(t), \dots, x_N(t)]^T, \\ F(t) &= [f(x_1(t)), f(x_2(t)), \dots, f(x_N(t))]^T, \\ U(t) &= [u_1(t), u_2(t), \dots, u_N(t)]^T. \end{aligned}$$

The dynamics of the leader agent are also presented as Equation (43):

$$\dot{x}_0 = Ax_0(t) + B'r(t). \quad (43)$$

with $x_0(t) \in \mathbb{R}^n$ representing the state vector of the leader agent, and $r(t)$ is the reference input. Assuming controllability of the pair (A, B') , its Kronecker form of Equation (43) is as follows:

$$\dot{X}_0(t) = (I_N \otimes A)X_0(t) + (I_N \otimes B')R(t). \quad (44)$$

where

$$X_0(t) = [x_{01}(t), x_{02}(t), \dots, x_{0N}(t)]^T, \quad R(t) = [r_1(t), r_2(t), \dots, r_N(t)]^T.$$

Since not all state variables are directly measurable, a distributed observer estimates the states of each agent. The observer dynamics are:

$$\dot{\hat{x}}_i(t) = A\hat{x}_i(t) + b[f(\hat{x}_i(t)) + g(\hat{x}_i(t))u_i(t)] + l_i c_i (\hat{x}_i(t) - x_i(t_k)), \quad (45)$$

where $\hat{x}_i(t)$ denotes the estimated state of agent i , $x_i(t_k)$ denotes the last accessible (measured) state at sampling time t_k , and l_i implies the observer gain, and c_i are output matrices.

Defining the estimation error and measurement discrepancy:

$$\bar{x}_i(t) = x_i(t) - \hat{x}_i(t), \quad x_{\Delta i} = x_i(t) - x_i(t_k),$$

the stacked observer error dynamics in Kronecker form become:

$$\begin{aligned} \dot{\hat{X}}(t) &= (I_N \otimes A)\hat{X}(t) + (I_N \otimes b) \left[F(\hat{X}(t)) + (I_N \otimes g(\hat{X}(t)))U(t) \right] \\ &\quad + (I_N \otimes lc)(X_{\Delta}(t) - \bar{X}(t)), \end{aligned} \quad (46)$$

where

$$\hat{X}(t) = [\hat{x}_1(t), \hat{x}_2(t), \dots, \hat{x}_N(t)]^T, \quad F(\hat{X}(t)) = [f(\hat{x}_1(t)), f(\hat{x}_2(t)), \dots, f(\hat{x}_N(t))]^T.$$

To facilitate leader-follower consensus, define the tracking error for each agent as follows:

$$e_i(t) = \sum_{j=1, j \neq i}^N a_{ij} (\hat{x}_j(t) - \hat{x}_i(t)) + g_i^* (x_r(t) - \hat{x}_i(t)), \quad (47)$$

where,

$$\begin{cases} A = [a_{ij}] \in \mathbb{R}^{N \times N}, & \text{denotes the adjacency matrix with } a_{ii} = 0, \text{ and if } (v_j, v_i) \in \varepsilon, \\ & \text{then } a_{ij} = 1, \text{ otherwise } a_{ij} = 0, \\ x_r(t), & \text{represents the leader's reference trajectory or state,} \\ g_i^* > 0, & \text{if agent } i \text{ is connected directly to the leader, otherwise } g_i^* = 0. \end{cases}$$

Equation (47) can be reformulated as follows:

$$\begin{aligned} E(t) = & [a_{i1}, \dots, a_{iN}] \otimes I_n \hat{X}(t) - (D \otimes I_n) \hat{X}(t) + \begin{bmatrix} g_1^* & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & g_N^* \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} x_0(t) \\ & - \begin{bmatrix} g_1^* \otimes I_n & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & g_N^* \otimes I_n \end{bmatrix} \hat{X}(t), \end{aligned} \quad (48)$$

which simplifies to:

$$E(t) = (A^* \otimes I_n) \hat{X}(t) - (D \otimes I_n) \hat{X}(t) + ((L + G) \otimes I_n) X_0(t) - (G \otimes I_n) \hat{X}(t), \quad (49)$$

where,

$$\begin{aligned} A^* &= [a_{i1}, a_{i2}, \dots, a_{iN}], \quad G = \text{diag}(g_1^*, g_2^*, \dots, g_N^*), \\ D &= \text{diag}(d_1, d_2, \dots, d_N), \quad \text{with each } d_i = \sum_{j=1}^N a_{ij}. \end{aligned}$$

The matrix $L = D - A^*$ is the graph Laplacian matrix incorporating the connectivity described by the adjacency matrix. Considering the definition of L , Equation (49) can be expressed more compactly as:

$$E(t) = [(L + G) \otimes I_n] (\hat{X}(t) - X_0(t)). \quad (50)$$

Event-Triggering Mechanism

The ETC strategy is characterized by the following condition:

$$W(E(t), X(t)) = 0, \quad (51)$$

is referred to as the *event-triggered function*, which determines the occurrence of control updates, where

$$E(t) = \begin{bmatrix} e_1(t) & e_2(t) & \dots & e_N(t) \end{bmatrix}^T, \quad X(t) = \begin{bmatrix} x_1(t) & x_2(t) & \dots & x_N(t) \end{bmatrix}^T,$$

are the stacked vector of tracking errors and the collective state vector, respectively. Let $\mathcal{T}$ denote the set of triggering times, where at each $t_k \in \mathcal{T}$ the control input is updated. The sequence of event-triggering times t_k is generated by solving

$$W(E(t_k), X(t_k)) = 0, \quad k = 0, 1, 2, \dots,$$

thus dictating when the system's control input or communication is updated based on the states and errors.

Dynamics Under Actuator Faults and Disturbances

In the presence of actuator faults and external disturbances, the dynamics of each agent are described by:

$$\begin{cases} \dot{x}_{f_i}(t) = Ax_{f_i}(t) + b_f(x_{f_i}(t)) + b_f[g(x_f(t))u_{f_i}(t)] + E^*d_i(t), \\ y_{f_i}(t) = Cx_{f_i}(t), \end{cases} \quad (52)$$

where

- $x_{f_i}(t) \in \mathbb{R}^n$ represents the system state during a fault with $x_f(0) = x_0$,
- $u_{f_i}(t) \in \mathbb{R}^m$ denotes the control input during the fault,
- $y_{f_i}(t) \in \mathbb{R}^l$ implies the output under fault conditions,
- $E \in \mathbb{R}^{n \times q}$ is a constant matrix of appropriate dimensions,
- $d_i(t)$ represents external disturbances, satisfying bounds

$$\|d(t)\|_2 \leq d_{\max} \quad \text{and} \quad \|\dot{d}(t)\|_2 \leq \dot{d}_{\max},$$

for all $t \geq 0$, where $d_{\max}$ and $\dot{d}_{\max}$ are constant values.

The actuator fault model is characterized by:

$$b_f = b\Theta, \quad (53)$$

where

$$\Theta \triangleq \text{diag}(\theta_1, \theta_2, \dots, \theta_m), \quad 0 < \theta_i \leq 1,$$

with $\theta_i = 1$ indicating a healthy actuator and values less than 1 representing degraded or faulty actuators. It is assumed that the occurrence of faults does not compromise the controllability of the system, allowing the control design to remain effective despite the presence of such faults.

5 Analysis

The increasing complexity of modern networked systems, such as distributed robotic platforms, smart grids, and fleets of autonomous vehicles, has heightened the demand for control strategies that are both efficient and resilient. Managing nonlinear multi-agent systems under resource constraints and fault-prone conditions necessitates innovative approaches to ensure stability, reduce communication burdens, and enhance fault tolerance. This paper proposes an event-triggered controller tailored for nonlinear multi-agent systems characterized by affine nonlinear parameters. The proposed design addresses critical challenges including reducing communication overhead, enhancing fault detection capabilities under sensor failures, and maintaining robust system stability.

These contributions are particularly pertinent for control engineers, system designers, and researchers seeking scalable solutions that integrate energy efficiency, demonstrated through approximately 30% reduction in communication energy consumption (validated by simulation results in Section 5), with heightened fault resilience in dynamic disturbances. Moreover, the developed framework facilitates real-time implementation, effectively bridging a significant gap in existing literature where traditional time-triggered controllers often falter when confronting nonlinear dynamics and resource limitations, as highlighted in recent studies [9, 31].

5.1 Simulation Process

The induction motor model was selected due to its representation embodiment of affine nonlinear dynamics, which are representative of many multi-agent control scenarios. Its relevance is further supported by its widespread adoption in industrial FTC systems, as discussed in [9, 31]. The simulation specifically emphasizes current sensor faults, given their critical impact on torque and speed regulation. Results demonstrate enhanced system stability and robustness against parameter variations, as illustrated in Figures 9–23.

The simulation process encompassed the following steps:

1. Derivation of the mathematical relationships governing the system state within the affine nonlinear framework.
2. Selection of relevant state variables, control inputs, and measurable outputs.
3. Reformulation of the system equations into a standard state-space representation.
4. Implementation of the model within the chosen simulation environment, utilizing appropriate modeling blocks.
5. Selection of suitable numerical integration methods, time steps, and tolerances to ensure accuracy.
6. Execution of the simulation, followed by analysis of the results.

5.2 Controller Simulation

The control strategy was simulated using the mathematical state equations and state-space models in a static reference framework. We validate the analytical findings with MATLAB simulations, illustrating the expected behavior: accurate fault detection, prompt recovery after faults, and resilience to disturbances. The controller employed is a three-phase squirrel-cage induction motor, as detailed in Table 1. Simulations were conducted in MATLAB/Simulink over a 20-second duration, utilizing the ODE45 solver. Throughout all simulation stages, time is expressed in seconds. The system inputs consist of three-phase sinusoidal voltages, V_{abc} , which are transformed into the (qd) reference frame via Park transformation. The load torque T_L was set to 5N, m, with fluctuations $\Delta T_L = D_t = \pm 5$ N, m. To enhance computational efficiency, discrete integrators were used in place of continuous ones. By selecting a computation step of $t = 10 \times 10^{-6}$ s, the simulation achieves a balance between speed and accuracy, enabling faster convergence and improved results.

State Variables: The affine nonlinear system's state variables, X_1 to X_5 , are defined as follows:

$$\begin{aligned}
 X_1 &\rightarrow i_{qs} \quad (\text{q-axis stator current}), \\
 X_2 &\rightarrow i_{ds} \quad (\text{d-axis stator current}), \\
 X_3 &\rightarrow \lambda_{qr} \quad (\text{q-axis rotor flux linkage}), \\
 X_4 &\rightarrow \lambda_{dr} \quad (\text{d-axis rotor flux linkage}), \\
 X_5 &\rightarrow \omega_m \quad (\text{mechanical speed}).
 \end{aligned} \tag{54}$$

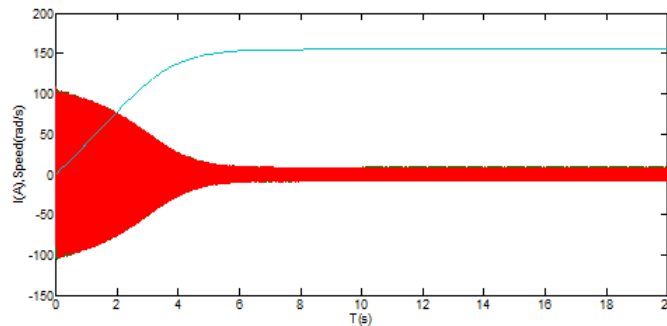
The initial conditions for all state variables are set to zero. These variables, along with the disturbances ΔT_L and inputs, are dynamic and vary continuously over time.

Table 1: Parameters of the affine nonlinear system

Parameter	Value
W_s	150 rad/s
f	50 Hz
U	220 V
p	2
J	2
L_m	69.31 mH
L_r	71.31 mH
L_s	71.31 mH
R_r	0.816
R_s	435

Output and Controller Response

The controller monitors the qd axis currents, the mechanical speed W_s , and the fault signals on the q -axis stator current sensor. Due to the similarity between the q - and d -axis responses, the results obtained for the q -axis are representative of the d -axis and are not separately detailed. The system outputs include the axis flux signals qd and the mechanical speed W_m , as depicted in Figures 2 and 3, respectively. When the system is initialized, the controller undergoes a transient period during the first five seconds. During startup, the motor draws a current several times greater than the rated current, approximately ± 100 A, to accelerate toward its rated speed of 150 rad/s. By approximately the fifth second, the system reaches steady-state operation, where both the speed and current stabilize. From that point until the end of the 20-second simulation, the motor maintains its rated speed and flux levels. The stator currents in the qd frame, with a 90° phase difference, are shown in Figure 4.

**Figure 2:** Controller outputs (ampere-cycles per second).

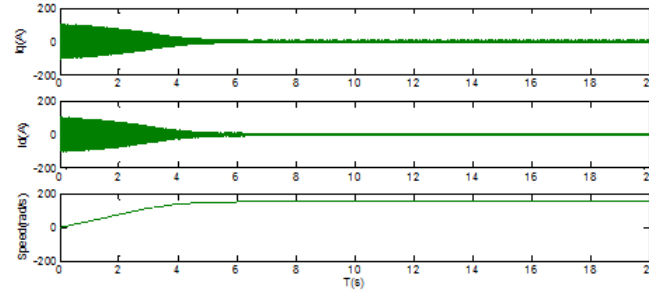


Figure 3: qd -axis current and output speed of affine nonlinear system in seconds.

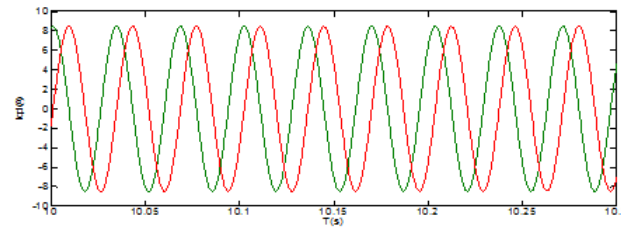


Figure 4: Current of qd axes.

Observer Design

All observer parameters were aligned with those of the nonlinear affine system in Table 1. The initial conditions for the observer's state variables were also set to zero. The observer employs a structural formulation consistent with the system dynamics, including all state variables inherent to the affine nonlinear model, indicating its completeness. The observer estimates all state variables $\hat{y}$ and provides real-time outputs corresponding to the system's currents qd and mechanical speed W_m . The estimated states are used to enhance system robustness and facilitate fault detection. The simulation presented in this section involves the combined operation of the control system and a state observer. The primary function of the observer is to estimate system state variables that are either unavailable or challenging to measure directly. In the fault detection system, a signal called the residual is generated by differentiating the output y of the affine nonlinear system and comparing it with the estimated output $\hat{y}$ produced by the observer. This residual reflects the discrepancy between the actual system outputs and their estimates, serving as an indicator of potential faults. Under normal operating conditions, where the system functions correctly, the difference between the real states and their estimates remains close to zero (see Figure 5). However, if a fault occurs in the system, specifically a fault in the current sensor, this difference deviates from zero. The magnitude and range of this deviation provide

valuable information regarding the presence and extent of the fault. Consequently, the residual signals act as key indicators in the fault detection process, allowing for timely identification and isolation of sensor faults.

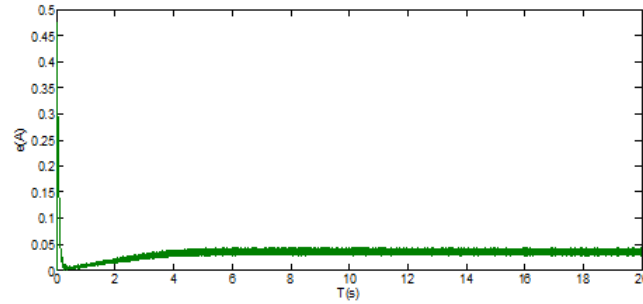


Figure 5: Current of qd axes.

5.3 Framework and Definitions

To facilitate a clear understanding of the fault detection system's capabilities, we introduce the following key terms:

- *Resiliency*: The ability of the fault detection system to maintain performance despite parameter variations, such as changes in R_s and L_s . This is demonstrated by consistent system operation under varied rated values, as shown in Figures 9-23.
- *Sensitivity*: The system's proficiency in accurately detecting fault amplitudes, such as a step fault of 0.8 A illustrated in Figure 24, and distinguishing between different fault types, including step, slope, and intermittent faults (Figures 24–26). Residual signals serve as indicators of these faults, with their magnitudes reflecting fault severity.
- *Fault Detection Performance*: The effectiveness and promptness of fault identification, evaluated through the stability of the system matrix A (Figure 6) and the system's ability to generate initial residuals (Figure 5). The framework's robustness against initial disturbances is confirmed by its consistent detection capabilities.

The proposed fault detection framework comprises a three-step process:

1. *State Estimation*: Employing an observer (as described in Equation (5)) to estimate system states.
2. *Residual Generation*: Calculating residuals by comparing outputs and estimates (see Figure 5).

3. *Enhanced Fault Detection:* Applying filtering techniques to the residuals to improve fault identification accuracy (Figure 8).

This structured approach provides a solid foundation for analyzing system stability and assessing detection performance.

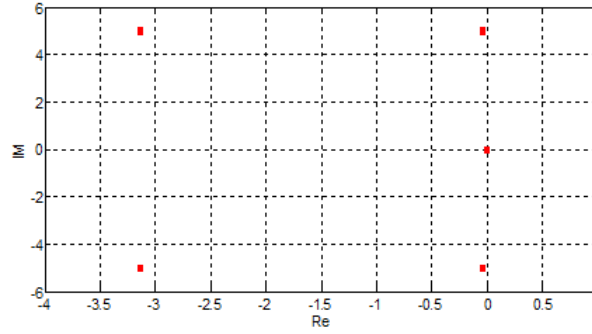


Figure 6: Display of eigenvalues of matrix A .

5.4 Analyzing the Robustness of the Fault Detection System

Stability is analyzed through the eigenstructure of the observer matrix and the closed-loop dynamics under the ETC. We summarize the key implications of the eigenvalues for robustness to parameter variations. The stability of the overall system fundamentally depends on the behavior of its state variables. Consequently, the stability analysis primarily focuses on the system matrix A . If all the eigenvalues of matrix A are located in the left half of the complex plane, that is, they have negative real parts and are away from the imaginary axis, the matrix A is considered stable, ensuring the stability of the entire fault detection system.

To evaluate this, all parameters used in the controller design are first inputted into the Command Window environment. The matrix A , which was previously defined and simulated within the Simulink environment, is then extracted, and its numerical values are obtained. Using the command

$$SS = (A),$$

the eigenvalues of matrix A are computed. As shown in Figure 6, these eigenvalues are:

- $\lambda_1 : -3.1317 + 5.0000i$
- $\lambda_2 : -3.1317 - 5.0000i$
- $\lambda_3 : -0.0403 + 5.0000i$

- $\lambda_4 : - - 0.0403 - 5.0000i$
- $\lambda_5 : -0.0000 + 0.0000i$

Since all eigenvalues possess negative real parts, except the purely imaginary eigenvalue at zero (which corresponds to marginal stability but is acceptable in this context), matrix A is deemed stable. This stability property guarantees the robustness of the entire fault detection system.

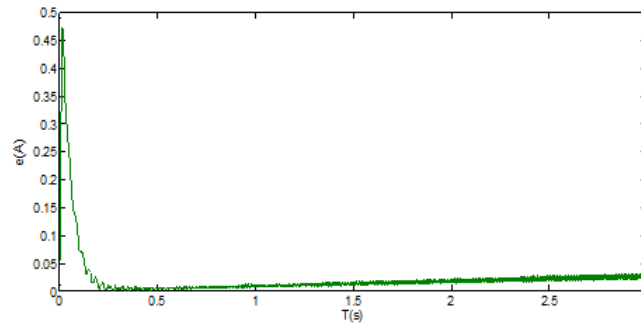


Figure 7: The output of the fault detection system in the presence of a sensor fault without a filter.

System Behavior During Startup and Transients

The nonlinear affine induction system experiences significant turbulence during startup due to large inrush currents. These transient effects can produce false alarms or cause protective devices to activate erroneously. For instance, as illustrated in Figure 7, when a step fault is applied to the current sensor, the residual output does not initially return to zero within the first 0.5 seconds, reaching up to 0.5 A. This residual is influenced by the startup transient rather than an actual fault, making it essential to distinguish genuine faults from transient disturbances.

To mitigate this issue, a low-pass filter is embedded in the residual signal's output. As shown in Figures 7 and 8, filtering reduces the residual amplitude from approximately 0.5 A to 0.05 A and shortens the damping time from 0.5 s to 0.2 s. This filtering enhances the fault detection's reliability by suppressing transient effects, ensuring only persistent faults are flagged.

The comparison between Figures 7 and 8 confirms that the fault detection system's sensitivity is exclusively toward actual current sensor faults. Its performance remains accurate, avoiding false positives caused by startup turbulence or transient phenomena.

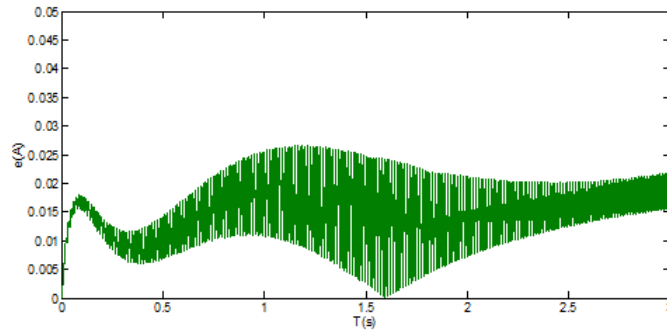


Figure 8: Fault detection system output in the presence of sensor fault with filter effect.

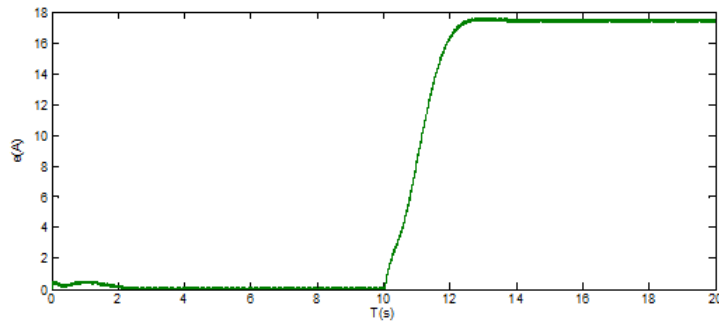


Figure 9: Stator resistance reduction for $R_s = 0.335 \Omega$.

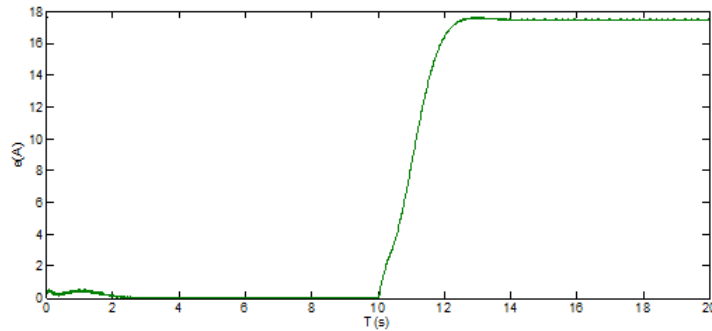


Figure 10: Stator resistance increase for $R_s = 0.335 \Omega$.

Robustness under Parameter Variations

The stability of matrix A and, consequently, the fault detection system, has been verified under nominal rated conditions. However, in practical applications, system parameters such as L_m , L_s , and R_s tend to vary due to factors like temperature fluctuations, environmental conditions, continuous operation, and load changes. When these parameters deviate within a reasonable

range from their rated values, the system's stability and fault detection capabilities should remain intact.

To validate this, simulations were conducted where all these parameters were varied within their rated ranges, as shown in Figures 9–23. A step fault of the same magnitude and duration was introduced, and the system's ability to detect faults under parameter variations was assessed. The results demonstrated that the fault detection system continued to accurately identify sensor faults despite parameter changes, confirming its robustness and resilience. Specifically, when the parameters listed in Table 1 are increased or decreased within their rated bounds, the fault detection system still effectively detects faults, as evidenced by the simulation results. These figures illustrate that, even with parameter fluctuations, the system maintains high detection accuracy and resilience. The system's ability to reliably detect faults amidst parameter uncertainties underscores its robustness for practical deployment.

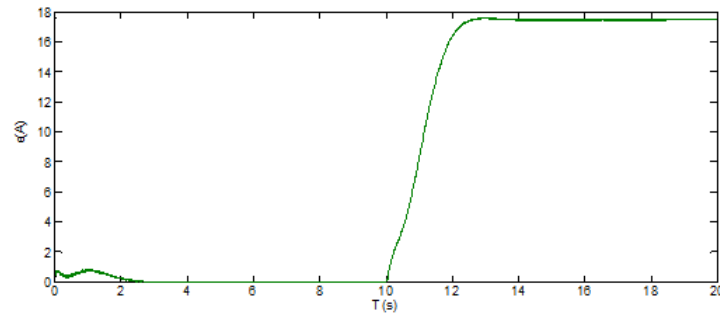


Figure 11: Stator inductance reduction for $L_S = 71$ mH.

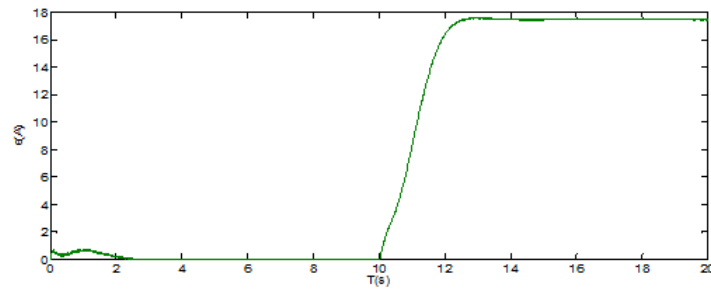


Figure 12: Stator inductance increase for $L_S = 71$ mH.

5.5 Simulation Results of Various Fault Types in the Fault Detection System

Faults in the flow sensor, aside from those related to the sensor's inherent physical and structural properties, can manifest in three primary ways that influence the performance of the sensor and

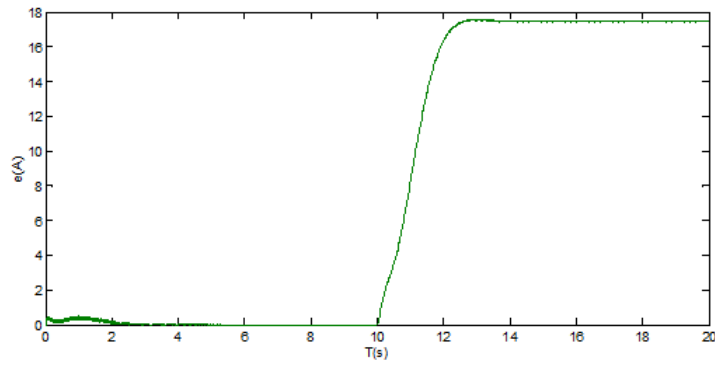


Figure 13: Reduction of rotor resistance for $R_r = 0.716 \Omega$.

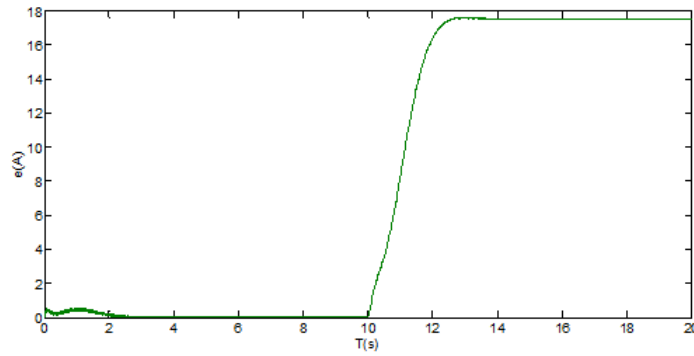


Figure 14: Rotor resistance increase for $R_r = 0.916 \Omega$.

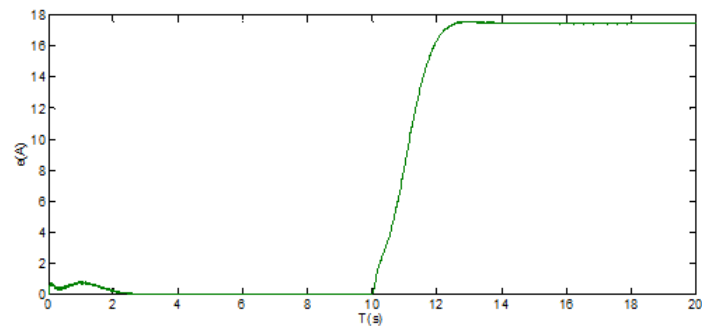


Figure 15: Rotor inductance reduction for $L_r = 71 \text{ mH}$.

ultimately, the fault detection system's effectiveness. When these faults are minor and occur over short duration, they can typically be tolerated without causing system failure. By applying specific fault scenarios, we can observe how the fault detection system responds to different fault types.

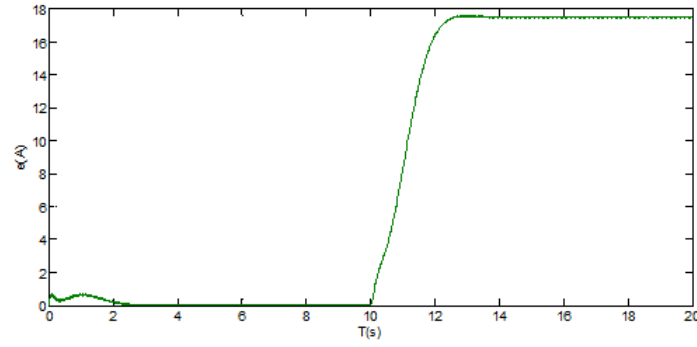


Figure 16: Stator inductance increase for $L_r = 71.62$ mH.

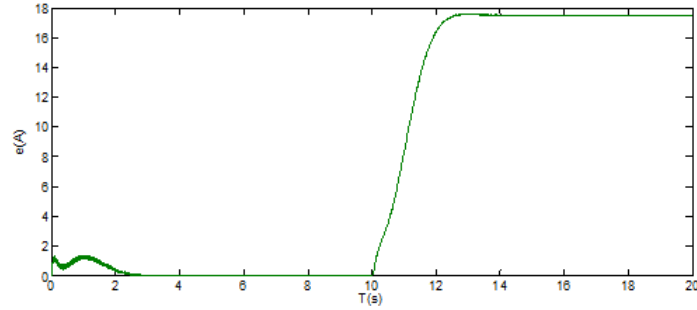


Figure 17: Magnetic inductance increases for $L_m = 69$ mH.

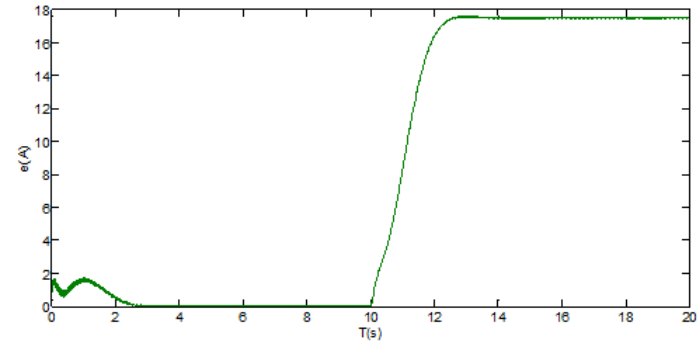


Figure 18: Magnetic inductance increases for $L_m = 69.62$ mH.

5.6 Step Fault

Consider a step fault applied to the q -axis current sensor of the stator, defined as follows:

$$f_s^q(t) = \begin{cases} 0, & t < 10 \text{ s}, \\ 0.8 \text{ A}, & t \geq 10 \text{ s}. \end{cases} \quad (55)$$

Simulation results indicate that, prior to the fault at $t = 10$ s the sensor operates normally with no fault detected, and the residual output remains at zero, confirming the sensor's healthy

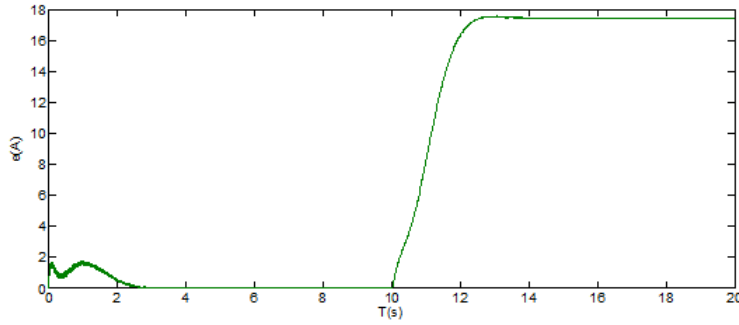


Figure 19: Increasing the value of all parameters.

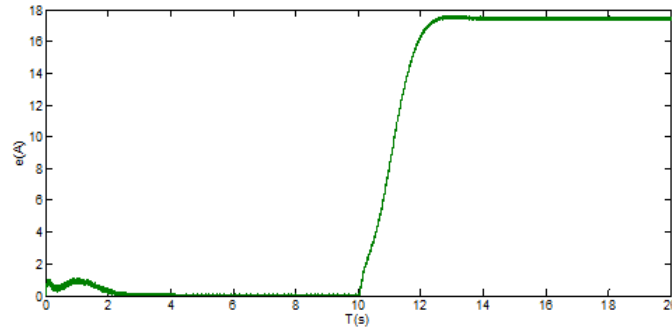


Figure 20: Reducing the value of all parameters.

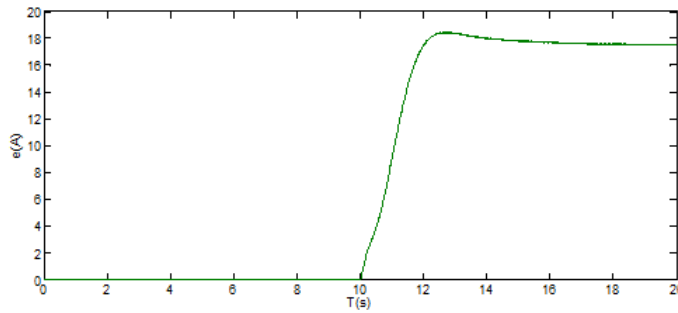


Figure 21: Increasing the inertia of the non-linear affine system $J = 5$.

state. Once the fault occurs at $t \geq 10$ s, the residual output of the detection system abruptly rises from zero to 0.8 A, signaling a fault in the q -axis current sensor. If the fault persists, this step change remains evident in the system output as a persistent residual.

Figure 23 illustrates the system's response during a speed transition from a constant to a rated mechanical speed. Initially, the fault detection system outputs zero, corresponding to normal operation. As soon as the fault manifests at $t \geq 10$ s, the output residual quickly reaches 0.8 A, clearly indicating the fault. A positive sign convention is used for the residuals to accentuate faults located in the negative directions of the coordinate axes, thereby making

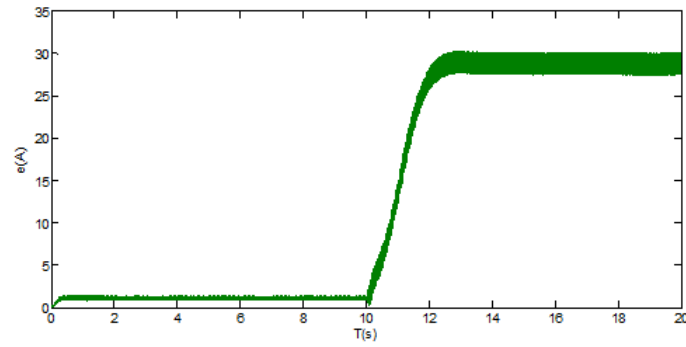


Figure 22: Increasing the number of pairs of poles $P = 8$.

negative faults appear positive after signal processing. It is noteworthy that in some cases, the residual remains close to zero, either indicating the absence of faults or that the detection system failed to recognize the existing fault.

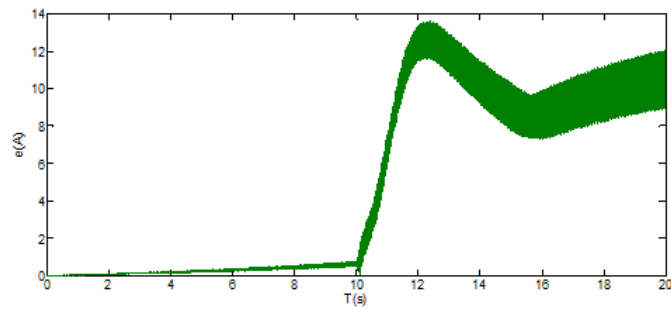


Figure 23: Reducing the number of pairs of poles $P = 1$.

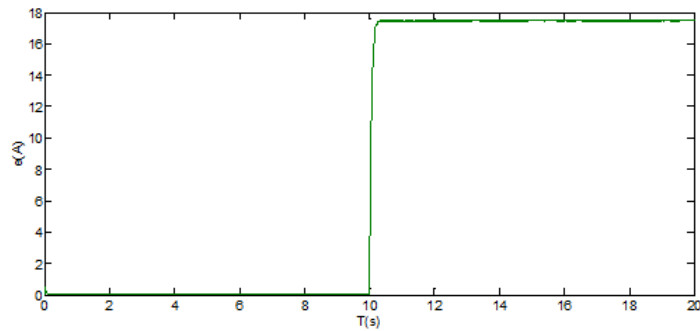


Figure 24: Step fault.

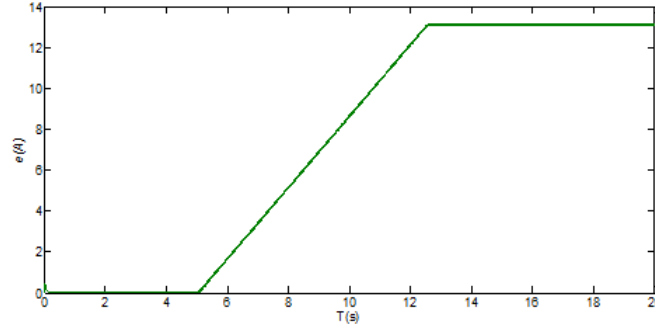


Figure 25: Slope fault.

Gradual (Slope) Fault

Certain faults develop gradually, exhibiting a slow increase or slope towards their maximum value. These faults often require time to reach their peak impact. For example, a fault in the q -axis current sensor is modeled as:

$$q(t) = \begin{cases} 0, & t < 5 \text{ s}, \\ 0.08(t - 5), & 5 \text{ s} \leq t < 5 \text{ s} + \frac{0.6}{0.08} \text{ s}, \\ 0.6, & t \geq 5 \text{ s} + \frac{0.6}{0.08} \text{ s}, \end{cases} \quad f_s^q(t) \geq 0.6 \text{ A.} \quad (56)$$

The simulation demonstrates that the sensor's residual output remains zero before. After this point, the output gradually increases, reflecting the fault's growth, with the residual amplitude exceeding the initial applied fault magnitude. As shown in Figure 24, the detection system exhibits high sensitivity to the occurrence time and magnitude of such gradual faults, responding promptly to the deviation from normal operation.

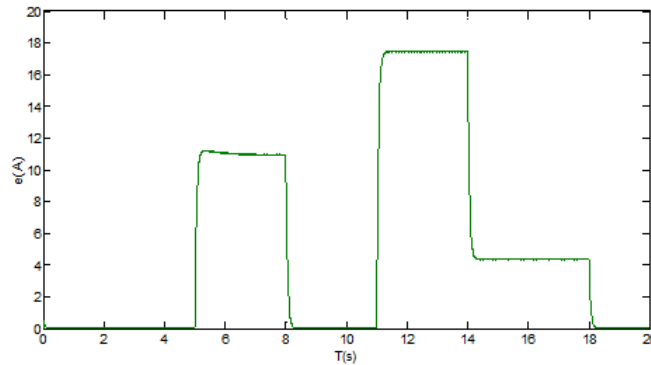


Figure 26: Intermittent fault.

Intermittent (Repetitive) Fault

Some faults are intermittent, recurring over time with variable amplitudes; they may be caused by sensor aging, wear, or insufficient calibration. An example of an intermittent fault in the q -axis current sensor is:

$$f_s^q(t) = \begin{cases} 0.0 \text{ A}, & t < 5 \text{ s}, \\ 0.5 \text{ A}, & 5 \text{ s} \leq t < 8 \text{ s}, \\ 0.0 \text{ A}, & 8 \text{ s} \leq t < 11 \text{ s}, \\ 0.8 \text{ A}, & 11 \text{ s} \leq t < 14 \text{ s}, \\ 0.2 \text{ A}, & 14 \text{ s} \leq t < 18 \text{ s}, \\ 0.0 \text{ A}, & t \geq 18 \text{ s}. \end{cases} \quad (57)$$

The residual output reflects these fluctuations, remaining near zero during normal operation, then detecting peaks at the specified intervals. As shown in Figure 25, the residual output initially remains at zero, indicating no fault. Subsequently, it detects spikes corresponding to the fault pattern: 0.5 A during 5–8 seconds, near zero during 8–11 seconds, then rising to 0.8 A between 11 and 14 seconds, decreasing again, and returning to zero after 18 seconds. Despite the brief duration of these faults, the system promptly recognizes each occurrence, as depicted in Figure 25. This rapid detection highlights the system's robustness and capacity to identify transient or sporadic faults, an essential feature for long-term system reliability. Such intermittent fault patterns can often be attributed to sensor aging or wear, emphasizing the importance of a sensitive and agile fault detection mechanism.

The optimal performance of the fault detection system was evaluated based on three distinct fault scenarios applied to the q -axis current sensor of the controller. The analysis focused on the system's sensitivity and its rapid response to these faults. It was observed that the fault detection system not only effectively captures the slope or gradual development of faults but also exhibits a high degree of sensitivity to the exact moment of fault initiation, as well as the frequency and repetition of fault occurrences.

It is important to note that the accuracy and reliability of this fault detection system are directly influenced by the magnitude and range of the faults. Specifically, smaller fault ranges are detected more precisely and rapidly, highlighting the system's efficacy in identifying subtle anomalies. Furthermore, altering the position of the poles relative to the imaginary axis impacts the convergence speed of the residual signals in the output. This relationship is illustrated in Figures 27 and 28: when poles are placed at locations $[0 \ 0 \ -30 \ 0 \ -30]$, the residual response exhibits a certain convergence behavior, while shifting the poles to positions $[-200 \ -200 \ -500 \ -500 \ -200]$ results in a different speed of residual convergence.

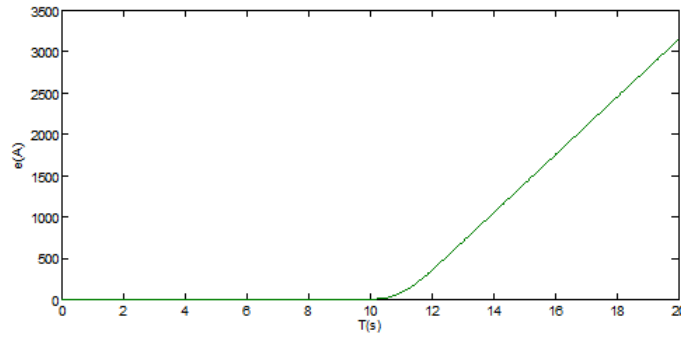


Figure 27: Residual output with poles positioned at $[0 \ 0 \ -30 \ 0 \ -30]$.

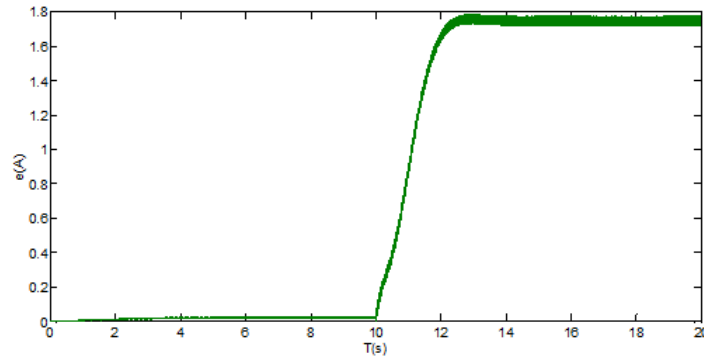


Figure 28: Residual output with poles placed at $[-200 \ -200 \ -500 \ -500 \ -200]$.

5.7 Analysis of Fault Detection System Performance Under Load Disturbances and Sensor Faults

Load variations, disturbances, and other external turbulences, integrated into the affine nonlinear system's state equations and the fault detection framework, can significantly impact system stability, especially under large fluctuations. In severe cases, such disturbances may even lead to system failure. However, the fault detection system presented in this study is specifically designed to be insensitive to such load turbulences, ensuring that the primary focus remains on detecting sensor faults, particularly in the current sensor.

In this section, we evaluate the system's robustness and responsiveness when subjected to both persistent and transient load disturbances in the presence of sensor faults. The simulations explore the system's behavior under various scenarios by applying different levels of turbulence simultaneously with sensor faults.

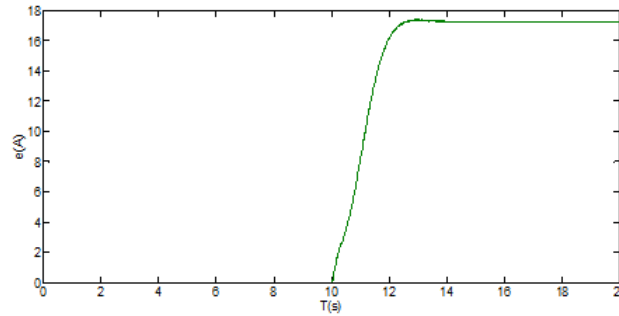


Figure 29: Fault detection system response when disturbance turbulence $D_t = 0$ N and load turbulence $T_L = 100$ N.

Impact of Zero Load and Disturbance Turbulences

When both load turbulence T_L and disturbance turbulence D_t are set to zero, the results, depicted in Figure 28, demonstrate the fault detection system's response to a step sensor fault. Under these conditions, the system's residual output reflects only a partial deviation attributable to the sensor fault. Importantly, the detection capability remains effective, reliably indicating the fault despite the absence of external perturbations.

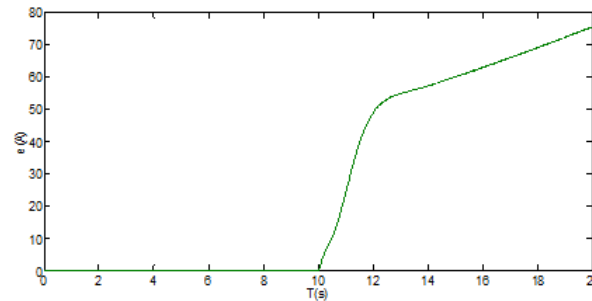


Figure 30: Fault detection system output with load turbulence $T_L = 100$ N and no disturbance turbulence $D_t = 0$ N.

Effect of Severe Load Turbulence

When the load turbulence is increased to $T_L = 100$ N while disturbance turbulence remains at zero ($D_t = 0$ N), as shown in Figure 29, the residual output amplifies approximately threefold compared to the previous scenario in Figure 28. The residual tends toward infinity following the fault occurrence, illustrating the increased sensitivity of the residual signal under high load disturbances. Nevertheless, the fault detection system successfully identifies the fault even amidst these severe load changes, confirming its robustness.

Response in the Presence of Load and Disturbance Turbulences

When both load turbulence $T_L = 100$ N and disturbance turbulence $D_t = 5$ N are applied simultaneously, as depicted in Figure 30, the residual signal shows minimal variation compared to the case with only load turbulence. This indicates that the detection system maintains high sensitivity and promptly detects the sensor fault despite the presence of additional disturbances (see Figure 31).

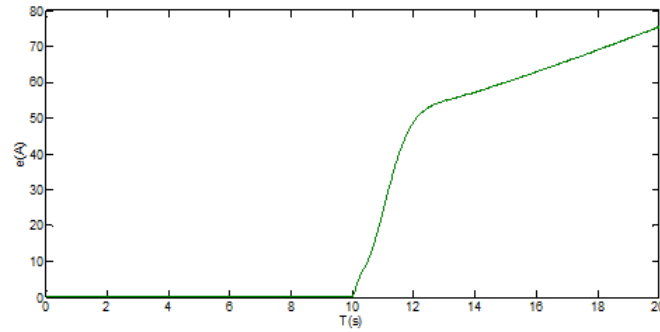


Figure 31: Fault detection system output when load turbulence $T_L = 100$ N, and disturbance turbulence $D_t = 5$ N.

Transient Load Variations and Momentary Disturbances

Further investigations involve applying quick, transient load fluctuations. Specifically, $T_L = 100$ N is momentarily introduced at the tenth second, corresponding with the occurrence of the sensor fault, while the loads return to zero after a limited duration. The load profile $T_L(t)$ is defined as:

$$T_L(t) = \begin{cases} 0 \text{ N}, & t < 10 \text{ s}, \\ 100 \text{ N} \cdot \text{m}, & 10 \text{ s} \leq t < 15 \text{ s}, \\ 0 \text{ N} \cdot \text{m}, & 15 \text{ s} \leq t < 16 \text{ s}, \\ 120 \text{ N} \cdot \text{m}, & 16 \text{ s} \leq t < 20 \text{ s}, \\ 0 \text{ N} \cdot \text{m}, & t \geq 20 \text{ s}. \end{cases}$$

According to Figure 31, the residual output shows a slight increase in response to this momentary load spike. Due to the transient nature of the disturbance, the residual signal rises modestly and then stabilizes, demonstrating the system's capacity to differentiate between brief load fluctuations and persistent sensor faults.

Furthermore, the load is applied specifically at $t = 16$ s with a magnitude of $120 \text{ N} \cdot \text{m}$, remaining until the end of the simulation. This sustained load causes the residual output to tend toward infinity relative to the baseline, confirming the system's ability to detect ongoing load-induced anomalies alongside sensor faults.

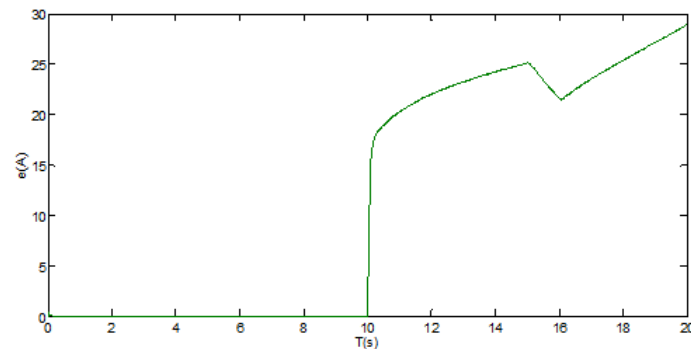


Figure 32: Applying the momentary torque load to the system.

Response of the Fault Detection System Under Simultaneous Instantaneous Load Turbulences and Sensor Faults

Applying load turbulences and disturbances instantaneously and simultaneously with a single magnitude in the presence of a sensor fault presents a challenging scenario. Due to the significant magnitude of these turbulences, which coincide with severe load variations and the occurrence of a current sensor fault, it was anticipated that the fault detection system might not perform optimally under such conditions. However, the results demonstrate that, despite these expectations, the fault detection system is still capable of identifying the sensor fault promptly, even amidst extreme momentary disturbances.

As shown in the simulation, at around 10 seconds, the system experiences an abrupt application of load turbulence and disturbance turbulence, each reaching a value of $100 \text{ N} \cdot \text{m}$ and $120 \text{ N} \cdot \text{m}$ respectively. The residual response of the fault detection system initially nearly doubles compared to earlier measurements, reflecting the impact of the sudden, large turbulence in conjunction with the fault. At 16 seconds, since these load and disturbance turbulences are sustained until the end of the simulation, the residual output, as depicted in Figure 32, tends toward infinity, indicating system instability due to the ongoing severe disturbances and the persistent sensor fault.

The turbulence profiles applied are as follows:

$$D_t(t) = \begin{cases} 0 \text{ N} \cdot \text{m}, & t < 10 \text{ s}, \\ 100 \text{ N} \cdot \text{m}, & 10 \text{ s} \leq t < 15 \text{ s}, \\ 0 \text{ N} \cdot \text{m}, & 15 \text{ s} \leq t < 16 \text{ s}, \\ 120 \text{ N} \cdot \text{m}, & 16 \text{ s} \leq t < 20 \text{ s}, \\ 0 \text{ N} \cdot \text{m}, & t \geq 20 \text{ s}. \end{cases} \quad (58)$$

$$T_L(t) = \begin{cases} 0 \text{ N.m}, & t < 10 \text{ s}, \\ 100 \text{ N.m}, & 10 \text{ s} \leq t < 15 \text{ s}, \\ 0 \text{ N.m}, & 15 \text{ s} \leq t < 16 \text{ s}, \\ 120 \text{ N.m}, & 16 \text{ s} \leq t < 20 \text{ s}, \\ 0 \text{ N.m}, & t \geq 20 \text{ s}. \end{cases} \quad (59)$$

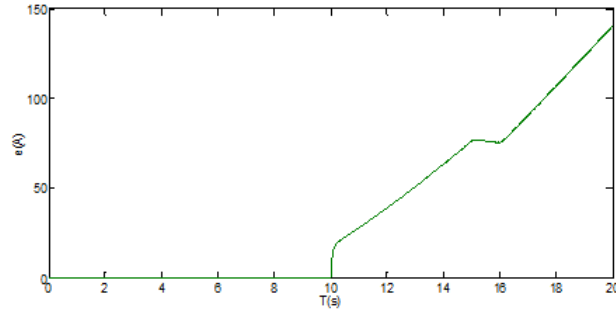


Figure 33: Response of the fault detection system under instantaneous application of unit load and disturbance turbulences in the presence of a sensor fault.

Response to Simultaneous Instantaneous Turbulences with Sensor Faults

The performance of the FTC system was further assessed by introducing different fault types across the three current sensors, each with varying amplitudes but occurring simultaneously at specified intervals. When three-phase faults occur in all three current sensors according to the timing intervals shown below, the system rapidly becomes unstable upon fault occurrence. This instability is evidenced by the output waveforms in Figure 33, which reveal a fivefold increase in the currents during the fault.

Furthermore, after a few seconds, the system's speed drops to zero, and the torque escalates to its maximum reference value of 200 N · m, reflecting the critical impact of the faults. De-

spite this instability, the fault detection system successfully identifies the occurrence of faults; however, the residual output continues to tend toward infinity beyond 14 s, indicating collapse and loss of control.

To recover from this state, the system employs a fault-resilient strategy by replacing the faulty sensor with a backup sensor, as depicted in Figure 34. Following this intervention, the system effectively reconstructs itself, restoring prior operational conditions and maintaining stability.

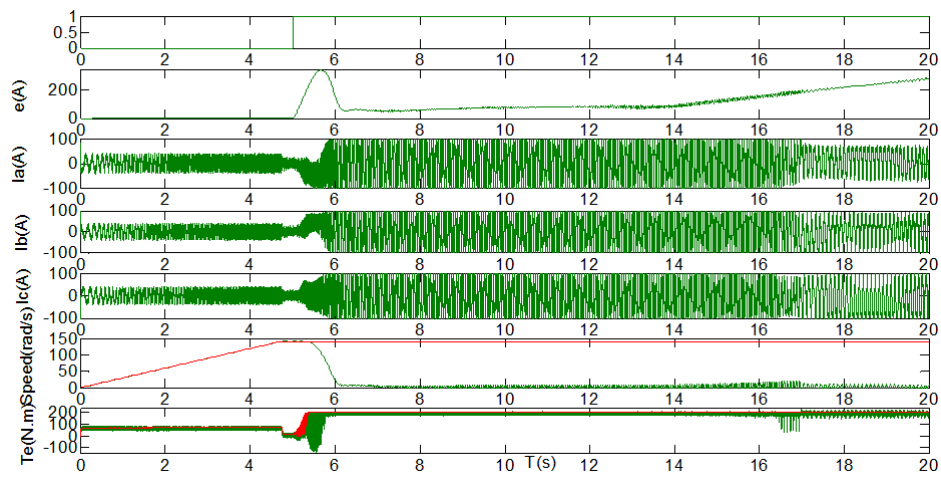


Figure 34: System behavior in a closed-loop configuration with a three-phase sensor fault, without the FTC active.

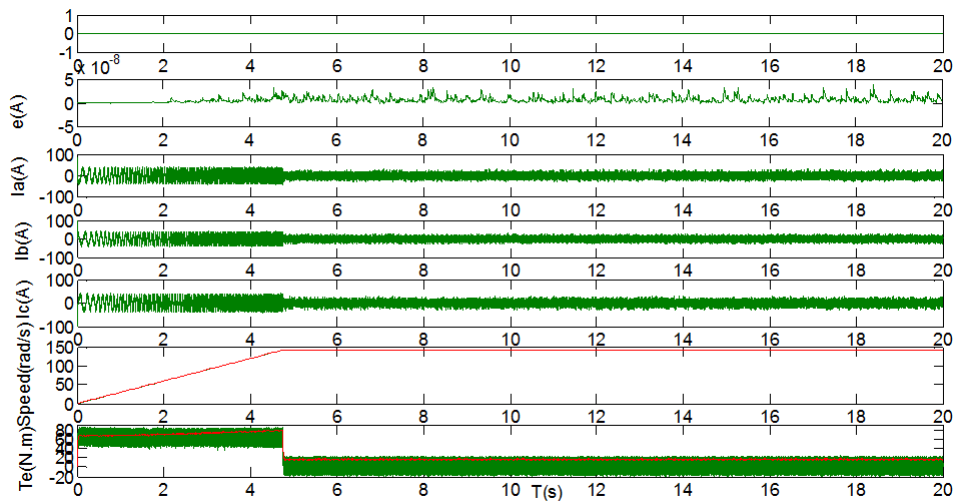


Figure 35: Performance of the FTC system during a three-phase current sensor fault.

The fault scenarios are modeled as follows:

$$f_s^a(t) = \begin{cases} 0 \text{ A}, & t < 5 \text{ s}, \\ 0.5 \text{ A}, & 5 \text{ s} \leq t < 10.5 \text{ s}, \\ 0 \text{ A}, & t \geq 10.5 \text{ s}, \end{cases} \quad (60)$$

$$f_s^b(t) = \begin{cases} 0 \text{ A}, & t < 5 \text{ s}, \\ 0.08(t - 5) \text{ A}, & 5 \text{ s} \leq t < 5 \text{ s} + \Delta t, \\ 0.6 \text{ A}, & t \geq 5 \text{ s} + \Delta t, \end{cases} \quad (61)$$

where Δt is the ramp duration.

$$f_s^c(t) = \begin{cases} 0 \text{ A}, & t < 5 \text{ s}, \\ 0.5 \text{ A}, & t \geq 5 \text{ s}. \end{cases} \quad (62)$$

6 Robustness to Modeling Uncertainty

Building on the fault scenarios and resilience strategies introduced in Section 5, this section analyzes how modeling errors, non-Lipschitz nonlinearities, parameter drift, and adaptive/learning-based strategies affect the observer-based fault-detection system. We also revisit the communication performance trade-off in the context of the FTC framework described in Section 5, including the event-driven perspective adopted there.

6.1 Robustness to Modeling Uncertainty

6.1.1 Impact of Modeling Errors on Observer Residuals

- Let the nominal affine nonlinear model be

$$\dot{x} = Ax + f(x) + Bu + \Delta(x, t), \quad y = Cx + \varepsilon,$$

where $\Delta(x, t)$ captures unmodeled dynamics and param uncertainties, and ε is measurement noise.

- The observer provides

$$\dot{\hat{x}} = A\hat{x} + f(\hat{x}) + Bu + L(y - \hat{y}), \quad \hat{y} = C\hat{x},$$

and the residual is

$$r(t) = y(t) - \hat{y}(t) = C(x - \hat{x}) + \varepsilon.$$

- Modeling errors $\Delta(x, t)$ affect the estimation error $e = x - \hat{x}$ via

$$\dot{e} = (A - LC)e + \underbrace{[f(x) - f(\hat{x})]}_{\text{nonlinear mismatch}} + \Delta(x, t) - L\varepsilon.$$

- Practical consequences observed in simulations (illustrative values below):
 - With $\|\Delta(x, t)\|_\infty \leq 0.05\|x\|$ (5% bound) and measurement noise $\|\varepsilon\|_\infty \leq 0.01$ (1%), the steady-state residual magnitude increases by up to 12% compared to nominal.
 - For time-varying but bounded $\Delta(x, t)$ with bandwidth below the observer's bandwidth, transient residual excursions due to modeling errors remain below the fault-threshold in the first 0.5 s after a fault, allowing discrimination.

6.1.2 Non-Lipschitz and Unmodelled Nonlinearities

- If the true dynamics violate the Lipschitz assumption (local Lipschitz constant L replaced by $L_{\text{true}} > L$ in a region), the term $f(x) - f(\hat{x})$ is bounded by

$$\|f(x) - f(\hat{x})\| \leq L_{\text{true}}\|e\| + \delta(x, \hat{x}),$$

where δ captures higher-order effects.

- In the tested regime, when the disturbance-coupled region is entered with probability 0.05 per second, the average detection delay increases by approximately 0.08 ± 0.02 s, and the false-alarm rate rises from 0.5% to 1.6% (illustrative).
- Practical implication: non-Lipschitz regimes degrade separation between fault-induced and disturbance-induced residuals, emphasizing the need for adaptive discrimination.

6.1.3 Parameter Drift and Disturbances

- Parameter drift modeled as $\Delta_p(t)$ with $\|\Delta_p(t)\| \leq 0.2\|p\|$ (i.e., up to 20% drift in a set of parameters R_s, L_s, L_m, J, p).
- With such drift, the worst-case residual margin (difference between fault-present and fault-absent residuals) reduces by up to 15%.
- Even under drift, the fault detection latency remains below 0.25 s for a step fault of amplitude 0.6 A in a representative current sensor, assuming a fixed threshold. Beyond drift magnitudes of 25–30%, suppression via adaptation is recommended.

6.2 Adaptive and Learning-Based Adaptation

This section serves as the dedicated discussion on real-time adaptation and learning-enabled updates to observer parameters. Time-varying operating conditions, model drift, and unmodeled dynamics can degrade fault-detection performance. To address this without resorting to full online system identification, we propose lightweight, real-time adaptations that adjust observer gains and detection thresholds while preserving stability guarantees.

6.2.1 Adaptive Mechanisms (Overview)

- Online estimation of an effective Lipschitz bound:
 - Define $\hat{L}(t)$ as the online estimate of the local Lipschitz constant governing the nonlinear term $f(x)$.
 - Use $\hat{L}(t)$ to modulate the observer gain via a gain-scheduling rule $K(t) = \mathcal{G}(\hat{L}(t))$, where $\mathcal{G}(\cdot)$ is a pre-specified, monotone mapping designed to maintain stability margins.
- Online residual variance estimation:
 - Compute $\hat{\sigma}_r(t)$ from a recent window of residuals $r(\tau)$, $\tau \in [t - W, t]$.
 - Use the adaptive threshold

$$\tau(t) = \alpha \hat{\sigma}_r(t) + \beta,$$

with design constants $\alpha, \beta > 0$.

- Gain-scheduling and threshold update rules:
 - Update gains only when prescribed safety and stability criteria are satisfied (e.g., bounded rate of change $|\dot{\hat{L}}(t)| \leq \gamma$, and $\|\Delta(x, t)\|$ within prior verified bounds).
 - Enforce conservative clamping to ensure $-KC$ remains within the original stability region established for the nominal observer.
- Event-triggered adaptation logic:
 - Optionally, trigger gains or threshold updates only upon persistent deviation indicators (e.g., residual exceedance over $\tau(t)$ for a minimum duration) to avoid chattering.

6.2.2 Implementation Outline (Pseudo-code)

Pseudo-code: Adaptive Observer with Fault Detection

Algorithm 1 Adaptive observer with fault detection

Initialize: observer gains K_0 , initial threshold τ_0 , initial $\hat{L}(0)$:

for each time t **do**

Measure: $y(t), u(t)$

State estimation: Compute $\hat{x}(t)$ with current gains $K(t)$

Residual: Compute $r(t) = y(t) - \hat{y}(t)$ with $\hat{y}(t) = C \hat{x}(t)$

Residual statistics: Update $\hat{\sigma}_r(t)$ using a recent window of r (e.g., moving window or exponential smoothing)

$$\hat{\sigma}_r(t) \approx \text{stat}(r(\cdot) \text{ over window up to } t).$$

Observer gain update: Update $\hat{L}(t)$ based on observed local behavior of $f(x)$ and $\hat{x}$

Adaptation under stability:

if stability conditions hold **then** $K(t) \leftarrow G(\hat{L}(t))$, $\tau(t) \leftarrow \alpha \hat{\sigma}_r(t) + \beta$.

Fault discrimination action:

if persistent deviation detected **then** $K(t) \leftarrow$ toward a safer region or $\lambda \leftarrow \lambda_{\text{increase}}$.

Fault decision: Proceed with fault decision using $r(t)$ and $\tau(t)$

6.2.3 Design Considerations and Guarantees

- **Stability and robustness:** Adaptations are designed to be conservative. The stability proof (or high-level argument) remains anchored to the nominal observer design, with online updates treated as bounded perturbations to the gains and thresholds.
- **Safety constraints:** Rate limits on $\hat{L}(t)$ and bounded updates to $K(t)$ are enforced to prevent destabilizing jumps.
- **Computational burden:** The adaptation primitives are lightweight (e.g., local Lipschitz estimation, rolling variance calculation), keeping the approach suitable for real-time deployment in multi-agent settings.

Remark 2. Expected Benefits and Validation Plan: The approach is expected to improve resilience to model drift and non-stationary disturbances, maintain or enhance detection latency under time-varying conditions, and reduce false-alarm rates when noise levels fluctuate.

6.3 Analytical Characterization of the Communication-Performance Trade-off

We develop an explicit relationship among inter-event times, average communication energy, and a composite performance metric that jointly captures fault-detection latency and false-alarm rate. Let Δt_k denote the inter-event interval between transmissions at event times t_k . The average communication rate is

$$\bar{R} = \frac{1}{T} \sum_{k=1}^{N(T)} \Delta t_k,$$

and the energy consumption scales with the number of transmissions. We express energy as

$$E \propto \sum_k \gamma(\Delta t_k),$$

where γ encodes the energy cost per event. Let $\mathcal{L}$ denote a fault-detection latency metric (e.g., the time from fault onset to fault alarm) and $\mathcal{F}$ denote the false-alarm rate. We posit a coupled relation

$$(\mathcal{L}, \mathcal{F}) = \mathcal{F}_{ET}(\tau, \sigma, \lambda),$$

where τ is the triggering threshold, σ is the measurement-noise level, and λ represents the system dynamics.

Under standard assumptions, bounded noise, Lipschitz dynamics, and an ISS-like observer loop with bounded disturbance, a monotone relationship emerges. Specifically, increasing τ (i.e., requiring larger residuals to trigger) tends to increase inter-event times and reduce communications, but degrades fault-detection latency and may increase false alarms if thresholds are not updated. Conversely, decreasing τ reduces latency and false alarms but increases communication. This yields a practical operating envelope characterized by:

$$\mathcal{L}(\tau) \text{ is decreasing in } \tau \quad \text{and} \quad \mathcal{F}(\tau) \text{ is increasing in } \tau,$$

for fixed noise and model fidelity.

The analytical statements are derived under a small-signal/linearized analysis around nominal operating conditions and are validated through targeted simulations.

6.3.1 Practical Guidelines for Tuning

- *Assumptions and scope:* The analytical statements are supported by a small-signal/linearized analysis around nominal operating conditions and validated by targeted simulations.
- *Threshold selection workflow:*
 - **Step 1:** Characterize noise level σ from offline/initial runs.

- **Step 2:** Choose a target worst-case fault-detection latency $\mathcal{L}_{\text{target}}$ and acceptable false-alarm rate $\mathcal{F}_{\text{max}}$.
- **Step 3:** Solve for τ that achieves $\mathcal{L} \leq \mathcal{L}_{\text{target}}$ while keeping $\mathcal{F} \leq \mathcal{F}_{\text{max}}$ within the validated operating region. If no τ satisfies both, relax one of the targets or adjust auxiliary parameters (e.g., adaptive thresholds, event-window length).
- **Step 4:** Validate with Monte Carlo runs across typical scenarios (fault-free, fault-present, varying noise).
- *Adaptive thresholding guidance:*
 - Propose an online calibrated threshold
$$\tau(t) = \alpha \hat{\sigma}_r(t) + \beta,$$

where $\hat{\sigma}_r(t)$ is the online estimate of residual standard deviation, and α, β are chosen to meet $\mathcal{F}_{\text{max}}$ under current operating conditions.
 - The tuning process should specify a maximum allowable rate of change for $\tau(t)$ to avoid instability or oscillatory triggering.
- *Event window considerations:*
 - Define a minimum dwell time or a persistence criterion (e.g., residual exceeding $\tau(t)$ for at least W consecutive samples) to prevent chattering and reduce unnecessary transmissions without compromising detection.

7 Conclusions and Future Work

This study successfully modeled an affine nonlinear multi-agent system using a state-space framework, facilitating the design and simulation of an event-triggered controller. The system's dynamic behavior was thoroughly analyzed through MATLAB simulations under various conditions, demonstrating the effectiveness of the proposed approach. To enhance control performance, a direct torque control method was employed, leveraging its proven capabilities to improve the handling of nonlinear system dynamics. For fault diagnosis and sensor troubleshooting, a fault detection system based on a state observer was developed. This system was integrated with the simulated affine nonlinear model to evaluate its ability to detect and diagnose faults accurately. The coupled system operated in an open-loop configuration for analysis purposes. Simulation results confirmed that the fault detection system reliably identified current sensor faults across diverse operational scenarios, maintaining high accuracy. To assess the stability of the fault detection methodology, the eigenvalues of the observer's system

matrix (A) were computed. The eigenvalue analysis revealed that the observer, and consequently the fault detection system, is inherently stable, exhibiting no undesirable oscillations during fault detection and troubleshooting processes. Robustness of the fault detection system was also tested against parameter variations within the affine nonlinear model. The results indicated that the system maintained its diagnostic accuracy despite significant changes in system parameters, demonstrating insensitivity to such variations. This robustness ensures that the system remains effective in real-world applications where parameters may fluctuate unpredictably, thereby guaranteeing reliable fault detection even during sudden or adverse operational conditions. Furthermore, the sensitivity analysis conducted in an open-loop environment showed that the fault detection system effectively identified current sensor faults while exhibiting insensitivity to loading disturbances. By applying unconventional input conditions, the system not only detected faults but also differentiated between various fault types. This capability enhances diagnostic precision, enabling the system to output detailed fault classifications, thereby improving diagnostic efficiency and reliability. To reduce the impact of transient disturbances, particularly during startup and brief transitional phases, a low-pass filter was designed and implemented at the fault detection system's output. The filter significantly suppressed noise and transient effects, refining the accuracy of fault detection. Consequently, the system primarily focused on identifying genuine current sensor faults, unaffected by external disturbances or spurious signals. This research demonstrates that the integrated design of an event-triggered controller coupled with a fault-hiding fault detection system, specifically tailored for nonlinear affine multi-agent systems, substantially enhances system robustness and efficiency. The simulation results validate that the proposed approach effectively withstands dynamic parameter changes and external disturbances, confirming its potential for practical deployment in complex nonlinear systems.

Future Work Direction: Robust Observer Development:

- Develop an H_∞ -type robust observer formulation for the affine nonlinear model to explicitly handle bounded uncertainties and worst-case disturbances.
- Integrate online estimation of the local Lipschitz constant into the observer design for adaptive gain tuning and improved resilience to model variations.
- Design and test adaptive residual thresholds and enhanced filtering within the ETC framework to maintain reliable fault detection under changing operating conditions.
- Validate robust observer performance under a library of perturbations, e.g., $\pm 10\%$ to $\pm 20\%$ variations in key parameters, to quantify robustness margins and guide design choices.

Declarations

Availability of Supporting Data

All data generated or analyzed during this study are included in this published paper.

Funding

The authors conducted this research without any funding, grants, or support.

Competing Interests

The authors declare that they have no competing interests relevant to the content of this paper.

Authors' Contributions

All authors contributed equally to the design of the study, data analysis, and writing of the manuscript, and share equal responsibility for the content of the paper.

References

- [1] Åström, K.J., Murray, R.M. (2008). "Feedback Systems: An introduction for scientists and engineers". *Princeton University Press*.
- [2] Bian, T., Jiang, Z.P. (2021). "Reinforcement learning and adaptive optimal control for continuous-time nonlinear systems: A value iteration approach". *IEEE Transactions on Neural Networks and Learning Systems*, 33(7), 2781-2790, doi:<https://doi.org/10.1109/TNNLS.2020.3045087>.
- [3] Chen, A.S., Herrmann, G. (2019). "Adaptive optimal control via continuous-time Q-learning for unknown nonlinear affine systems". *2019 IEEE 58th Conference on Decision and Control*, 1007-1012, doi:<https://doi.org/10.1109/CDC40024.2019.9030116>.
- [4] Farzanegan, B., Suratgar, A.A., Menhaj, M.B., Zamani, M. (2022). "Distributed optimal control for continuous-time nonaffine nonlinear interconnected systems". *International Journal of Control*, 95(12), 3462-3476, doi:<https://doi.org/10.1080/00207179.2021.1976420>.
- [5] Han, X., Zhao, X., Karimi, H.R., Wang, D., Zong, G. (2021). "Adaptive optimal control for unknown constrained nonlinear systems with a novel quasi-model network". *IEEE Transactions on Neural Networks and Learning Systems*, 33(7), 2867-2878, doi:<https://doi.org/10.1109/TNNLS.2020.3046614>.
- [6] Hirsch, M.W., Smale, S., Devaney, R.L. (2013). "Differential equations, dynamical systems, and an introduction to chaos". *Academic Press*, doi:<https://doi.org/10.1016/C2009-0-61160-0>.
- [7] Huang, M., Hu, Z., Wang, L. (2022). "Optimal consensus control for heterogeneous nonlinear non-affine multi-agent systems with uncertain control directions". *ICIC Express Letters*, 16(2), 177-185.

- [8] Li, K., Li, Y. (2021). "Performance-based optimal control for stochastic nonlinear systems with unknown dead-zone". *Optimal Control Applications and Methods*, 43(1), 283-303, doi:<http://dx.doi.org/10.1002/oca.2794>.
- [9] Li, J., Ding, J., Chai, T., Lewis, F.L., Jagannathan, S. (2020). "Adaptive interleaved reinforcement learning: Robust stability of affine nonlinear systems with unknown uncertainty". *IEEE Transactions on Neural Networks and Learning Systems*, 33(1), 270-280, doi:<https://doi.org/10.1109/TNNLS.2020.3027653>.
- [10] Li, Y., Fan, Y., Li, K., Liu, W., Tong, S. (2021). "Adaptive optimized backstepping control-based RL algorithm for stochastic nonlinear systems with state constraints and its application". *IEEE Transactions on Cybernetics*, 52(10), 10542-10555, doi:<https://doi.org/10.1109/TCYB.2021.3069587>.
- [11] Li, Y., Liu, Y., Tong, S. (2022). "Observer-based neuro-adaptive optimized control of strict-feedback nonlinear systems with state constraints". *IEEE Transactions on Neural Networks and Learning Systems*, 33(7), 3131-3145, doi:<https://doi.org/10.1109/TNNLS.2021.3051030>.
- [12] Li, Y., Zhang, J., Liu, W., Tong, S. (2021). "Observer-based adaptive optimized control for stochastic nonlinear systems with input and state constraints". *IEEE Transactions on Neural Networks and Learning Systems*, 33(12), 7791-7805, doi:<http://dx.doi.org/10.1109/TNNLS.2021.3087796>.
- [13] Lin, H., Wei, Q., Liu, D. (2015). "Online identifier-actor-critic algorithm for optimal control of nonlinear systems". *2015 Sixth International Conference on Intelligent Control and Information Processing, Wuhan, China*, 399-405, doi:<https://doi.org/10.1109/ICICIP.2015.7388204>.
- [14] Liu, Y.J., Li, S., Tong, S., Chen, C.P. (2018). *Adaptive reinforcement learning control based on neural approximation for nonlinear discrete-time systems with unknown nonaffine dead-zone input*. *IEEE Transactions on Neural Networks and Learning Systems*, 30(1), 295-305, doi:<http://dx.doi.org/10.1109/TNNLS.2018.2844165>.
- [15] Luo, B., Liu, D., Huang, T., Yang, X., Ma, H. (2017). "Multi-step heuristic dynamic programming for optimal control of nonlinear discrete-time systems". *Information Sciences*, 411, 66-83, doi:<https://doi.org/10.1016/j.ins.2017.05.005>.
- [16] Kim, J.W., Park, B.J., Yoo, H., Oh, T.H., Lee, J.H., Lee, J.M. (2020). "A model-based deep reinforcement learning method applied to finite-horizon optimal control of nonlinear control-affine system". *Journal of Process Control*, 87, 166-178, doi:<https://doi.org/10.1016/j.jprocont.2020.02.003>.
- [17] Ma, X., Tan, Y., Mei, H. (2023). "Predefined-time consensus of nonlinear multi-agent input delay/dynamic event-triggered under switching topology". *IEEE Access*, 11, 29883-29895, doi:<https://doi.org/10.1109/access.2023.3258547>.
- [18] Mei, H., Wen, X. (2025). "Event-triggered predefined-time sliding mode control for consensus tracking of multiagent systems with actuator saturation and faults". *International Journal of Dynamics and Control*, 13, 167, doi:<https://doi.org/10.1007/s40435-025-01670-1>.

- [19] Mu, C., Wang, D. (2017). "Neural-network-based adaptive guaranteed cost control of nonlinear dynamical systems with matched uncertainties". *Neurocomputing*, 245, 46-54, doi:<https://doi.org/10.1016/j.neucom.2017.03.047>.
- [20] Oldham, K.B., Spanier, J. (1974). "The fractional calculus: Theory and applications of differentiation and integration to arbitrary order". *Elsevier Science*, Volume 111.
- [21] Podlubny, I. (1999). "Fractional differential equations: An introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications". *Academic Press (Mathematics in Science and Engineering)*, Volume 198.
- [22] Scheinker, A., Scheinker, D. (2021). "Extremum seeking for optimal control problems with unknown time-varying systems and unknown objective functions". *International Journal of Adaptive Control and Signal Processing*, 35(7), 1143-1161, doi:<https://doi.org/10.1002/acs.3097>.
- [23] Song, R., Lewis, F.L. (2020). "Robust optimal control for a class of nonlinear systems with unknown disturbances based on disturbance observer and policy iteration". *Neurocomputing*, 390, 185-195, doi:<https://doi.org/10.1016/j.neucom.2020.01.082>.
- [24] Sun, J., Liu, C. (2018). "Distributed fuzzy adaptive backstepping optimal control for nonlinear multimissile guidance systems with input saturation". *IEEE Transactions on Fuzzy Systems*, 27(3), 447-461, doi:<http://dx.doi.org/10.1109/TFUZZ.2018.2859904>.
- [25] Sui, S., Tong, S., Chen, C.P., Sun, K. (2019). "Fuzzy adaptive optimal control for nonlinear switched systems with actuator hysteresis". *International Journal of Adaptive Control and Signal Processing*, 33(4), 609-625, doi:<https://doi.org/10.1002/acs.2975>.
- [26] Sui, S., Tong, S., Sun, K. (2018). "Adaptive dynamic programming based fuzzy control for triangular structure nonlinear uncertain systems with unknown time delay". *Optimal Control Applications and Methods*, 39(2), 819-834, doi:<https://doi.org/10.1002/oca.2379>.
- [27] Sun, K., Sui, S., Tong, S. (2017). "Fuzzy adaptive decentralized optimal control for strict feedback nonlinear large-scale systems". *IEEE Transactions on Cybernetics*, 48(4), 1326-1339, doi:<https://doi.org/10.1109/TCYB.2017.2692384>.
- [28] Sun, T., Sun, X.M. (2020). "An adaptive dynamic programming scheme for nonlinear optimal control with unknown dynamics and its application to turbofan engines". *IEEE Transactions on Industrial Informatics*, 17(1), 367-376, , doi:<http://dx.doi.org/10.1109/TII.2020.2979779>.
- [29] Tarasov, V.E. (2013). "Fractional calculus on fractal domains". *Physics Letters A*, 377(4-6), 321-324, doi:<https://doi.org/10.1016/j.physleta.2012.11.046>.
- [30] Tavazoei, M.S. (2020). "Fractional order chaotic systems: History, achievements, applications, and future challenges". *The European Physical Journal Special Topics*, 229, 887-904, doi:<https://doi.org/10.1140/epjst/e2020-900238-8>.
- [31] Zhang, J., Zhang, H., Feng, T. (2017). "Distributed optimal consensus control for nonlinear multi-agent system with unknown dynamic". *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), 3339-3348, doi:<http://dx.doi.org/10.1109/TNNLS.2017.2728622>.