N°xx/2026

People's Democratic Republic of Algeria
Ministry Of Higher Education and Scientific Research
Echahid Cheikh Larbi Tebessi University - Tebessa
Faculty OF Science and Technology

Thesis

Presented for obtaining the diploma of

3rd Cycle LMD DOCTORATE

In Field of : **Eletrotechnics**

Speciality: **Eletrotechnics**

by : **Nizar BENAYAD**

Title

Diagnosis and Fault-Tolerant Control of Nonlinear Systems

Discussed publicly at: 18/06/2026 in front of the assessment committee:

Youcef SOUFI	*Professor*	*Echahid Cheikh Larbi Tebessi University*	Chairman
Abdelaziz AOUICHE	*Professor*	*Echahid Cheikh Larbi Tebessi University*	Supervisor
Abdelghani DJEDDI	*Professor*	*Echahid Cheikh Larbi Tebessi University*	Co-supervisor
Tahar BAHI	*Professor*	*Badji Mokhtar - Annaba University*	Examiner
Abla BOUGUERNE	*Lecturer A*	*Echahid Cheikh Larbi Tebessi University*	Examiner
Laatra YOUSFI	*Lecturer A*	*Echahid Cheikh Larbi Tebessi University*	Examiner

"Success consists of going from failure to failure without loss of enthusiasm."
Winston Churchill.

Acknowledgement

First and foremost, I thank *Allah the Almighty* for granting me the strength, patience, and will to complete this work and overcome the challenges faced during this research journey.

I would like to express my deepest gratitude to my supervisor, *Pr. Abdelaziz AOUICHE*, for his invaluable guidance, scientific insight, and continuous encouragement. His rigorous standards and deep understanding of control systems have been instrumental in shaping this thesis.

My sincere thanks also go to my co-supervisor, *Pr. Abdelghani DJEDDI*. I am grateful for his availability, constructive criticism, and the technical support provided throughout the development of the fault-tolerant control strategies.

This work is dedicated to the people who mean the most to me. To my beloved *parents*, no words can express my gratitude. To my *father*, for your endless sacrifices and for teaching me the value of hard work. To my *mother*, for your unconditional love and your prayers that have been my fortress. You are the reason I am here today.

To my dear *brother and sisters*, thank you for being my source of joy and for your constant encouragement.

A very special acknowledgement goes to my teachers *Dr. Mohamed Salah DJEBBAR* and *Dr. Yousfi LAATRA*, and my best friend and brother, **Tayeb BOUTARFA**. Thank you for your unwavering friendship, for listening to my endless discussions and my philosophical theorems about life, and for always being there to lift my spirits. This journey would have been much harder without your support.

Finally, I thank everyone who contributed, directly or indirectly, to the realization of this thesis.

Abstract

The demand for reliable, autonomous operation in complex, safety-critical non-linear systems such as Unmanned Aerial Vehicles (UAVs) necessitates advanced Fault-Tolerant Control (FTC) capabilities. Conventional FTC methods struggle with system non-linearity and lack the adaptive performance required for robust operation. Conversely, purely AI-based approaches, while highly adaptive, suffer from a lack of formal safety guarantees and increased computational complexity. This dissertation bridges this critical gap by proposing and validating a novel Hybrid Fault-Tolerant Control (HFTC) strategy that strategically integrates the verifiable rigor of conventional control with the optimizing power of metaheuristic techniques. The core of this framework is a robust Sliding Mode Control (SMC) architecture enhanced by a proposed Enhanced Triple Power Reaching Law (ETPRL), which provides superior chattering attenuation and accelerated finite-time convergence. The HFTC architecture employs a unique hybrid compensation strategy that includes an Online Diagnosis component, where a conventional extended kalman filter (EKF-based) fault detection and identification (FDI) unit provides real-time fault estimates, and an Offline Optimization component, where the particle swarm optimization (PSO) algorithm is utilized to pre-compute and store a "Bank of Parameters" for numerous actuator fault scenarios. The efficacy of this framework is rigorously validated through a comprehensive case study on a UAV quadrotor subjected to severe Loss of Effectiveness (LOE) actuator faults. MATLAB/Simulink simulations, including challenging helical and square trajectory maneuvers, quantitatively demonstrate the superior performance of the HFTC-ETPRL system, confirming its ability to maintain stability and achieve minimal tracking error (e.g., up to a 50% reduction in Peak Deviation compared to benchmark controllers), underscoring the benefits of non-iterative, pre-optimized reconfiguration. In synthesis, this research advances the state-of-the-art by validating an efficient and high-performance hybrid solution that addresses the fundamental trade-off between robustness and adaptability.

Keywords: Fault-Tolerant Control (FTC), Sliding Mode Control (SMC), Enhanced Triple Power Reaching Law (ETPRL), Particle Swarm Optimization (PSO), Extended Kalman Filter (EKF), Unmanned Aerial Vehicle (UAV), Loss of Effectiveness (LOE).

Contents

List of Figures

List of Tables

Nomenclature

Acronyms / Abbreviations

AFTC	Active Fault-Tolerant Control
AI	Artificial Intelligence
ANNs	Artificial Neural Networks
AREs	algebraic Riccati equations
ARR	analytical redundancy relations
BLDC	Brushless DC Motor
CNNs	Convolutional Neural Networks
DFT	Discrete Fourier Transform
DL	Deep Learning
EKF	Extended Kalman Filter
ES	Expert System
ESC	Electronic Speed Controller
ETPRL	Enhanced Triple Power Reaching Law
FDA	Fisher Discriminant Analysis
FDD	fault detection and diagnosis

FDI	fault detection and isolation
FDP	Fault Diagnosis and Prognosis
FFT	Fast Fourier Transform
FLBD	Fuzzy Logic Based Diagnosis
FMU	Flight Management Unit
FPM	First-Principle Modelling
FTC	Fault-Tolerant Control
GPS	Global Positioning System
GRU	Gated Recurrent Unit
HFTC	Hybrid Fault-Tolerant Control
HGO	High-Gain Observers
HJI	Hamilton-Jacobi-Isaacs
HOSM	Higher-Order Sliding Mode Control
IMU	Inertial Measurement Unit
IV	Instrumental Variable
KBS	Knowledge-Based System
KCL	Kirchhoff's Current law
KVL	Kirchhoff's Voltage law
LDA	Linear Discriminant Analysis
LOE	loss of effectiveness
LS	Least Squares
LSTM	Long Short-Term Memory

LTI	linear time-invariant
ML	Machine Learning
MRAC	Model Reference Adaptive Control
NARMAX	Nonlinear Autoregressive Moving Average with exogenous inputs
NARX	nonlinear autoregressive with exogenous inputs
PCA	Principal Component Analysis
PFTC	Passive Fault-Tolerant Control
PMSM	Permanent Magnet Synchronous Motor
PSD	Power Spectral Density
PSO	Particle Swarm Optimization
RLS	Recursive Least Squares
RNNs	Recurrent Neural Networks
SISO	single-input single-output
SMC	Sliding Mode Control
SMO	Sliding Mode Observer
SPE	squared prediction error
STFT	Short-Time Fourier Transform
SVMs	Support Vector Machines
UAVs	Unmanned Aerial Vehicles
UKF	Unscented Kalman Filter
VTOL	Vertical Take-Off and Landing

General Introduction

The modern technological landscape is dominated by complex, high-performance physical and engineering systems, the vast majority of which exhibit inherently non-linear behaviour. Unlike their linear counterparts, whose dynamics are governed by the principle of superposition, nonlinear systems are described by nonlinear differential or difference equations that frequently give rise to complex and unpredictable phenomena such as multiple equilibria, limit cycles, and chaotic dynamics [1] [2]. The accurate mathematical representation and robust control of these systems—which include aerospace vehicles like Unmanned Aerial Vehicles (UAVs), large-scale power grids, and advanced mechatronic devices—presents a fundamental challenge to engineers and researchers. Consequently, the development of robust, reliable, and mathematically sound frameworks for managing system integrity under these nonlinear conditions is a critical necessity [3].

In safety-critical sectors, the failure of a single component can have rapid, cascading, and often catastrophic effects. Therefore, designing systems capable of maintaining stability and acceptable performance in the presence of faults—a concept known as Fault Management—is no longer a desirable design feature but a non-negotiable requirement. This management process is bifurcated into two essential stages [4]:

1. ***Fault Diagnosis and Prognosis (FDP)***: The process of detecting (determining a fault has occurred), isolating (locating the faulty component), and identifying (quantifying the severity of the fault). FDP forms the indispensable prerequisite for any corrective action [5].

2. ***Fault-Tolerant Control (FTC)***: The ability of the control system to actively or passively compensate for the effects of an identified fault, thereby maintaining the desired operational objectives. FTC methodologies are broadly classified as *Passive FTC (PFTC)*, which rely on a single, fixed robust controller (e.g., $H\infty$) to suppress expected faults; *Active FTC (AFTC)*, which explicitly use diagnostic information to reconfigure the control loop in real-time; and *Hybrid FTC (HFTC)*, which strategically blend the verifiable robustness of PFTC with the adaptability of AFTC, representing a modern approach that is the focus of this research [6].

The core objective of this research is to advance HFTC strategies by developing a highly effective solution that addresses the profound challenges presented by nonlinear system dynamics and actuator faults [7].

The evolution of system integrity management has been shaped by a succession of control and diagnostic paradigms, each serving as the historical bedrock for industrial automation. However, their efficacy falters when confronted with the high-speed dynamics, severe non-linearities, and unpredictable fault scenarios of modern UAVs. These foundational methods each possess inherent limitations that necessitate a more advanced, hybrid approach [8]:

- **Classical Linear Foundations (PID, PI):** Proportional-Integral-Derivative (PID) and PI controllers are the most widely adopted industrial solutions due to their simplicity and ease of tuning. However, they are fundamentally ill-equipped for nonlinear quadrotor dynamics; they lack an inherent mechanism to adapt to structural changes, and a single actuator fault often pushes the system beyond the narrow linear range where these controllers remain stable [9].

- **Optimal and State-Space Methods (LQR, LQS):** Techniques such as the Linear Quadratic Regulator (LQR) and Linear Quadratic State-feedback (LQS) offer optimal performance under ideal, nominal conditions. Yet, they are notoriously *brittle*; because they rely heavily on a precise linear model, any parameter drift or actuator Loss of Effectiveness (LOE) significantly degrades their optimality, often leading to rapid system divergence [10].

- **Robust Control (H_∞):** Developed to handle model uncertainties and bounded disturbances, H_∞ control provides a high degree of stability. However, it is frequently critiqued for being *overly conservative*. By designing for the absolute worst-case scenario, it often sacrifices the tracking performance and agility required for quadrotors to perform critical recovery maneuvers [11].

- **Nonlinear Backstepping Control:** This technique offers an elegant recursive way to handle system nonlinearities. However, it is plagued by the *"explosion of complexity"* phenomenon; as the system order increases, the repeated differentiation of virtual control laws leads to mathematically massive expressions that are computationally expensive to implement on limited embedded flight hardware [12].

While conventional FDP and FTC techniques have served as the historical bedrock for industrial automation, their efficacy falters when confronted with the growing complexity, nonlinearity, and operational variability of modern systems. These limitations are primarily rooted in their two main paradigms [13]:

- ***Model-Based Limitations:*** These methods (e.g., Luenberger Observers, classical Parity Space) inherently rely on the prerequisite of an accurate mathematical model. Developing a high-fidelity model that captures all non-linearities, time-varying parameters, and complex coupling effects (such as those in a UAV) is exceptionally difficult. Even advanced techniques like the Extended Kalman Filter (EKF) suffer from linearization errors in highly nonlinear environments, and observers like the Sliding Mode Observer (SMO) are often plagued by the chattering phenomenon. This fundamental model-dependency renders them brittle and computationally intractable in the face of significant unmodeled dynamics and novel fault scenarios [13, 14].

- ***Signal-Based Limitations:*** These methods (e.g., PCA, Spectral Analysis) bypass the need for a physical model but introduce dependence on extensive empirical data. Their key limitations include the need for large, representative historical datasets (which is often impossible for rare fault conditions), a heavy reliance on laborious manual feature engineering, and a resulting lack of physical insight (black-box tendency), which hinders accurate root-cause analysis [15, 16].

The inherent limitations of conventional methods have motivated a paradigm shift toward data-driven, learning-based techniques. Artificial Intelligence (AI) methodologies offer a powerful alternative capable of addressing system non-linearity and overcoming model-dependency [17]:

- ***Universal Approximation:*** Techniques like Artificial Neural Networks (ANNs) and Fuzzy Logic Controllers possess the ability to approximate any arbitrary nonlinear function from input-output data. This makes them highly effective for identifying and controlling complex dynamics without requiring an explicit analytical model [18].
- ***Optimal Synthesis:*** AI algorithms, particularly metaheuristic optimization methods like Particle Swarm Optimization (PSO), are indispensable tools for solving high-dimensional tuning problems, ensuring that complex multi-variable controllers (like: Sliding Mode Controllers operate at their absolute optimum across a range of conditions [19].
- ***Adaptive Learning:*** Certain AI techniques, including Neural Network-based adaptive control and Reinforcement Learning, enable the control system to learn new policies or parameters online, offering a degree of adaptive flexibility that passive conventional methods cannot match [20].

Research into Fault-Tolerant Control for Unmanned Aerial Vehicles is a mature field, yet it continues to evolve toward more adaptive and intelligent solutions.

Early works, such as the theses by *Avram and Ziquane*, focused on establishing core FTC techniques for quadrotors. *Avram's* work investigated both integrated fault diagnosis and accommodation schemes, including a nonlinear adaptive fault-tolerant controller without an explicit diagnosis mechanism [21]. *Ziquane* similarly explored model reference adaptive control (MRAC) and enhanced multiple model adaptive estimation (EMMAE) using Extended Kalman Filters (EKF) for fault diagnosis. These works established the critical role of both adaptive control and model-based diagnosis (EKF) in FTC [22].

More recent research, exemplified by the thesis from *Khattab*, has concentrated on leveraging Sliding Mode Control (SMC) schemes for multi-rotor UAVs, emphasizing the hardware implementation and the combination of SMC with control allocation. This body of work highlights the established utility of SMC for its robustness but implicitly accepts its limitations (chattering, tuning complexity) [23].

The shift toward AI-based paradigms is represented by *Aoun's* thesis, which successfully demonstrates a Fault Tolerant Deep Reinforcement Learning architecture combined with an Observer-based Kalman Filter. This work showcases the superior adaptivity and optimality potential of AI while confirming the ongoing need for conventional observer-based diagnosis to estimate adverse inputs [24]. Another approach, seen in the work by *He*, uses a Multiple Model Approach to design adaptive FTC strategies, an established concept that bridges model-based and adaptive control by switching between a bank of simplified local models [25].

This thesis builds directly upon the established foundation of SMC's robustness and EKF's model-based diagnosis, but addresses the critical need for a non-iterative, AI-optimized approach to overcome the limitations of manual tuning and chattering inherent in previous SMC-based FTC schemes.

The central challenge in developing next-generation FTC for complex nonlinear systems is rooted in the *fundamental scientific trade-off* between two desirable system attributes: *Verifiable Robustness* and *Adaptive Performance* [26].

Conventional, model-based methods (e.g., the EKF-SMC framework) offer mathematically *verifiable robustness* and direct physical insight, but their performance is severely limited by model inaccuracy and their inherent *inability* to adapt to unforeseen faults. Conversely, purely AI-based systems offer superior *adaptive performance* and model-free flexibility, but their "black-box" nature leads to a critical lack of formal safety guarantees and high computational overhead, making them often unsuitable for safety-critical, time-sensitive applications like UAVs [27].

The problem addressed by this thesis is the need to bridge this gap. There is a compelling need for a unified, Hybrid Fault-Tolerant Control framework for nonlinear systems that successfully integrates the verifiable rigor of conventional control theory with the adaptive, optimizing power of AI techniques. Specifically, the framework must demonstrate the capacity to provide superior performance and robustness to actuator faults while simultaneously maintaining the speed and reliability required for real-time deployment. In response to the identified problem, the primary goal of this research is to develop, implement, and rigorously validate a novel hybrid fault-tolerant control framework for nonlinear systems. The specific objectives are as follows:

1. ***Systematic Literature Review:*** To conduct a comprehensive classification and critical analysis of conventional and AI-based methodologies for Fault Diagnosis, Fault-Tolerant Control, and System Modelling in nonlinear systems.

2. ***Novel Control Law Development:*** To develop a robust, high-performance Sliding Mode Control (SMC) law—the Enhanced Triple Power Reaching Law (ETPRL)—specifically designed to maintain the benefits of SMC while significantly accelerating convergence and attenuating the problematic chattering phenomenon.

3. ***Hybrid Architecture Synthesis:*** To synthesize a unique HFTC architecture that combines a conventional model-based diagnosis unit EKF-FDI with a metaheuristic-optimized parameter bank PSO to ensure fast and precise control reconfiguration.

4. ***Empirical Validation:*** To apply the developed HFTC-ETPRL framework to a challenging, real-world case study—the UAV quadrotor—and perform a rigorous comparative analysis against conventional SMC controllers under severe, cumulative actuator faults.

The successful execution of the research objectives yields several key contributions to the field of control engineering and fault-tolerant systems:

1. ***A Unified Theoretical Framework:*** The thesis provides a structured, comparative analysis of conventional and AI-based techniques across the three pillars of system integrity (Diagnosis, FTC, and Modelling), establishing the necessary justification for the adoption of hybrid solutions.

2. ***Novel ETPRL Control Law:*** The thesis proposes and validates the Enhanced Triple Power Reaching LawEnhanced Triple Power Reaching Law, a novel SMC discontinuous term that demonstrably achieves faster finite-time convergence and superior chattering suppression than existing power, double power, and constant reaching laws.

3. ***Optimal Hybrid FTC Architecture:*** Architecture: A novel HFTC architecture is presented and validated. This system's innovation lies in its hybrid, two-stage compensation strategy: *(i) Offline Optimization:* PSO is utilized to meticulously tune a "Bank of Parameters" to compensate for various fault scenarios, and *(ii) Online Reconfiguration:* The EKF provides the real-time fault identification to index this bank, resulting in a swift, non-iterative, and highly effective control reconfiguration.

4. ***Rigorous Empirical Validation:*** The thesis offers extensive simulation results demonstrating the superior performance of the HFTC-ETPRL on the nonlinear quadrotor model against severe Loss of Effectiveness LOE actuator faults in both dynamic helical and demanding square trajectory scenarios. The quantitative data provides concrete evidence of enhanced robustness, tracking accuracy, and control efficiency.

The remainder of this dissertation is organized into four main chapters, following the logical progression from fundamental theory to applied validation:

- *Chapter 01:* Diagnosis of Nonlinear Systems: Conventional and AI-Based Approaches, (FDP). It systematically covers Model-Based (EKF, SMO) and Signal-Based (PCA, FDA) methods, contrasts them with AI-based solutions (NNs, Fuzzy Logic), and establishes the critical limitations that necessitate the advanced diagnostic components used in this thesis.

- *Chapter 02:* Fault-Tolerant Control of Nonlinear Systems: Conventional and AI-Based Approaches, This chapter establishes the theoretical foundations of Fault-Tolerant Control (FTC). It defines the classification of PFTC, AFTC, and Hybrid approaches, detailing the architecture of prominent conventional techniques ($H\infty$, SMC) and examining the potential, challenges, and architecture of AI-based FTC solutions.

- *Chapter 03:* Modelling of Nonlinear Systems: Conventional and AI-based Methods, This chapter addresses the foundational problem of system representation. It reviews methodologies for Nonlinear System Modelling Nonlinear System Modelling, distinguishing between First-Principle (White-Box), System Identification (Grey-Box), and AI-Based approaches (NNs, Fuzzy), thereby providing the essential model-based context for the subsequent control design.

- *Chapter 04:* Case Study: Fault-Tolerant Control of UAV Quadrotors, This chapter serves as the capstone of the research. It applies the synthesized HFTC-ETPRL framework to the UAV quadrotor. It details the FTC system design, the PSO optimization procedure, and presents the rigorous MATLAB/Simulink simulation and comparative analysis that validates the superior performance of the proposed strategy under critical fault conditions.

Chapter 1

Diagnosis of Nonlinear Systems: Conventional and AI-Based Approaches

1.1 Introduction

The increasing complexity and non-linearity of modern safety-critical systems, such as UAVs, necessitate sophisticated Fault Diagnosis and Prognosis (FDP) methods [2]. Undetected faults in these systems can rapidly escalate into catastrophic failures, compromising both safety and operational efficiency. Consequently, developing robust diagnostic techniques capable of handling nonlinear dynamics is a critical research priority [3, 28].

This chapter lays the foundational groundwork for the thesis by conducting a comprehensive review of the state-of-the-art diagnostic methodologies applicable to nonlinear systems. To provide a structured understanding of this domain, the discussion proceeds from fundamental definitions to a critical evaluation of methodological paradigms.

The investigation begins by establishing the necessary theoretical framework, clarifying the distinctions between degradation, faults, and failures, and classifying the various fault types based on their time evolution and location. This foundational analysis highlights the specific challenges inherent in diagnosing nonlinear systems, setting the stage for a detailed examination of solution strategies.

The review then systematically explores the historical evolution of diagnostic techniques. It first scrutinizes Conventional Methods, encompassing both Model-Based approaches, such as State Observers and Parameter Estimation, and Signal-Based techniques like Statistical Analysis. While these methods offer mathematical rigor, their limitations in the face of unmodeled dynamics and uncertainty pave the way for the emerging class of Artificial Intelligence (AI) Methods. The chapter examines how data-driven techniques, including Neural Networks and Fuzzy Logic, offer powerful alternatives capable of approximating complex nonlinear functions.

Finally, the chapter culminates in a formal Comparative Analysis that synthesizes these findings. By juxtaposing the verifiable robustness of conventional techniques against the adaptive flexibility of AI-based solutions, the discussion identifies the core technological trade-offs. This analysis reveals the specific research gaps that motivate the development of the hybrid diagnostic approach proposed later in this thesis.

1.1.1 Definition and Terminology

Degradation: It refers to the process or state where the performance, capability, or physical condition of a system or component gradually worsens over time due to factors like wear, aging, stress, or environmental exposure. It signifies a decline from an initial or optimal condition. Crucially, a system can be undergoing degradation without having failed yet, meaning it might still be performing its function, albeit less effectively or efficiently, and potentially still within its broadest operational limits. Degradation is often a precursor to failure [29].

Fault: It is defined as the hypothesized or confirmed root cause—an underlying abnormal physical or logical condition or defect within system hardware, software, parameters, or their interaction—that instigates an unpermitted deviation of at least one characteristic property or variable from the acceptable, standard behavior [30].

Malfunction: It refers to the state or occurrence where a system, component, or process performs its intended function incorrectly, erratically, or deviates significantly from its specified operational characteristics or performance criteria, without necessarily having reached a state of complete failure (cessation of function). It can be summarized as the effect or the manifestation of a fault in a degraded performance [31].

Failure: It is defined as the termination of the ability of a system, component, or process to perform its required function within specified performance limits, or an unacceptable deviation from its expected operational state. This event signifies that the entity under consideration is no longer meeting its design specifications or operational objectives, manifesting as an observable discrepancy (e.g., incorrect output, cessation of operation, degraded performance) [32].

Diagnosis: It is the systematic process of identifying the nature, location, and underlying cause (fault) of an undesired state or behaviour (such as a malfunction or failure) within a system, component, or process. This process involves acquiring and analysing observational data (e.g., sensor readings, alarms, performance metrics), utilizing system knowledge (models, design specifications, historical data), and employing logical reasoning or algorithmic techniques to infer the specific defect(s) responsible for the deviation from expected or specified operation. The primary goal of diagnosis is to provide actionable information that enables effective troubleshooting, repair, maintenance, or reconfiguration to restore the system to its intended functional state [28].

Figure 1.1 below represents the failure chronology of an industrial non-linear system, from normal working condition, going through abnormal condition and initial observed symptoms, to finally a breakdown.

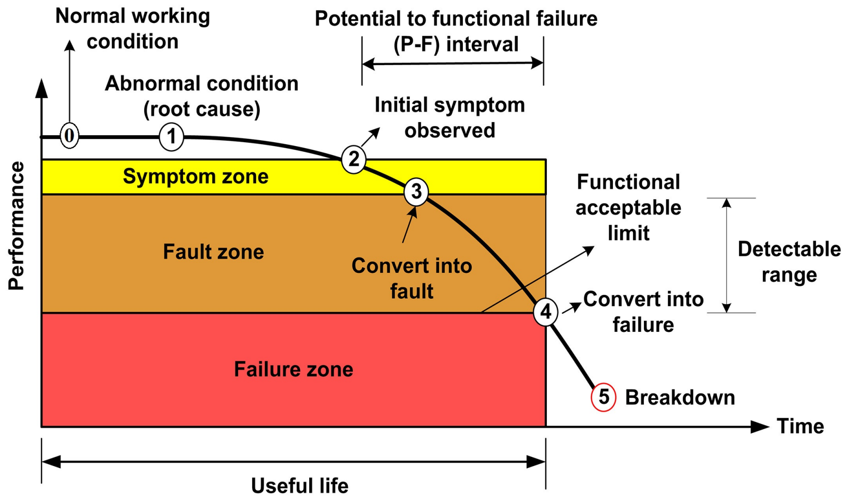

Figure 1.1: failure chronology of industrial non-linear system

1.1.2 Different Stages of Diagnosis

The procedure for diagnosing failures and degradations that may affect the different entities of an industrial process revolves around the following steps (these steps are also represented in *Figure 1.2* bellow) :

Data Acquisition: The systematic gathering and collection of relevant operational signals, parameters, event logs, and other informational sources from the monitored system, providing the necessary observational basis for subsequent analysis and assessment of its health status.

Detection: The process of analysing acquired system data to ascertain whether the system's behaviour deviates significantly from its expected or nominal operational state, thereby indicating the presence of an anomaly or potential fault requiring further investigation.

Localization (or Isolation): The stage following detection that aims to determine the specific physical or functional origin of the fault, narrowing down the source of the anomalous behaviour to a particular component, subsystem, connection, or region within the monitored system [33].

Identification: The process of determining the specific type, nature, and often the severity or magnitude of the fault that has been detected and localized, thereby characterizing the precise defect affecting the system component or function.

Taking Decision: The concluding stage where, based on the identified fault's type, location, and severity, appropriate responsive actions are determined and potentially initiated, such as triggering alarms, executing mitigating control strategies, initiating safety procedures, scheduling maintenance, or reconfiguring the system [34].

Figure 1.2: Diagnosis Stages

1.1.3 Diagnosis Performance Criteria

In order to make a clear judgement about some diagnosis system, there are several performance criteria's that must be achieved, among them we can mention :

Detection Rate (Sensitivity/Recall): Detection Rate, often termed sensitivity or recall, is formally defined as the probability that a diagnostic system correctly indicates the presence of a fault, given that a fault within its designated coverage scope has actually occurred; it quantifies the system's fundamental effectiveness in successfully identifying deviations from normal operation and avoiding missed detections (False Negatives) [32].

False Alarm Rate (FAR): The False Alarm Rate quantifies the probability or frequency with which the diagnostic system erroneously indicates the presence of a fault when the monitored system is actually operating correctly within its specified normal conditions; this criterion measures the system's reliability in avoiding spurious or unnecessary alarms (False Positives), which can erode operator confidence and lead to inefficient interventions.

Isolation Accuracy: Isolation Accuracy refers to the conditional probability that, given a fault has been correctly detected, the diagnostic system correctly identifies and distinguishes the specific type and/or location of that fault from all other potential fault candidates considered within its diagnostic scope; it measures the precision and correctness of the fault classification capability, crucial for directing effective remedial actions [35].

Diagnosis Time (Latency): Diagnosis Time, also known as diagnostic latency, represents the temporal delay measured from the instant a fault occurs (or becomes sufficiently manifest to be detectable) until the diagnostic system provides its conclusive output regarding the fault's detection, isolation, and/or identification; this metric evaluates the system's responsiveness, which is critical for timely intervention in dynamic or safety-related applications.

Robustness: Robustness characterizes the ability of a diagnostic system to maintain its specified performance levels (particularly in terms of detection rate, false alarm rate, and isolation accuracy) consistently despite the presence of operational uncertainties, such as measurement noise, inherent system parameter variations, unmodeled dynamics, disturbances, and fluctuations in operating conditions, ensuring reliable performance in real-world environments [36].

1.1.4 Fault Classification

To effectively develop and apply diagnostic strategies, it is crucial to understand the diverse nature of faults that can affect a system. Faults are not uniform; they differ significantly in their origin, manifestation, dynamics, and impact. Therefore, establishing systematic classification schemes is essential for characterizing fault behaviour, selecting appropriate detection and isolation methodologies, and ultimately designing effective fault-tolerant control systems [37].

1.1.4.1 According to Their Evolution in Time:

Every and each fault has a certain propagation pattern in time, the major 3 families patterns are :

Abrupt Fault: A system failure mode characterized by a sudden, often instantaneous, transition from a nominal operational state to a significantly degraded or non-functional state. This fault manifests as a discrete event, resulting in an immediate and substantial change in system parameters or capabilities [38], *figure 1.3* bellow represents the evolution of an abrupt fault in function of time.

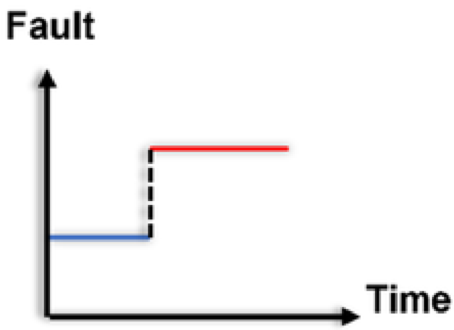

Figure 1.3: Temporal Characteristic of an Abrupt Fault

Intermittent Fault: A fault condition that manifests sporadically and non-persistently, causing a system to alternate between correct and incorrect operational behavior over time without intervention. The fault's activation is often stochastic or dependent on specific, transient operational or environmental conditions, posing significant challenges for detection and diagnosis [39], *figure 1.4* bellow represents the evolution of an intermittent fault in function of time.

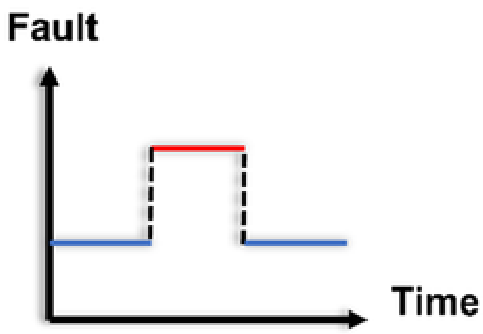

Figure 1.4: Temporal Characteristic of an Intermittent Fault

Gradual Fault (or Degradation Fault): A type of fault characterized by a slow, progressive deterioration in system performance or a drift in operational parameters over an extended period. This degradation typically results from cumulative damage mechanisms (e.g., wear, aging, fatigue) and may eventually lead to functional failure when performance crosses a predefined threshold [40], *figure 1.5* bellow represents the evolution of a gradual fault in function of time.

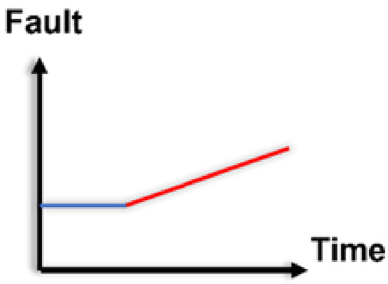

Figure 1.5: Temporal Characteristic of a Gradual Fault

1.1.4.2 According to Their Location

Every and each fault can happen and strike in different sections within the industrial systems, the three major locations are :

Actuator Faults: It refers to abnormal physical or operational conditions originating within the actuator component itself (e.g., electric motor, valve, solenoid, hydraulic/pneumatic cylinder) that degrade or prevent its intended function of accurately and reliably converting a control signal into a specific physical action (such as force, torque, motion, or flow regulation) [38]. These faults represent the underlying defects—like mechanical wear, stiction, internal leakage, electrical winding shorts/opens, sensor feedback errors within the actuator, or power stage failures—that cause the actuator's output behaviour to deviate significantly from the commanded input or design specifications, manifesting as malfunctions (e.g., incorrect position, sluggish response, being stuck) or complete failure, thereby necessitating diagnostic processes to identify and isolate the specific issue within the actuator mechanism [41].

Sensor Faults: It refers to abnormal physical or functional conditions originating within the sensing element, its transduction mechanism, or associated signal conditioning circuitry, which compromise or prevent its intended function of accurately and reliably converting a measured physical variable into a representative output signal. These faults constitute the underlying defects—such as bias offset, drift, incorrect gain (scaling error), excessive noise, complete failure (e.g., open/short circuit), stuck output, intermittent operation, or loss of precision—that cause the sensor's output to provide an erroneous, misleading, or absent representation of the actual physical quantity. Consequently, sensor faults can lead to incorrect monitoring, flawed control decisions, false alarms, and system instability, thus necessitating diagnostic techniques to detect, isolate, and identify the specific defect within the sensing system [37].

Component Faults: It refers to the underlying physical or logical abnormalities originating within specific constituent parts (components) or subsystems of a larger engineered system, distinct from dedicated input (sensor) or output (actuator) elements unless explicitly included. These faults represent the root defects—such as parameter drift in passive elements (resistors, capacitors), processing errors in logic units or microcontrollers, connection failures (e.g., solder joints, connectors), internal shorts/opens within integrated circuits, or material degradation—that cause a component to deviate from its specified operational characteristics or intended function at the component level. Such faults are the target of diagnostic procedures aiming to pinpoint the source of system-level malfunctions or failures by isolating the problem to one or more specific faulty components within the system's architecture [40].

Figure 1.6 bellow represent faults classification according to their potential location in the feedback control system [35].

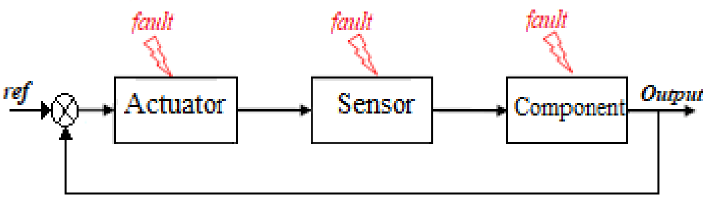

Figure 1.6: Fault Potential Locations in a Feedback Control System

1.1.4.3 According to Their Modelization Type

Every and each fault must be mathematically modelled so it can be studied and analysed, therefore, we can classify the faults according to their way of modelling which they are basically two types :

Additive Faults: It represents an unwanted signal or offset that is added to the nominal (fault-free) signal or system variable. It acts like an external disturbance or bias affecting the measurement or output [36]. In reality it is caused by a sensor bias, an actuator offset errors (e.g., a valve stuck partially open causing constant leakage flow), or an external disturbances (like a constant unmeasured force).

A state space representation with an additive fault can be presented as [34]:

$$\Sigma_{\text{add}} : \begin{cases} x_{k+1} = Ax_k + Bu_k + Ff_k \\ y_k = Cx_k + Du_k + Ef_k \end{cases} \tag{1.1}$$

With:

- Σ_{add} denoting for State-space representation of the system with additive faults,
- $x_{k+1} \in (n * 1)$ representing the state vector at the next time step,
- $x_k \in (n * 1)$ Represents the internal state of the system at time k,
- $u_k \in (m * 1)$ is the input vector,
- $y_k \in (p * 1)$ is the output vector,
- $A \in (n * n)$ is the State/System Matrix,
- $B \in (n * m)$ is the Input/Control Matrix,
- $C \in (p * n)$ is the Measurement Matrix,
- $D \in (p * m)$ is the Direct Transmission Matrix,
- $E \in (p * q)$ is the Fault Distribution Matrix,

- $f_k \in (p * 1)$ denotes for fault signal.

Multiplicative Faults: This type represents usually the actuators and sensors faults, where an actuator faces a brutal change in control action (scales the nominal signal) [31], In reality they are caused by a changes in sensor sensitivity, loss of actuator effectiveness (LOE), a degradation in component efficiency, or changes in process parameters like heat transfer coefficients.

A state space representation with a multiplicative fault can be presented as [34]:

$$\Sigma_{\text{mult}} : \begin{cases} x_{k+1} = Ax_k + B\Gamma_A u_k \\ y_k = Cx_k + D\Gamma_A u_k \end{cases} \tag{1.2}$$

With: Σ_{Mult} denoting for State-space representation of the system with multiplicative faults; and Γ_A represent the multiplicative fault factor.

Figure 1.7 bellow represent faults classification according to their modelization type (additive or multiplicative type) [33].

Figure 1.7: Multiplicative and Additive Fault Models

1.1.5 Challenges in Nonlinear System Diagnosis

The accurate and timely diagnosis of faults within engineered systems constitutes a critical prerequisite for ensuring operational safety, reliability, and efficiency, particularly in complex automation and control applications. While diagnostic methodologies for linear systems are relatively mature, a vast majority of real-world physical systems inherently exhibit nonlinear characteristics, causing their behaviour under nominal and faulty conditions to fundamentally diverge from linear approximations [42]. This inherent nonlinearity presents formidable challenges to the design and implementation of effective diagnostic frameworks, stemming from complex dynamic interactions, state-dependent fault manifestations, and the limitations of classical linear analysis tools. Consequently, achieving robust fault detection and isolation in nonlinear systems necessitates confronting a distinct set of obstacles, which demand specialized modelling techniques, advanced signal processing methods, and sophisticated diagnostic paradigms, forming the central focus of the subsequent discussion.

Diagnosing faults in nonlinear systems presents significant challenges compared to linear systems due to their inherent complexities.

-*Invalidity of Superposition and Complex Interaction Effects:* The absence of superposition in nonlinear systems is arguably the most fundamental challenge. It dictates that fault effects are not simply additive to the nominal system behaviour but interact complexly with the system's state and inputs, and also among themselves in multiple-fault scenarios, drastically complicating fault signature isolation, analysis, and the applicability of linear diagnostic paradigms.

-*State-Dependent Dynamics and Complex Fault Manifestations:* Nonlinear systems intrinsically exhibit behaviour that varies significantly with operating conditions, rendering diagnostic models or thresholds developed for one regime potentially invalid in others. Furthermore, faults can induce or interact with rich nonlinear phenomena (e.g., limit cycles, bifurcations, chaos), leading to complex, non-proportional, and often non-intuitive fault signatures that are difficult to characterize and generalize [25].

-*Modelling Complexity and Inherent Uncertainty:* Accurately capturing the relevant dynamics of nonlinear systems through mathematical models is substantially more challenging than for linear systems. This inherent difficulty in modelling, coupled with unavoidable uncertainties, poses a significant barrier to model-based diagnostic approaches, as model inaccuracies can obscure fault signatures or generate false diagnostic indications.

-*Difficulties in Diagnostic Signal Processing and Decision Logic:* Generating informative diagnostic signals (residuals), extracting robust fault features, and establishing reliable decision thresholds are significantly complicated by the state-dependent, often non-Gaussian nature of signals in nonlinear systems. This extends to the design and implementation of nonlinear observers or estimators, which are often required but entail higher complexity, potential convergence issues, and lack the general analytical tractability of their linear counterparts [30].

1.2 Conventional Diagnosis Methods for Nonlinear Systems

The challenge of fault detection and diagnosis (FDD) in non-linear systems stems fundamentally from their complex dynamics, where principles like superposition do not hold, rendering direct application of linear FDD techniques inadequate. Conventional approaches predominantly leverage model-based strategies, relying on the explicit mathematical representation of the system's non-linear behaviour. Prominent among these are observer-based methods, which extend linear concepts through techniques such as the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) for state estimation and residual generation, or employ dedicated non-linear observer designs like High-Gain Observers (HGO) and Sliding Mode Observers (SMO) known for robustness properties against certain uncertainties and disturbances. Another key conventional direction involves adaptations of analytical redundancy relations (ARR) or parity equations, though deriving computationally tractable and sensitive relations for general non-linear systems is often difficult. Parameter estimation techniques, which track deviations in model parameters indicative of faults, also constitute a conventional approach, frequently relying on recursive identification algorithms adapted for non-linear models [43, 44]. Despite their foundational importance, these conventional methods often face limitations concerning robustness to significant modelling uncertainties, computational complexity, sensitivity to operating point variations, and the difficulty of guaranteeing global convergence or performance, particularly for highly complex or poorly modelled non-linear processes.

1.2.1 Model Based Methods

Model-based fault detection and diagnosis (FDD) leverages analytical redundancy derived from a mathematical system model to predict nominal behaviour from measured inputs. The core principle involves generating a residual signal representing the discrepancy between model predictions and actual system measurements. Significant deviations in this residual, evaluated against thresholds or statistical tests, indicate fault occurrences. Prominent methodologies include state estimation using observers (e.g., Kalman filters, non-linear observers), parity space approaches assessing input-output consistency, and parameter estimation techniques tracking fault-induced model parameter changes [45]. The performance of model-based FDD fundamentally depends on model fidelity and robustness against uncertainties and noise.

1.2.1.1 Parameter Estimation techniques

These techniques constitute a class of methodologies predicated on the explicit identification of physical or behavioural parameters embedded within a mathematical model representing the monitored system. The fundamental premise is that incipient or manifest faults induce discernible deviations in one or more of these system parameters (e.g., resistance, friction coefficient, gain, time constant) from their nominal, fault-free values [46]. These techniques employ system identification algorithms, utilizing measured input and output signals (e.g., voltage, current, speed, position), to continuously or recursively estimate the values of selected model parameters online or offline. Fault detection is subsequently triggered when the estimated parameter trajectories significantly diverge from their expected nominal range or exhibit statistically improbable behaviour, while fault isolation may be achieved by associating specific parameter deviations with particular component failure modes or structural changes within the system [43].

The application of parameter estimation for diagnosis fundamentally relies on the availability of a parametric model of the system, typically derived from first principles or identified from experimental data, where specific coefficients or terms directly correspond to physical properties susceptible to change under faulty conditions [42]. Common identification algorithms employed include variations of Least Squares (LS), such as Recursive Least Squares (RLS) for online adaptation, Maximum Likelihood ML estimation, or Instrumental Variable (IV) methods designed to mitigate bias from noise. The diagnostic process involves several key steps :

- *Model Selection:* Choosing an appropriate model structure (e.g., state-space, ARX, OE, physical equations) that accurately captures the system dynamics relevant to the faults of interest and includes parameters sensitive to those faults.

- *Nominal Parameter Identification:* Establishing the baseline values and expected variance/uncertainty of the parameters under normal operating conditions, often through offline identification using fault-free data.

- *Online/Periodic Estimation:* Implementing an algorithm to track the parameters using real-time or periodically sampled operational data. This requires ensuring sufficient excitation in the system signals for reliable parameter convergence (identifiability) [47].

- *Decision Logic:* Defining criteria for declaring a fault based on parameter deviations. This often involves statistical tests (e.g., comparing estimates against confidence intervals derived from nominal data) or thresholding based on the magnitude and persistence of the parameter change.

- *Fault Isolation (Optional):* Utilizing knowledge about how different faults affect specific parameters (a fault signature matrix or diagnostic model) to pinpoint the root cause based on which parameter(s) have deviated [46, 47].

The Strengths of this approach include its potential to directly diagnose parametric faults and quantify their severity through the magnitude of the parameter change. It can also be effective for detecting slowly developing faults that manifest as gradual parameter drift. However, its might face several challenges that lies in the requirement for an accurate parametric model, potential issues with parameter identifiability under certain operating conditions, and sensitivity to measurement noise and unmodeled dynamics. Furthermore, distinguishing parameter variations caused by genuine faults from those arising due to normal changes in operating points or environmental conditions necessitates robust estimation and decision-making strategies.

1.2.1.2 Parity Space Relations

Parity Space Relations constitute a fundamental model-based diagnostic technique centred on exploiting analytical redundancy inherent within a system's mathematical model to generate diagnostic signals, termed residuals. These residuals are formulated through algebraic manipulation of the model equations (typically discrete-time state-space or input-output representations) over a finite time horizon, such that they are ideally zero under nominal (fault-free) operating conditions and demonstrably non-zero in the presence of faults affecting the system dynamics or sensor/actuator paths [48]. Crucially, parity relations are often designed to be decoupled from, or insensitive to, unknown quantities like initial system states or certain classes of disturbances, thereby enhancing diagnostic robustness.

The core idea behind the Parity Space approach is to leverage the inherent consistency constraints imposed by the system model on the temporal evolution of its inputs and outputs. For a linear discrete-time system model, equations relating inputs, outputs, and states at different time steps can be stacked over a chosen time window of length s. This results in an overdetermined set of equations relating a sequence of known inputs u(k) and outputs y(k) to the unknown initial state x(k-s) within that window [49].

The derivation process seeks a linear transformation (represented by a matrix or vector W) that effectively eliminates the unknown initial state term from the stacked equations. This transformation projects the measured input/output data onto a subspace the "parity space" which is orthogonal to the subspace spanned by the system's free response originating from the initial state. The resulting equation, r(k) = W* [vector of inputs/outputs over window], defines the parity relation, and r(k) is the residual signal [49].

Under fault-free conditions and assuming an accurate model, the measured inputs and outputs satisfy the underlying system equations, and thus the residual r(k) will be (theoretically) zero, or close to zero within the bounds of noise and modelling uncertainty. When a fault occurs, it violates the assumptions embedded in the model equations (e.g., by changing a system parameter, adding an offset, or altering the input/output relationship). This inconsistency causes the measured data to no longer satisfy the original model constraints, resulting in a non-zero residual r(k), thereby detecting the fault [50].

The Strengths of parity space approach include its systematic design procedure rooted in linear algebra, relatively low computational cost for implementing the residual calculation once W is derived, and the potential for structured residual design for fault isolation. However, it might include sensitivity for modelling errors, noise (which can be amplified by the transformation W), and the trade-offs involved in selecting the window length s (affecting detection delay, noise sensitivity, and computational load) [47, 48].

1.2.1.3 State observers

They represent a pivotal class of algorithms within model-based diagnosis, fundamentally designed to reconstruct the internal state vector of a dynamic system based on knowledge of its mathematical model (typically formulated in state-space representation) and real-time measurements of its inputs and outputs. In the context of fault detection and isolation (FDI), their primary utility stems from the generation of diagnostic signals, known as residuals, which quantify the discrepancy between the observer's estimated system output and the actual measured system output [51]. Significant deviations of these residuals from their expected nominal (near-zero) behaviour under fault-free conditions serve as indicators of potential system malfunctions, sensor faults, or actuator faults.

The underlying principle of a state observer is to create a dynamic replica, or simulation, of the monitored system running in parallel with the actual process. The observer model takes the same measured inputs as the real system and calculates an estimate of the system's internal states $\hat{x}$. A crucial component is a correction term, typically proportional to the output estimation error (the difference between the measured output y and the output predicted by the observer based on its state estimate, $\hat{y} = C\hat{x} + Du$). This correction term continuously nudges the observer's state estimate towards the true system state, aiming for $\hat{x}(t)$ to converge to x(t) over time [52].

The residual signal r(t) is formally defined as [51]:

$$r(t) = y(t) - \hat{y}(t) \tag{1.3}$$

$$r(t) = y(t) - (C.\hat{x}(t) + D.u(t)) \tag{1.4}$$

With:

- n: Number of states, m: Number of inputs, p: Number of outputs,
- $r(t) \in (p * 1)$: is the Residual Signal,
- $y(t) \in (p * 1)$: is the Measured Output Vector,
- $\hat{y}(t) \in (p * 1)$: is the Estimated Output Vector
- $\hat{x}(t) \in (n * 1)$: is the Estimated State Vector
- $u(t) \in (m * 1)$: is the Control Input Vector

Under ideal, fault-free conditions, assuming an accurate model and negligible noise, the state estimate $\hat{x}$ converges to the true state x, causing the estimated output $\hat{y}$ to match the measured output y, resulting in a residual r(t) close to zero. When a fault occurs, it alters the system's dynamics or its input/output relationships in a way not captured by the observer's internal model. This discrepancy prevents the observer from accurately tracking the system's behaviour, leading to a persistent, non-zero residual signal, thus detecting the fault [53].

The Strengths of observer-based methods include their systematic design framework, inherent capability to handle dynamic systems, provision of state estimates potentially useful for control, and the optimality properties (for KF variants) in handling noise. However it might face challenges in the strong dependence on the accuracy of the system model, the need for the system to be observable, potential computational complexity (especially for nonlinear observers), and the tuning effort required to balance performance and robustness.

1. ***Luenberger Observer:*** The Luenberger Observer constitutes a foundational deterministic state estimation algorithm within the framework of model-based diagnosis, specifically designed for linear time-invariant (LTI) systems commonly encountered in automation and electrical engineering applications. It operates by constructing a dynamic model that runs in parallel to the actual system, utilizing the same measured input signals and incorporating a corrective feedback term proportional to the discrepancy between the measured system output and the observer's estimated output [54]. for the diagnostic purposes, the primary function of the Luenberger observer is the systematic generation of a residual signal defined as the output estimation error (r = y - $\hat{y}$) which ideally remains

near zero under nominal, fault-free operating conditions but deviates significantly in the presence of faults affecting the system dynamics, sensors, or actuators, thereby enabling fault detection [55].

The Luenberger observer is predicated on the availability of an accurate LTI state-space model of the monitored process [54]:

$$\begin{cases} \dot{x}(t) &= A.x(t) + B.u(t) \\ y(t) &= C.x(t) + D.u(t) \end{cases} \tag{1.5}$$

Where x is the state vector, u is the input vector, y is the output vector, and A, B, C, D are system matrices of appropriate dimensions.

The observer itself is a dynamical system described by [56]:

$$\begin{cases} \dot{\hat{x}}(t+1) &= A.\hat{x}(t) + B.u(t) + L[y(t) - \hat{y}(t)] \\ \hat{y}(t) &= C.\hat{x}(t) + D.u(t) \end{cases} \tag{1.6}$$

Here, $\dot{\hat{x}}(t+1)$ represents the estimated state vector, $\hat{y}$ is the estimated output vector, and L is the crucial observer gain matrix.

The term [y(t) - $\hat{y}$(t)] represents the output estimation error, which serves directly as the residual signal r(t) for diagnostic purposes. This residual is fed back through the observer gain L to correct the state estimate dynamics. The design objective for L is critical; it must be chosen such that the estimation error dynamics, governed by [57]:

$$\dot{e}(t) = (A - LC).e(t) \tag{1.7}$$

With the error defined as:

$$e(t) = x(t) - \hat{x}(t) \tag{1.8}$$

Is stable and converges sufficiently quickly. This is typically achieved through pole placement, assigning the eigenvalues of the matrix $(A - LC)$ to desired locations in the left-half complex plane, provided the system pair (A, C) is observable.

Figure 1.8 bellow, represents a Model-Based Fault Detection System using a Luenberger Observer, where the Physical System is in top half, and the luenberger observer is in bottom dashed box [55].

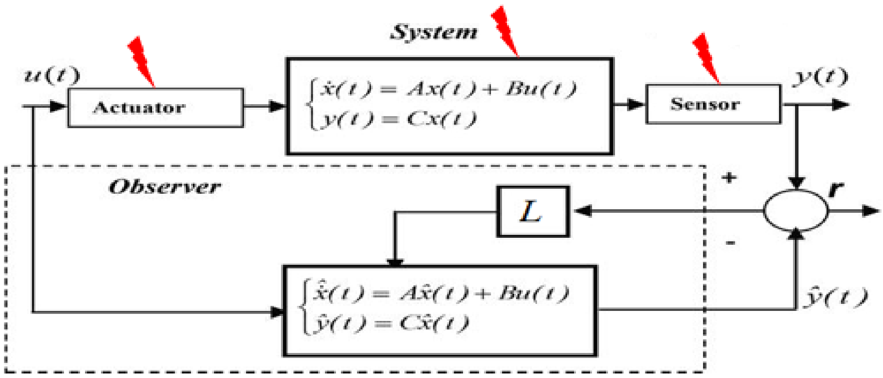

Figure 1.8: Luenberger Observer

The Strengths of the Luenberger observer in diagnosis include its conceptual simplicity for linear systems, systematic design procedure via pole placement, and relatively low computational burden compared to stochastic observers. While the limitations include its inherent restriction to linear systems (though sometimes applied to linearized nonlinear systems), sensitivity to modelling errors and parameter uncertainties (which directly affect the residual), lack of inherent mechanism to optimally handle stochastic noise (unlike Kalman filters), and the requirement for system observability [54–57]. The choice of observer gain L often involves a trade-off between fast error convergence (potentially amplifying noise) and noise robustness (potentially slowing down fault detection).

2. ***Extended Kalman Filter:*** The Extended Kalman Filter (EKF) is a recursive Bayesian estimation algorithm fundamentally employed for state and parameter estimation in discrete-time, non-linear dynamic systems operating under uncertainty. It addresses systems where the process dynamics or measurement relationship are non-linear, situations commonly encountered in automation and electrical engineering. The EKF approximates the system's state probability distribution as Gaussian and propagates its mean (the state estimate $\hat{x}$) and covariance (P) over time. This propagation relies on analytically linearizing the non-linear functions (state transition f and measurement h) around the most recent state estimate at each time step, using Jacobian matrices (F_k, H_{k+1}). The system is typically assumed to be affected by additive, zero-mean Gaussian process noise (W_k with covariance Q_k) and measurement noise (V_{k+1} with covariance R_{k+1}), which are mutually uncorrelated and white [58].

   Within the discipline of fault diagnosis, the EKF provides a powerful framework for model-based residual generation, particularly when dealing with the inherent non-linearities found in many real-world systems (e.g., actuator saturation, non-linear component characteristics, complex process kinetics, robotic dynamics). The core diagnostic strategy involves implementing the EKF using a nominal model of the system that is, mathematical representations (f, h) describing the system's expected behaviour in a fault-free condition. The EKF then processes the actual measurements (y_{k+1}) obtained from the potentially faulty physical system [59]. By comparing the filter's predictions (based on the fault-free model) with the actual measurements, the EKF generates a residual signal that quantifies the discrepancy. This ability to estimate internal states and systematically account for non-linearities and noise makes the EKF a cornerstone for detecting deviations indicative of faults [58].

The diagnostic process leveraging the EKF revolves around the filter's recursive prediction and update cycle, specifically focusing on the generated residual. Starting with an initial estimate $(\hat{\mathbf{x}}_{0|0}, \mathbf{P}_{0|0})$ and $(x0|0, P0|0)$, the filter iterates through the following steps for each time interval k [59]:

i. ***Prediction Ahead:*** Based on the previous state estimate $\hat{x}_{k|k}$ and input u_k, predict the state and its error covariance for the next time step $k+1$, assuming fault-free dynamics (f) and known process noise covariance Q_k [60]:

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u_k) \tag{1.9}$$

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k \tag{1.10}$$

(Where $F_k = \partial f/\partial x$ evaluated at $\hat{x}_{k|k}, u_k$)

ii. ***Update based on Measurement:*** Incorporate the actual measurement y_{k+1} taken from the system at time $k+1$. This involves calculating the *Kalman Gain* K_{k+1} and updating the state estimate and error covariance, using the nominal measurement model h and known measurement noise covariance R_{k+1} [61]:

-Compute Kalman Gain:

$$K_{k+1} = P_{k+1|k} H_{k+1}^T \cdot \left(H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}\right)^{-1} \tag{1.11}$$

- Compute Residual (Innovation):

$$\tilde{y}_{k+1} = y_{k+1} - h(\hat{x}_{k+1|k}) \tag{1.12}$$

- Update Estimate:

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \cdot \tilde{y}_{k+1} \tag{1.13}$$

- Update Error Covariance [59]:

$$P_{k+1|k+1} = (I - K_{k+1} \cdot H_{k+1}) \cdot P_{k+1|k} \tag{1.14}$$

(where $H_{k+1} = \partial h/\partial x$ evaluated at $\hat{x}_{k+1|k}$)

with:

- n: Number of states, m:Number of measurements.
- $K_{k+1} \in (n * m)$: is Kalman Gain Matrix,
- $P_{k+1|k} \in (n * n)$: is the Predicted (A Priori) Error Covariance Matrix,
- $H_{k+1} \in (n * m)$: is the Measurement Matrix (or Observation Matrix),
- $R_{k+1} \in (m * m)$: is the Measurement Noise Covariance Matrix,
- $H^T \in (n * m)$: is the Transpose of the Measurement Matrix.

Figure 1.9 bellow, represents the algorithm flowchart for an Extended Kalman Filter [61].

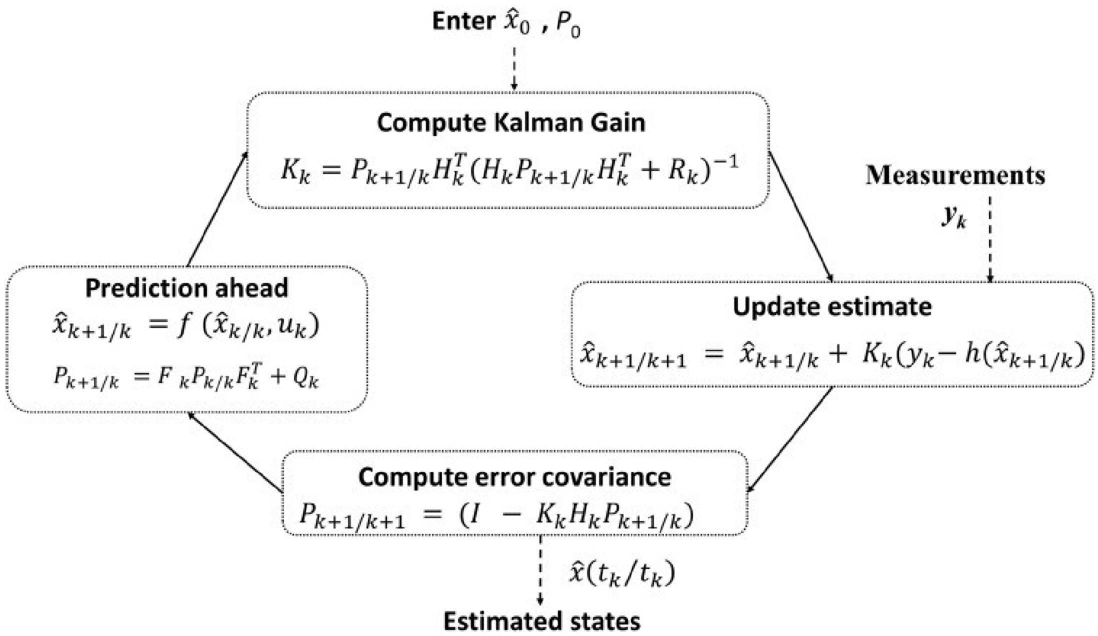

Figure 1.9: Algorithm Flowchart for an Extended Kalman Filter

The crucial element for fault diagnosis is the residual vector $\tilde{y}_{k+1}$. It represents the difference between the actual measurement y_{k+1} and the measurement predicted by the EKF $h(\hat{x}_{k+1|k}$ based on the *nominal, fault-free model.* If the system operates as expected (no faults) and the model/noise parameters are accurate, the residual sequence $\tilde{y}_{k+1}$ should be a zero-mean, white noise sequence with known covariance $S_{k+1} = H_{k+1}.P_{k+1|k}.H_{k+1}^T + R_{k+1}$. The occurrence of a fault introduces behaviour inconsistent with the nominal model f and/or h, causing the residual $\tilde{y}_{k+1}$ to deviate statistically from these fault-free properties. Analysing these deviations (e.g., using statistical tests on its mean or variance) allows for fault detection and potentially isolation. The final output $\hat{x}_{k+1|k+1}$ represents the estimated system state under the assumptions made [58–61].

3. ***Sliding Mode Observer:*** It is a non-linear, deterministic state estimation technique renowned for its intrinsic robustness against a class of modelling uncertainties, parameter variations, and external disturbances. It is architecturally characterized by the incorporation of a discontinuous feedback term, typically based on the signum function of the output estimation error (or a function thereof), into the observer dynamics [62].

The fundamental operating principle relies on designing this discontinuous term such that the trajectory of the estimation error is steered towards and subsequently constrained into a predefined sliding manifold (or surface) within the state error space [63]. To achieve this convergence in finite time.

$$S = \{e \in \mathbb{R}^n | \sigma(e) = 0\} \tag{1.15}$$

with:

- n: number of states, p: number of outputs,
- $S \in \mathbb{R}^n$: Sliding Manifold,
- $e \in (n * 1)$: is the State Estimation Error Vector,
- $\sigma(e)$: is the Sliding Function (or Switching Function).

Once established, the resulting sliding motion forces the error dynamics to conform to the dynamics defined by the sliding surface [64].

$$\sigma(e) = 0 \tag{1.16}$$

And:

$$\frac{d\sigma(e)}{dt} = 0 \tag{1.17}$$

Crucially, these constrained dynamics are, by design, independent of (or significantly less sensitive to) certain "matched" uncertainties and disturbances affecting the system. The equivalent output error injection signal representing the average value of the discontinuous term required to maintain the sliding motion implicitly reconstructs the unknown inputs or state-dependent terms necessary to enforce the sliding condition, thereby containing rich information about system anomalies [65].

Formally, for a system [62]:

$$\frac{dx}{dt} = f(x, u, d_1, w_1) \tag{1.18}$$

And:

$$y = h(x) + (d_2, w_2) \tag{1.19}$$

Where d_1, d_2 represent disturbances/faults and w_1, w_2 represent uncertainties.

An SMO is often structured as [63]:

$$\frac{d\hat{x}}{dt} = f_0(\hat{x}, u) + G(y - \hat{y}) + v \tag{1.20}$$

Where f_0 is the nominal model, G is a linear gain (optional or combined), and v is the discontinuous sliding mode term, frequently of the form:

$$v = \rho(\hat{x}, y, u).\text{sgn}[\sigma(y - \hat{y})] \tag{1.21}$$

The sliding surface σ and the switching gain ρ are designed using Lyapunov stability theory to guarantee finite-time convergence to $\sigma = 0$ and subsequent sliding motion, provided the uncertainties/faults satisfy certain matching conditions and boundedness assumptions [64].

In the field of fault detection and isolation (FDI), the Sliding Mode Observer provides a robust framework for fault detection and isolation (FDI), particularly effective in scenarios where precise system models are unavailable or where systems are subject to significant operational variations and disturbances that are difficult to model accurately [65,66]. Its utility in diagnosis stems primarily from its robustness properties and the information embedded within the equivalent injection signal.

Unlike observers that rely on the innovation $y - \hat{y}$ as the primary residual (like *Kalman Filters* or *Luenberger observers*), the SMO generates a fundamentally different diagnostic signal. The core idea is that the discontinuous term v actively compensates for discrepancies between the observer model $f_0(\hat{x}, u)$ and the actual system dynamics projected onto the output, including the effects of matched uncertainties *and* faults. The equivalent output error injection signal v_{eq} obtained typically by low-pass filtering v to extract its average value necessary to maintain $\sigma \approx 0$, serves as the primary residual. Under nominal (fault-free) conditions, v_{eq} reflects the magnitude and structure of the modelled uncertainties and disturbances the SMO is designed to reject [66].

1.2.2 Signal Based Methods

Signal-based diagnostic methodologies represent a distinct class of fault detection and isolation (FDI) techniques within automation and electrical systems, characterized by their direct processing and analysis of measured system outputs (signals) without explicit reliance on a fundamental mathematical model of the system's behaviour. These approaches operate on the premise that faults manifest as detectable alterations in the statistical properties, frequency content, or temporal patterns inherent within sensor readings such as voltages, currents, vibrations, or temperatures [67].

The core methodologies often involve feature extraction from raw signals across different domains. In the time domain, this may include calculating statistical moments like mean, variance, skewness, kurtosis, or employing limit checking. Frequency domain analysis commonly utilizes the Fast Fourier Transform (FFT) for spectral analysis to identify characteristic fault frequencies [68]. For signals with changing characteristics over time, time-frequency domain techniques like wavelet transforms or the Short-Time Fourier Transform (STFT) are employed to capture non-stationary features.

The diagnostic decision is typically made by comparing these extracted features against predefined thresholds or patterns established during normal operation, often learned from historical data or through expert knowledge.

While signal-based methods can offer advantages in terms of lower modelling effort and computational simplicity for specific, well-characterized fault signatures, their efficacy can be sensitive to variations in operating conditions (e.g., load, speed) that may alter signal characteristics independently of faults, and they may struggle to diagnose novel or unanticipated fault modes lacking predefined signal patterns [69].

The *Figure 1.10* bellow represents the process of Signal-Based Fault Diagnosis [67].

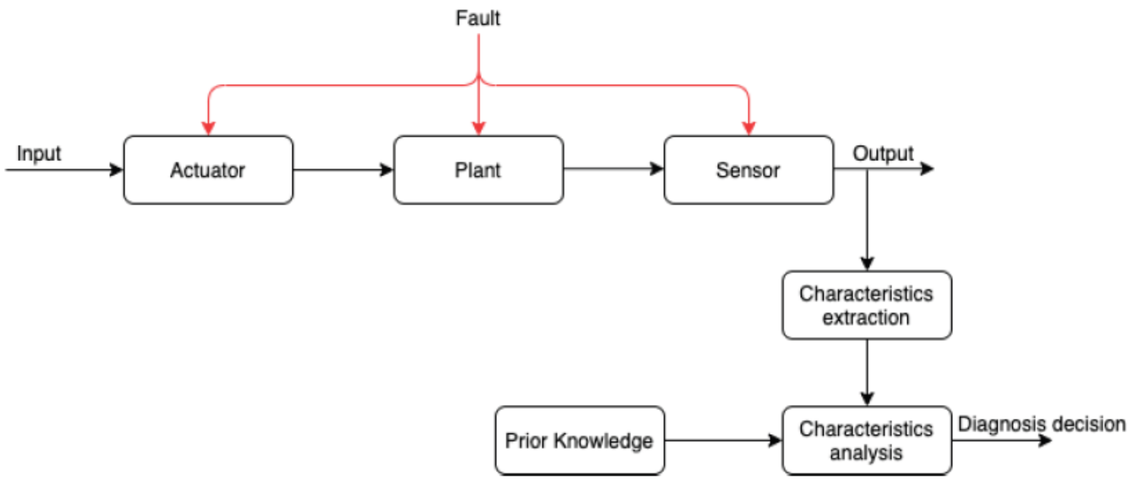

Figure 1.10: Signal Based Diagnosis

1.2.2.1 Spectral Analysis

Spectral Analysis encompasses a suite of signal processing techniques fundamental to signal-based diagnosis. Its primary focus is decomposing time-domain signals $x[n]$ acquired from automation or electrical systems into their constituent frequency components to reveal underlying periodicities or characteristic spectral signatures. This is primarily achieved using the Fourier Transform, particularly its efficient implementation for discrete signals, the Fast Fourier Transform (FFT), which calculates the Discrete Fourier Transform (DFT). The DFT transforms a sequence of N time-domain samples $x[n]$ into a sequence of N frequency-domain components $X[k]$, represented as [68]:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi kn}{N}}, \quad \text{for } k = 0, 1, \ldots, N-1 \tag{1.22}$$

Where $x[n]$ is the signal value at time sample n, $X[k]$ is the complex frequency component corresponding to frequency k, N is the total number of samples, and j is the imaginary unit. From $X[k]$, representations such as the amplitude spectrum $|X[k]|$ or the power spectral density ((PSD), often proportional to $|X[k]|^2$) are derived, quantitatively describing the signal's magnitude or energy distribution across different frequencies.

Spectral analysis is instrumental for diagnostics as it allows for the identification of fault-specific frequency patterns within these spectral representations. Examples include detecting abnormal harmonics in current/voltage signals indicative of power electronic converter faults, identifying characteristic frequencies corresponding to bearing defects in rotating machinery vibration signals, or observing shifts in system resonant frequencies [67]. The diagnosis relies on detecting deviations in the spectral content compared to the expected spectrum under nominal operating conditions.

1.2.2.2 Statistical Methods

a) *Principal Component Analysis (PCA)*: it is an unsupervised multivariate statistical technique employed within signal-based diagnostics to achieve dimensionality reduction and feature extraction from correlated sensor data arrays. By performing an orthogonal linear transformation, PCA identifies a set of uncorrelated variables known as principal components (PCs), ordered such that the initial PCs capture the maximum variance present in the original dataset [70].

   To perform diagnosis purposes the PCA gets applied to datasets representing nominal system operation to establish a reduced-dimension model of normal behaviour; subsequently, deviations of new process measurements from this established low-dimensional subspace, often quantified by metrics like the squared prediction error ((SPE) or Q-statistic) or Hotelling's T^2 statistic, serve as indicators for fault detection, signalling anomalies inconsistent with the learned nominal data structure [68, 70].

b) *Fisher Discriminant Analysis (FDA) / Linear Discriminant Analysis (LDA)*: Fisher Discriminant Analysis (FDA), commonly implemented as Linear Discriminant Analysis (LDA), is a supervised dimensionality reduction and classification technique. Within signal-based diagnosis, it is utilized to explicitly enhance the separability between predefined operational states (e.g., normal condition and various specific fault classes). Unlike PCA's variance-maximization objective, FDA/LDA seeks a projection subspace (defined by projection vectors w) that maximizes the ratio of between-class variance to within-class variance. This objective is often expressed via maximizing Fisher's criterion [71]:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{1.23}$$

   Where S_B is the between-class scatter matrix (measuring separation of class means) and S_W is the within-class scatter matrix (measuring compactness of each class). This process optimally discriminates among the known categories based on features extracted from system signals [69].

Its application in diagnostics involves training the discriminant functions using labelled datasets corresponding to different system health states; subsequently, newly acquired signal features are projected onto these discriminant axes to facilitate the classification of the system's current operational state into one of the predefined fault categories, thereby enabling fault isolation [71].

1.2.3 Advantages and Limitations of Conventional Methods

Fault Diagnosis and Prognosis (FDP) are critical functions within modern automation and electrical engineering systems, ensuring reliability, safety, and operational efficiency. Historically, diagnostic methodologies have largely fallen into two primary paradigms: *model-based* and *signal-based (or data-driven)* approaches. While these conventional techniques have yielded significant successes and form the bedrock of many industrial monitoring systems, their efficacy faces growing challenges when confronted with the increasing complexity, non-linearity, interconnectivity, and operational variability inherent in contemporary engineered systems.

1.2.3.1 Advantages of Conventional Methods

Despite their limitations in dealing with complex modern systems, conventional methods offer distinct advantages that make them a foundation for current industrial practice:

- *Physical Interpretability (Model-Based)*: Model-based methods leverage explicit mathematical representations of the system's physics [43]. This provides direct physical insight, allowing for clear root cause analysis and understanding of the fault mechanism, which is essential for corrective maintenance.

- *Predictability in Novel Conditions (Model-Based)*: Once a high-fidelity model is established, these methods can predict system behaviour and potential faults even under operating conditions or fault types that were not explicitly tested, provided the underlying physics remain valid [45].

- *Minimal Data Requirement (Model-Based)*: Unlike data-driven methods, model-based techniques do not require substantial amounts of historical fault data, which is often scarce for critical industrial systems [44].

- *Computational Efficiency (Signal-Based)*: Standard signal-based techniques (like FFT) can be computationally fast and easy to implement in hardware for online monitoring [47], relying only on basic processing of sensor data.

1.2.3.2 Limitations of Conventional Methods

The major shortcomings of conventional methods are categorized by their reliance on models or data, as summarized below.

a) Limitations of Model-Based Diagnosis Methods

Model-based diagnosis fundamentally relies on the comparison between observed system behaviour and the behaviour predicted by a mathematical model.

- ***Model Fidelity and Complexity***: The prerequisite for all model-based methods is an accurate mathematical model. Developing such high-fidelity models for complex, non-linear, and time-varying systems is exceptionally challenging [48]. Unmodeled dynamics and disturbances (e.g., noise, external forces) can easily be misinterpreted as fault signatures, leading to false alarms or missed detections.

- ***Technique-Specific Constraints***: State Observers (EKF, SMO) and Parity Space Relations often inherit restrictions, such as the assumption of linearity (or requiring linearization), sensitivity to initial conditions, and the violation of Gaussian noise assumptions (for EKF). Furthermore, techniques like Parameter Estimation face challenges with parameter unidentifiability, and Sliding Mode Observers suffer from chattering [52, 58].

b) Limitations of Signal-Based Diagnosis Methods

Signal-based methods bypass the need for explicit physical models, instead extracting statistical features directly from sensor data to infer system health.

- ***Data Scarcity and Quality***: The most significant limitation is the reliance on substantial amounts of representative, historical, and often labeled data covering various operational modes and health states [67]. Obtaining sufficient, high-quality data, especially for rare fault conditions or new systems, is often a major hurdle.

- ***Lack of Physical Insight and Feature Dependency***: These methods often lack direct physical interpretation, making root cause analysis difficult (black-box tendency). Their performance is critically dependent on laborious manual feature engineering, where the extracted features may not generalize well to unseen conditions or fault types [68].

- ***Sensitivity and Assumptions***: Techniques like PCA and FDA often rely on linearity and Gaussian distribution assumptions, which are frequently violated in real-world, non-linear systems. Furthermore, Spectral Analysis often relies on a signal stationarity assumption [69].

1.2.3.3 Comparative Summary

The distinct requirements, assumptions, and limitations of the two primary conventional diagnosis paradigms are summarized in *Table 1.1*.

Table 1.1: Comparative summary between Model-Based and Signal-Based diagnosis methods

Method Category	**Specific Techniques Exemplified**	**Key Limitations**	**Core Requirements / Assumptions**
Model-Based	- Parameter Estimation, - Parity Space Relations, - State Observers (Luenberger, EKF, SMO).	- High dependence on accurate model fidelity, - Difficulty handling complexity, non-linearity, time-variation, - Sensitivity to unmodeled dynamics & disturbances, - Linearization errors (EKF), chattering (SMO). - Tuning complexity (Observers, Parity).	- Accurate mathematical model. - Known system structure. - Noise characteristics often assumed (e.g., Gaussian for EKF). - Analytical redundancy (Parity).
Signal-Based	- Spectral Analysis, - Principal Component Analysis (PCA), - Fisher Discriminant Analysis (FDA)	- Need for large, representative historical datasets. - Performance heavily dependent on feature engineering. - Sensitivity to operating conditions & noise. - Difficulty interpreting models (black-box tendency). - Linearity/Stationarity assumptions (PCA, Spectral). - Requires labelled data (FDA). - Difficulty with novel/incipient faults.	- Sufficient historical data. - Assumption of statistical difference between health states. - Features capture fault effects. - Labelled data (Supervised methods like FDA).

1.3 Artificial Intelligence Diagnosis Methods for Nonlinear Systems

Artificial Intelligence (AI) based diagnostic approaches, encompassing primarily Machine Learning (ML) and Deep Learning (DL) paradigms, constitute a data-centric framework for the detection, isolation, and identification of malfunctions within complex systems. These methods are particularly relevant for the non-linear dynamic systems characteristic of modern automation and electrical engineering applications, such as power electronics, robotic systems, process control, and electric drives [72].

Diverging significantly from traditional model-based techniques, which depend heavily on accurate mathematical representations of system physics (often analytically intractable for complex systems), AI-based diagnosis leverages computational algorithms to learn intricate patterns, correlations, and latent feature representations directly from operational data [73]. This data can include sensor time series, control signals, spectral data, or images.

AI methods excel in implicitly capturing the complex, non-linear mappings between high-dimensional input data and underlying system health states, thereby circumventing the need for explicit model derivation or linearization. Techniques vary widely, including established methods like Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs), as well as more recent deep learning approaches. Convolutional Neural Networks (CNNs) are often used for spatial or spatio-temporal feature extraction (e.g., from images or spectrograms), while (RNNs), including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) variants, are adept at analysing time-series data. Unsupervised methods like Autoencoders are also frequently employed for anomaly detection by learning compressed representations of normal operation [74].

These diverse AI techniques enable the development of diagnostic systems capable of identifying subtle incipient faults, adapting to time-varying operating conditions, and potentially discovering unforeseen failure modes by recognizing deviations from learned nominal behaviour within the data itself. This offers enhanced robustness and sensitivity, particularly in scenarios where traditional model-based or signal-processing methods face limitations due to model uncertainty or the sheer complexity of non-linear interactions.

1.3.1 Fuzzy Logic Based Diagnosis

Fuzzy Logic Based Diagnosis (FLBD) represents a knowledge-based fault diagnosis methodology particularly suited for complex, non-linear dynamic systems prevalent in engineering domains. It leverages the principles of fuzzy set theory and fuzzy inference systems to systematically handle the detection, isolation, and identification of faults [72].

A key strength of FLBD lies in its ability to provide a formal framework for representing and processing imprecise, uncertain, or qualitative information. This includes ambiguous sensor measurements, heuristic expert knowledge, and linguistic descriptions of fault symptoms. This capability allows diagnostic reasoning under conditions where traditional analytical model-based or purely data-driven approaches may face significant limitations due to model uncertainty, system complexity, or the non-linear nature of fault propagation [73].

The application of diagnostic techniques to non-linear systems in engineering field (e.g., power electronic converters, robotic manipulators, chemical processes, power grids) presents considerable challenges. These systems often exhibit complex behaviours (e.g., bifurcations, limit cycles, chaotic dynamics) that are difficult to capture accurately with conventional mathematical models [74]. Furthermore, faults in such systems can manifest in subtle, non-proportional ways, and their effects can be masked by inherent system non-linearities and operational uncertainties.

FLBD emerges as a potent paradigm to address these challenges. By incorporating principles of approximate reasoning, akin to human expert judgment, it allows for the development of diagnostic systems that can interpret complex system behaviour and fault manifestations effectively, even in the presence of significant uncertainty and non-linearity [75].
The *Figure 1.11* bellow represents the general workflow for a fuzzy logic based diagnosis [75].

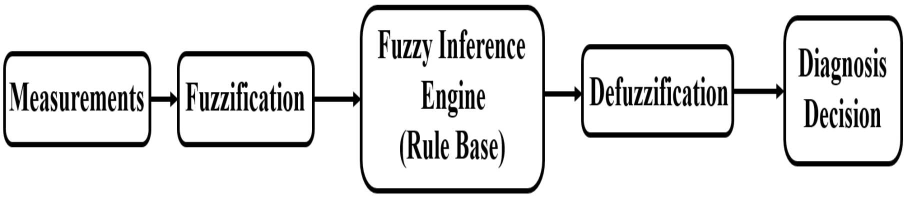

Figure 1.11: General Workflow of Fuzzy Logic Based Diagnosis

1.3.2 Neural Network Based Diagnosis

Artificial Neural Network Based Diagnosis constitutes a data-driven fault diagnosis paradigm employing ANNs. These networks are computational models, inspired by biological neural structures, designed to learn complex, potentially highly non-linear mappings between system measurements (symptoms) and underlying fault conditions within dynamic systems [76].

ANNs achieve this by processing information through interconnected nodes (neurons) organized in layers. The fundamental computation within a neuron involves calculating a weighted sum of its inputs and then applying a non-linear activation function (ϕ). The output (y) of a single neuron can be represented as [77]:

$$y = \varphi\left(\sum_{i=1}^{n} \omega_i x_i + b\right) \tag{1.24}$$

Where x_i are the inputs, ω_i are the corresponding weights, b is a bias term, and φ is the activation function. By adjusting these weights and biases during a training process based on example data, the network learns to approximate arbitrary non-linear functions. Applied specifically to non-linear systems, this capability allows ANN's to circumvent the need for explicit mathematical models of the system or fault mechanisms, which are often difficult or intractable to derive for complex non-linear dynamics.
The *Figure 1.12* bellow represents the basic artificial neuron model for diagnosis process [78].

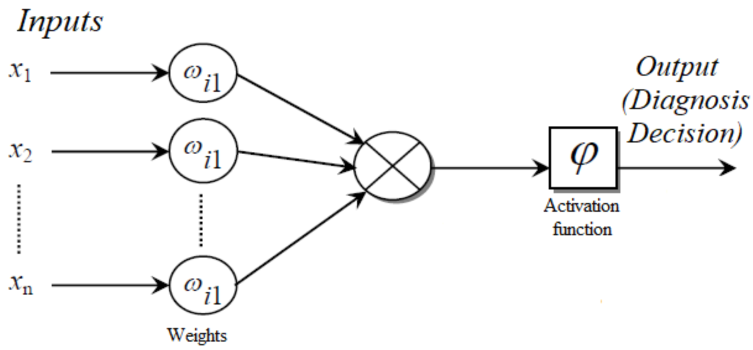

Figure 1.12: Basic Artificial Neuron Model for Diagnosis Process

The diagnosis of non-linear systems in fields like automation (e.g., robotic systems, manufacturing processes) and electrical engineering (e.g., power electronic converters, electric machines, power grids) is challenged by intricate system behaviours, parameter uncertainties, and the complex propagation of fault effects. Traditional model-based diagnosis often struggles due to difficulties in obtaining accurate, comprehensive non-linear models. ANNBD thus emerges as a powerful alternative or complementary approach, fundamentally relying on the principle of learning function mappings directly from historical or simulated operational data [76–78]. ANN's emerges as a powerful alternative or complementary approach, fundamentally relying on the principle of learning from examples.

1.3.3 Knowledge Based Diagnosis

A Knowledge-Based System (KBS), often referred to as an Expert System (ES) in the diagnostic context, is a computer program designed within the field of Artificial Intelligence to emulate the reasoning and problem-solving capabilities of a human expert within a specific, typically narrow, domain [79]. Applied to the diagnosis of systems in automation and electrical engineering, its primary goal is to infer the most plausible fault(s) based on observed system symptoms.

The core of a KBS/ES lies in its explicitly represented domain knowledge. This knowledge base typically encompasses factual information about the system, heuristic rules of thumb derived from expert experience, procedural logic for diagnostic steps, and potentially deeper causal relationships pertaining to system operation and failure modes. An inference engine utilizes this knowledge base, applying symbolic reasoning techniques to the input symptoms (measurements, alarms, qualitative observations) to reach a diagnostic conclusion [78].

KBS/ES provides a powerful framework particularly well-suited for complex diagnostic logic where analytical models are insufficient or intractable, often due to system non-linearities, and where human expertise provides significant diagnostic value. Diagnosing faults in non-linear systems (e.g., power electronic circuits, robotic systems, complex process plants, power grids) presents unique challenges stemming from their intricate dynamics and often non-intuitive fault propagation paths [76]. While traditional model-based methods may falter due to modelling difficulties and purely data-driven approaches might struggle with interpretability or data requirements, KBS/ES offers a complementary or alternative paradigm focused on capturing, representing, and operationalizing the valuable knowledge possessed by human diagnostic experts.

The *Figure 1.13* bellow represents the general workflow for an expert system [79].

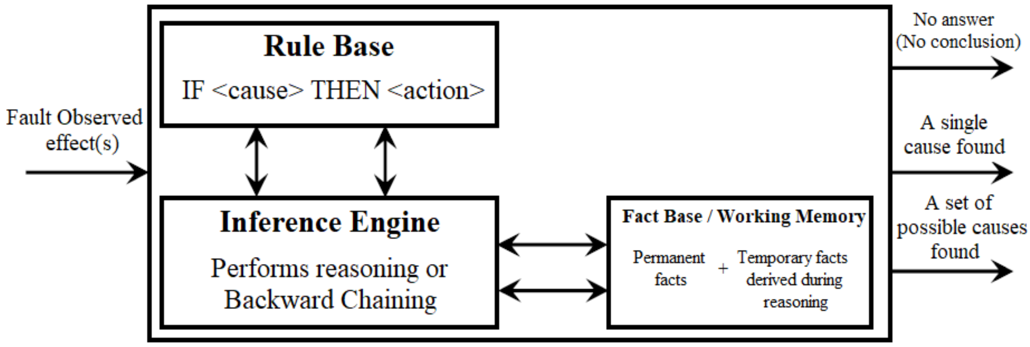

Figure 1.13: General Workflow of an expert system

1.3.4 Advantages and Challenges of AI Diagnosis Methods

Artificial Intelligence (AI) techniques, including Fuzzy Logic (FL), Neural Networks (NNs), and Knowledge-Based Systems (KBS), offer powerful alternatives and complementary approaches to conventional methods for diagnosing faults in nonlinear systems. Their ability to handle complexity, learn from data, and incorporate heuristic knowledge presents significant opportunities. However, the application of these methods also entails specific challenges that must be carefully considered.

1.3.4.1 Advantages of AI Diagnosis Methods

-*Handling Complex Nonlinearities:* AI methods, particularly NNs and fuzzy systems, are inherently capable of approximating highly complex, nonlinear input-output relationships without requiring explicit mathematical models or linearization [72]. This makes them well-suited for systems where deriving accurate physics-based models is intractable.

-*Data-Driven Capability:* Many AI techniques can learn diagnostic models directly from historical or simulated system data (inputs, outputs, fault labels). This reduces the dependency on precise analytical models, which are often difficult or impossible to obtain for complex industrial nonlinear systems [74].

-*Learning and Adaptation:* AI models, especially NNs, can be trained and potentially retrained or adapted online as new data becomes available [73]. This allows the diagnostic system to adapt to changing system dynamics, varying operating conditions, or even novel fault types not seen during initial training.

-*Robustness to Noise and Uncertainty:* Techniques like fuzzy logic are designed to handle imprecise or vague information effectively [75]. NNs can often learn to be robust to noisy sensor measurements if trained on representative data. This tolerance is crucial for real-world applications where data is rarely perfect.

-*Implicit Feature Extraction:* Deep Learning models (a subset of NNs) can automatically learn hierarchical features relevant for fault diagnosis directly from raw sensor data, potentially reducing the need for extensive manual feature engineering often required in signal-based conventional methods [72].

1.3.4.2 Challenges of AI Diagnosis Methods

-*Data Dependency and Requirements:* The performance of data-driven AI models heavily relies on the availability, quantity, and quality of training data. Obtaining sufficient data covering diverse operating conditions and, critically, various fault types (including incipient faults) can be difficult, expensive, or even unsafe in practice. Labeled fault data is often particularly scarce.

-*Interpretability and Explainability ("Black Box" Problem):* Many AI models, especially deep NNs, function as "black boxes." Understanding why a particular diagnosis was made can be challenging. This lack of transparency can be a significant barrier in safety-critical applications where traceability and justification for diagnostic decisions are required for validation and certification [76].

-*Computational Complexity:* Training complex AI models (e.g., deep NNs) can be computationally intensive, requiring significant processing power (GPUs) and time. While inference (using the trained model) is often faster, deploying highly complex models on resource-constrained hardware for real-time diagnosis can still be challenging.

-*Generalization and Robustness:* Ensuring that an AI model generalizes well to unseen data or operating conditions slightly different from the training set is a critical challenge. Models can overfit the training data, leading to poor performance in real-world deployment. Robustness against adversarial perturbations or unexpected system changes needs careful validation [77].

-*Design and Tuning Complexity:* Selecting the appropriate AI architecture (e.g., NN layers, fuzzy rule base structure) and tuning hyperparameters (e.g., learning rate, membership functions) often requires significant expertise and extensive experimentation (trial-and-error) to achieve optimal performance.

-*Verification and Validation:* Formally verifying the correctness and guaranteeing the performance bounds of complex AI-based diagnostic systems is difficult. Establishing confidence in their reliability, especially for all possible fault scenarios and operating conditions, remains an active area of research [78].

Table 1.2: A Summary for the Advantages and challenges of AI diagnosis methods

Feature	Advantages	Challenges
Modelling	Handles complex nonlinearities; Reduced need for explicit analytical models.	Difficult to incorporate known physical constraints; "Black box" nature
Data	Learns directly from data; Can handle high-dimensional data	Requires large amounts of representative (often labelled) data; Sensitive to data quality
Knowledge	Can integrate heuristic/expert knowledge (FL, KBS)	Formal knowledge representation can be complex
Adaptation	Potential for online learning and adaptation to changing conditions	Ensuring stable adaptation; Catastrophic forgetting in NNs
Robustness	Can handle noise and uncertainty (FL, robust NNs)	Ensuring generalization to unseen data/conditions; Vulnerability to adversarial attacks
Development	Powerful feature extraction capabilities (Deep Learning)	Requires AI expertise; Complex design/tuning; High computational cost for training
Deployment	Fast inference possible for trained models	Real-time constraints for complex models; Interpretability/Explainability issues
Validation	Data-driven validation possible	Formal verification and performance guarantees are difficult to establish

1.4 Comparative Analysis: Conventional vs. AI Techniques for Diagnosis

Having reviewed both conventional (model-based and signal-based) and Artificial Intelligence (AI) approaches for diagnosing faults in nonlinear systems, this section provides a comparative analysis. The goal is to highlight the relative strengths, weaknesses, and applicability domains of each category, guiding the selection of appropriate diagnostic strategies based on system characteristics, available resources, and performance requirements.

The choice between conventional and AI methods is often not straightforward and depends heavily on the specific challenges posed by the nonlinear system under consideration. Key aspects for comparison include modelling requirements, data dependency, handling of complexity and uncertainty, interpretability, and development effort.

a- Modelling Approach & Effort:

-**Conventional (Model-Based)**: Require an explicit mathematical model (often physics-based) of the system's nominal behaviour. Developing accurate models for complex nonlinear systems can be highly challenging, time-consuming, and may necessitate significant domain expertise [43]. Linearization or simplification techniques are often employed, potentially limiting accuracy.

-**Conventional (Signal-Based)**: Do not require an explicit process model but rely on identifying characteristic features or statistical properties within measured signals. Require expertise in signal processing and understanding how faults manifest in signal patterns. May struggle if fault signatures are subtle or masked by nonlinear dynamics and noise [67].
-*AI*: Often learn system behaviour implicitly from data, reducing the need for precise analytical models. This is advantageous for highly complex or poorly understood nonlinear systems. However, the *data acquisition and preparation* effort can be substantial, and designing/training the AI model itself requires specific expertise [72].

b- Data Requirements:
-**Conventional (Model-Based)**: Primarily need data for model validation and potentially parameter tuning. Performance relies heavily on model accuracy rather than large datasets of operational behaviour, though data under faulty conditions is needed to test residual sensitivity [44].
-**Conventional (Signal-Based)**: Require sufficient representative data from both healthy and faulty operation to extract distinguishing features and set thresholds or train statistical classifiers [68].
-**AI**: Generally data-hungry, especially deep learning methods. Require large, diverse datasets covering various operating conditions and, critically, different fault types and severities. Performance is highly dependent on the quantity and quality of labelled training data, which can be scarce for fault conditions [73].

c- Handling Nonlinearity & Complexity:
-**Conventional (*Model-Based*)**: Explicitly addressing strong nonlinearities can be difficult. Methods like EKF involve linearization, while SMOs require specific system structures. Accuracy can degrade significantly if the model or its linearization assumptions are poor [45].
-**Conventional (*Signal-Based*)**: Nonlinearities can complicate signal feature extraction and interpretation, as they can distort signals in complex ways that vary with operating points.
-*AI*: Methods like NNs and Fuzzy Logic are inherently well-suited to approximating complex nonlinear functions and capturing intricate relationships in high-dimensional data, making them powerful tools for systems where conventional modelling is intractable [69, 74].

d- Interpretability & Explainability:
-**Conventional (*Model-Based*)**: Generally offer higher interpretability. Residuals or parameter estimates can often be linked back to physical phenomena or specific system components, aiding understanding of the fault's nature and location [46].
-**Conventional (*Signal-Based*)**: Interpretability depends on the chosen features (e.g., specific frequency peaks, statistical moments) [67], which can often be related to known physical effects.
-*AI*: Often suffer from the "black box" problem, particularly complex NNs. Understanding why an AI model reached a specific diagnostic conclusion can be challenging, hindering trust and validation, especially in safety-critical applications. Fuzzy logic systems can offer higher

interpretability if rules are human-defined [75].

e- Adaptability & Learning:

-*Conventional*: Typically static once designed. Adapting to changing system dynamics or new fault types usually requires manual redesign or retuning of the model or signal processing algorithms.

-*AI*: Possess inherent learning capabilities. They can potentially be retrained or adapted online/offline as new data becomes available, allowing them to adjust to system degradation or novel operating conditions/faults (though this introduces its own challenges like catastrophic forgetting) [47, 76].

f- Robustness to Noise & Uncertainty:

-**Conventional (*Model-Based*)**: Performance depends on model accuracy and explicit handling of noise (e.g., *Kalman filters*). Unmodeled dynamics or parameter uncertainties can degrade performance. SMOs offer good robustness properties under certain conditions.

-**Conventional (*Signal-Based*)**: Effectiveness depends on the robustness of feature extraction and classification algorithms to noise. Pre-processing steps are often crucial.

-*AI*: Can learn to be robust to noise if trained on noisy data. Fuzzy logic is explicitly designed to handle imprecise information. However, robustness to *unforeseen* variations or adversarial inputs can be a concern [49, 78].

1.5 Conclusion

This chapter has established the theoretical groundwork for the thesis by conducting a systematic survey of diagnostic methodologies applicable to nonlinear systems. The investigation underscored a fundamental technological trade-off: while conventional model-based techniques offer physical interpretability and mathematical rigor, they remain critically sensitive to modeling uncertainties. Conversely, data-driven AI paradigms demonstrate superior robustness in handling complex nonlinear dynamics but often suffer from a lack of formal transparency and a heavy reliance on training data. The comparative analysis suggests that no single approach is universally superior; rather, robust diagnosis in complex engineering environments often necessitates a hybrid strategy that synergizes the strengths of both verifiable control theory and adaptive artificial intelligence.

However, fault diagnosis is merely the prerequisite for system survival, not the solution. The ultimate objective is to utilize this diagnostic information to actively compensate for failures and maintain operational stability. Consequently, having defined the principles of fault detection, the thesis naturally progresses to the management of these faults. *Chapter 02* builds upon this diagnostic foundation to explore the principles, architectures, and challenges of Fault-Tolerant Control (FTC) systems.

Chapter 2

Fault-Tolerant Control of Nonlinear Systems: Conventional and AI-Based Approaches

2.1 Introduction

The proliferation of complex, high-performance, and increasingly autonomous systems across critical sectors makes their safe and reliable operation a matter of paramount importance. As the failure of a single component can have cascading, catastrophic effects, designing control systems that can maintain stability and acceptable performance in the presence of faults is no longer a desirable feature but a fundamental requirement [80, 81]. This is the central objective of fault-tolerant control (FTC).

This chapter systematically investigates the field of FTC as applied to nonlinear systems, charting a path from its classical, model-based foundations to the modern, data-driven frontiers of Artificial Intelligence. The goal is to establish a comprehensive theoretical landscape, identifying the capabilities and, more importantly, the inherent limitations of established methods to construct a compelling rationale for the AI-based and hybrid approaches that form the core of this dissertation.

The discussion begins with the field's fundamental dichotomy: passive FTC (PFTC) versus active FTC (AFTC). Passive approaches use a single, fixed, robust controller that is inherently insensitive to a predefined class of faults. In stark contrast, active approaches employ a dynamic architecture where faults are explicitly detected, diagnosed, and then accommodated through a real-time reconfiguration of the control strategy, highlighting the trade-off between guaranteed reliability and high-performance potential.

Building on this, the chapter first reviews conventional FTC Methodologies, covering prominent PFTC techniques (H-∞ control, SMC) and active strategies (observer-based diagnosis, control reconfiguration). This review culminates in a critical synthesis of their overarching limitations, focusing on the challenges posed by model-dependency, strong nonlinearities, and inflexibility to novel fault scenarios. The analysis further highlights the necessity of HFTC Approaches, which strategically integrate the robustness of passive components with the adaptability of active reconfiguration, serving as a critical bridge between the two paradigms.

These limitations serve as the direct motivation for exploring AI-based FTC methods, which introduce a paradigm shift from analytical, equation-based design to empirical, learning-based synthesis. We will investigate three principal AI methodologies: Fuzzy Logic Controllers, which leverage heuristic expert knowledge to manage uncertainty; and Neural Network-based FTC, highlighting their power as universal function approximators capable of adapting online, including through methods like Reinforcement Learning, where an agent autonomously discovers an optimal fault-tolerant policy through goal-directed trial-and-error.

Ultimately, this chapter serves a dual purpose: to provide a critical literature review of the FTC field while simultaneously constructing a narrative that justifies the research direction of this thesis. By juxtaposing the verifiable but brittle nature of conventional methods with the adaptive but opaque character of AI-based systems, we will delineate the research gap this work aims to address: harnessing the power of intelligent systems to create a new generation of fault-tolerant controllers that are not only high-performing and adaptive but also reliable and trustworthy.

2.1.1 Definition and Objectives

The increasing complexity, autonomy, and stringent performance requirements of modern engineering systems, particularly in safety-critical applications such as aerospace, automotive systems, chemical processes, and power grids, necessitate operation beyond traditional robustness paradigms. These systems are inevitably susceptible to various types of faults—unexpected and undesirable deviations in system parameters or structure—originating from component degradation, malfunction, or complete failure [80]. Fault-Tolerant Control (FTC) emerges as a critical field dedicated to ensuring the continued, reliable, and safe operation of dynamic systems in the presence of such faults.

Formally, *Fault-Tolerant Control* refers to the inherent capability of a control system to maintain stability and acceptable levels of performance, thereby fulfilling predefined objectives, despite the occurrence of faults within its components (e.g., sensors, actuators, plant dynamics, and communication links). Unlike conventional control strategies primarily designed assuming fault-free operation, or standard robust control methods focused on handling model uncertainties and external disturbances within predefined bounds, FTC specifically addresses the challenges posed by abrupt or incipient component malfunctions that can significantly alter system dynamics or functionality [81]. The core principle of FTC involves automatically detecting, identifying (diagnosing), and, crucially, accommodating or compensating for the effects of these faults to preserve system integrity. Achieving this often involves mechanisms for fault detection and diagnosis (FDD) coupled with control law adaptation or reconfiguration [82].

The implementation of FTC is driven by a set of critical objectives aimed at ensuring the overall integrity and mission success of the controlled system. These objectives can often be viewed hierarchically:

1. **Safety Assurance**: As the highest priority, FTC prevents faults from leading to hazardous situations, equipment damage, or harm, ensuring the system remains within safe operating limits, even if functionality is reduced [80].

2. **Stability Preservation**: It is essential to maintain system stability after a fault to avoid catastrophic failure.

3. **Performance Maintenance/Recovery**: FTC aims to keep the system operating near its nominal performance. If not possible, it seeks "graceful degradation," where performance declines predictably rather than collapsing suddenly.

4. **Reliability and Availability Enhancement**: By allowing continued operation despite faults, FTC increases system uptime and dependability, which is vital for minimizing economic losses and for autonomous systems where manual intervention is difficult [81,82].

Achieving these objectives leads to tangible benefits like minimizing unscheduled downtime and maintenance costs, and potentially extending the system's operational life. However, designing effective FTC systems for complex nonlinear systems presents significant challenges. In essence, FTC provides a crucial layer of resilience, allowing systems to withstand component failures while meeting essential safety and performance goals [81].

2.1.2 Classification: Passive FTC vs. Active FTC

Fault-Tolerant Control (FTC) methodologies are fundamentally categorized into two principal classes, distinguished by their architectural response to fault occurrences: Passive Fault-Tolerant Control (PFTC) and Active Fault-Tolerant Control (AFTC). The core distinction lies in whether the control system possesses a fixed, predetermined structure designed to inherently withstand certain faults, or if it incorporates mechanisms to dynamically adapt its behaviour upon detecting a fault [83]. This classification provides a foundational framework for understanding the diverse strategies employed in FTC design, *Figure 2.1* below, represents the FTC approaches classification.

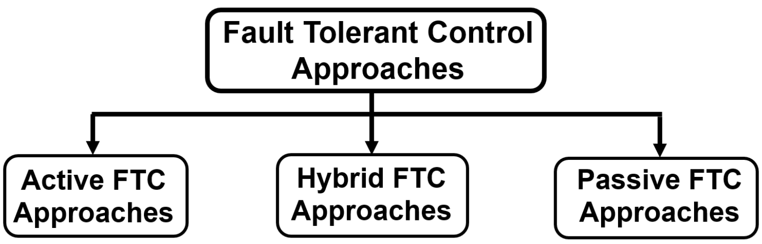

Figure 2.1: Classification of fault tolerant control approaches

2.1.2.1 Passive Fault-Tolerant Control (PFTC)

Passive FTC, often referred to as robust FTC, relies on controllers synthesized a priori with a fixed structure and parameters. The design philosophy centres on embedding sufficient robustness within the nominal control law to ensure stability and maintain an acceptable level of performance across a predefined range of anticipated fault scenarios, without requiring explicit online fault detection or controller modification [84].

-*Operating Principle*: Potential faults are treated as uncertainties or disturbances during the design phase. The controller is inherently designed to mitigate these specific faults through its fixed feedback mechanisms, with no adaptation based on real-time fault information.

-*Key Characteristic*: The control law remains static and invariant throughout operation, regardless of whether a fault within its tolerance range occurs.

-*General Approach*: This method heavily relies on robust control theories, such as H-infinity control and Sliding Mode Control, to guarantee performance and stability despite uncertainties, including faults, will be explored in *Subsection 2.2.1*.

-*Inherent Trade-offs*: While PFTC is simple to implement (avoiding separate fault detection and reconfiguration logic) and reacts instantly, it often leads to conservative control designs. The required robustness can compromise optimal performance under normal, fault-free conditions. Furthermore, its effectiveness is limited to the specific faults considered during the design stage and it cannot handle unforeseen or severe faults that exceed its pre-designed margins [84–86].

2.1.2.2 Active Fault-Tolerant Control (AFTC)

In contrast to the passive approach, Active FTC systems are characterized by their ability to explicitly detect, diagnose, and dynamically respond to faults during operation. AFTC architectures typically comprise two essential components operating in synergy: a Fault Detection and Diagnosis (FDD) subsystem and a reconfigurable or adaptive controller. The FDD module continuously monitors the system, identifies the occurrence of a fault, and provides diagnostic information (e.g., fault type, location, magnitude, time profile) to the control system. Based on this information, the control law is then adjusted, retuned, or entirely reconfigured online to actively counteract the detrimental effects of the detected fault [87].

Operating Principle: AFTC follows a detect-diagnose-compensate paradigm. Faults are first identified by the FDD unit, and this diagnostic information subsequently triggers a modification in the control strategy to accommodate the specific fault scenario.

Key Characteristic: The control system's structure or parameters are dynamically altered online in direct response to fault detection and diagnosis information [88].

General Approach: AFTC encompasses a variety of techniques, including explicit control reconfiguration strategies, adaptive control schemes tailored for fault compensation, and observer-based methods where fault estimates are actively used in the control loop. These diverse active strategies will be examined in detail in *Subsection 2.2.2*.

Inherent Trade-offs: AFTC can achieve better performance after a fault compared to PFTC because the control action is specifically tailored to the diagnosed fault, potentially allowing the system to recover closer to its normal operation. It can also handle a wider range of faults, including unforeseen ones, provided the FDD and reconfiguration capabilities are versatile enough. However, this comes at the cost of increased system complexity and the need for a reliable and timely FDD subsystem. Potential challenges include detection delays, diagnostic inaccuracies, and ensuring stability during control transitions or adaptations [89].

2.1.2.3 Summary Comparison

The fundamental differences between the passive and active FTC paradigms are summarized in *Table 2.1* Below:

Table 2.1: *Comparison of Passive and Active Fault-Tolerant Control Approaches.*

Feature	Passive FTC (PFTC)	Active FTC (AFTC)
Core Principle	Inherent robustness via fixed design	Adaptation/reconfiguration via FDD
Controller Structure	Fixed, determined a priori	Dynamic, changes online post-fault
Response Mechanism	Pre-designed tolerance	Online detection, diagnosis, compensation
FDD Requirement	Not required for control operation	Essential component
Fault Scope	Predefined, anticipated faults	Potentially wider range, incl. unforeseen
Nominal Performance	Often conservative (compromised)	Potentially closer to optimal
Post-Fault Performance	Limited by fixed design (degradation)	Higher potential for recovery
Implementation Complexity	Lower (controller only)	Higher (Controller + FDD + Logic)
Initial Reaction Speed	Potentially faster (no detection delay)	Delayed by detection/diagnosis
Handling Unforeseen Faults	Generally limited	Potentially better (depends on FDD/logic)

The distinction between passive and active FTC represents a fundamental dichotomy in fault management strategies. PFTC emphasizes inherent, pre-designed robustness with a static controller, whereas AFTC focuses on adaptive resilience achieved through online fault diagnosis and dynamic control modification. As highlighted in *Table 2.1*, each approach presents distinct advantages and disadvantages. The selection between these paradigms, or potentially hybrid approaches merging their characteristics (as alluded to in *Subsection 2.2.3*), is contingent upon application-specific factors such as the criticality of performance degradation, the types and likelihood of faults, system complexity, and the feasibility of implementing a reliable FDD component [84,89]. Recognizing this classification is essential for navigating the diverse landscape of FTC techniques, particularly when addressing the complexities inherent in nonlinear systems, which are the focus of the subsequent sections.

2.2 Conventional FTC Methods for Nonlinear Systems

This section examines the traditional, model-based strategies that historically formed the foundation of Fault-Tolerant Control (FTC) for nonlinear systems, prior to the widespread use of artificial intelligence. These methods primarily relied on mathematical models to ensure system stability and performance during faults and are divided into two distinct approaches: passive and active FTC.

Passive strategies focus on designing a single, robust controller that can tolerate a predefined set of faults, essentially treating them as model uncertainties. In contrast, active strategies utilize a dedicated fault detection and diagnosis (FDD) module to explicitly identify the fault, which then triggers a reconfiguration of the control law to accommodate it [85].

The key methodologies within both paradigms will be explored, including robust techniques like H-infinity and Sliding Mode Control (SMC) for passive FTC, and reconfiguration, adaptive control, and observer-based methods for active FTC. The review will also cover hybrid approaches that combine elements of both. This analysis will conclude by highlighting the inherent limitations of these conventional methods when dealing with complex nonlinearities and uncertainties, thereby establishing the need for the more advanced, data-driven techniques discussed in the following section.

2.2.1 Passive FTC Approaches

Passive Fault-Tolerant Control (PFTC) represents a paradigm wherein the control system is designed a priori to possess inherent robustness against a predefined set of potential faults. Unlike active approaches, PFTC systems do not require a separate fault diagnosis module or controller reconfiguration mechanism. Instead, they employ a single, fixed-parameter controller designed to maintain system stability and acceptable performance degradation across both fault-free and faulty operating conditions. The fundamental principle of PFTC is to treat the effects of faults—such as changes in system dynamics or loss of actuator effectiveness—as parametric uncertainties or external disturbances. By designing a controller that is inherently insensitive to these uncertainties, fault tolerance is achieved implicitly [84, 85].

This approach heavily relies on the principles of robust control theory, which provides a systematic framework for designing controllers that can handle significant model uncertainty and external disturbances. The primary advantage of PFTC is its simplicity and high reliability for anticipated faults, as it avoids the complexities and potential time delays associated with fault detection and control reconfiguration. However, this approach can be conservative, as the controller must be robust enough to handle the worst-case fault scenario, which may lead to compromised nominal performance when the system is healthy [86], *figure 2.2* below represent the Diagram of a Passive FTC approach.

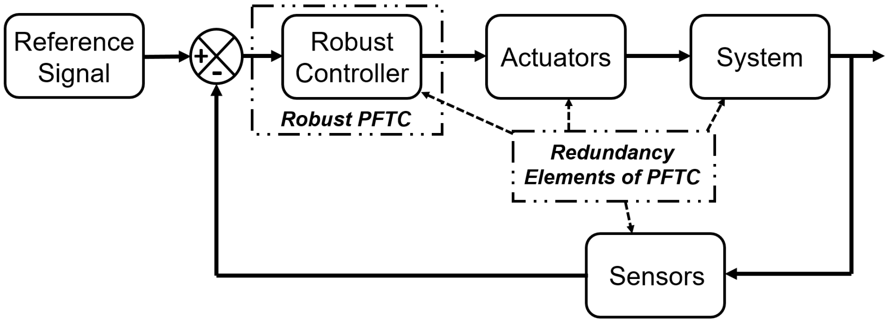

Figure 2.2: Diagram of PFTC

The following subsections delve into three prominent robust control methodologies that are widely employed for the design of passive fault-tolerant systems for nonlinear applications.

2.2.1.1 H-infinity Control

H-infinity (H_∞) control is a cornerstone of modern robust control theory, offering a powerful method for designing passive fault-tolerant control systems (PFTC). Its goal is to synthesize a controller that minimizes the worst-case effect of external inputs, including faults which are modelled as energy-bounded disturbances. The H_∞ framework provides a rigorous mathematical structure to design a fixed-gain controller that guarantees closed-loop stability and attenuates the impact of fault signals to a prespecified level [90].

A. Foundations: The Linear H_∞ Control Problem: To understand its application to nonlinear systems, it is essential first to grasp the standard H_∞ control problem in the linear time-invariant (LTI) domain. The problem is formulated using a generalized plant structure [91], as shown in *Figure 2.3*, which encapsulates the system dynamics, performance objectives, and interconnections.

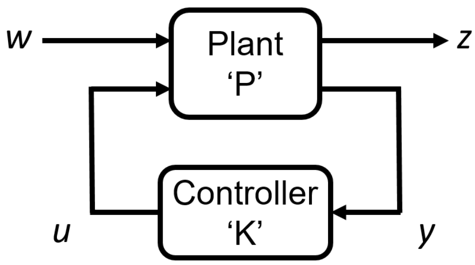

Figure 2.3: Blok Diagram of H infinity control

-*Exogenous Inputs (w):* All external signals not manipulated by the controller, such as disturbances, sensor noise, reference commands, and, crucially for PFTC, fault signals.

-*Control Inputs (u):* The signals generated by the controller 'K'.

-*Performance Outputs (z):* These are the signals that need to be minimized or regulated, typically representing tracking errors or weighted control effort.

-*Measured Outputs (y):* The signals available to the controller for feedback, typically from sensors [92].

The generalized plant 'P' can be described by a state-space representation [93]:

$$\begin{cases} \dot{x}(t) &= Ax(t) + B_1 w(t) + B_2 u(t) \\ z(t) &= C_1 x(t) + D_{11} w(t) + D_{12} u(t) \\ y(t) &= C_2 x(t) + D_{21} w(t) + D_{22} u(t) \end{cases} \tag{2.1}$$

with:

- $x \in (n * 1)$: The State Vector,
- $w \in (m_1 * 1)$: The Exogenous Input Vector,
- $z \in (p_1 * 1)$: The Performance Output (or Controlled Output) Vector,
- $A \in (n * n)$: State matrix (system dynamics),
- $B_1 \in (n * m_1)$: Disturbance input matrix,
- $B_2 \in (n * m_2)$: Control input matrix,
- $C_1 \in (p_1 * n)$: Performance output matrix,
- $C_2 \in (p_2 * n)$: Measurement output matrix,
- $D_{11} \in (p_1 * m_1)$: Feedthrough from disturbance to performance output,
- $D_{12} \in (p_1 * m_2)$: Feedthrough from control input to performance output (often represents control effort weighting),
- $D_{21} \in (p_2 * m_1)$: Feedthrough from disturbance to measurement (typically represents sensor noise),
- $D_{22} \in (p_2 * m_2)$: Feedthrough from control input to measurement.

The controller 'K' generates the control input '$u = Ky$' based on the measured output. When connected to the plant, this forms a closed-loop system described by the Lower Linear Fractional Transformation (LFT), denoted '$F_l(P, K)$' [90]. This results in a closed-loop transfer function from the exogenous inputs 'w' to the performance outputs 'z', denoted 'T_{zw}'.

The standard H_∞ control problem is defined as, For a given disturbance attenuation level '$\gamma > 0$', find an internally stabilizing controller 'K' such that the H_∞ norm of the closed-loop transfer function 'T_{zw}' is bounded by 'γ'. Mathematically [91]:

$$\|T_{zw}(s)\|_\infty = \|F_l(P, K)\|_\infty < \gamma \tag{2.2}$$

The H_∞ norm is defined as the supremum of the largest singular value of the transfer function matrix over all frequencies 'ω' :

$$\|T_{zw}(s)\|_\infty = \sup_{\omega} \sigma_{\max}[T_{zw}(j\omega)] \tag{2.3}$$

Physically, this norm represents the maximum possible ratio of the energy of the output signal '$z(t)$' to the energy of the input signal '$w(t)$' over all possible energy-bounded input signals [92].

$$\sup_{\{w(t)\neq 0, \|w\|_2<\infty\}} \left(\frac{\|Z(t)\|_2}{\|w(t)\|_2}\right) < \gamma \tag{2.4}$$

Where '$\|.\|_2$' denotes the L2-norm (or energy) of the signal. This inequality is why H_∞ control is often called an energy-to-energy gain minimization problem. The solution is famously tied to the existence of positive semi-definite solutions to two algebraic Riccati equations (AREs) [93].

B. Extension to Nonlinear Systems and the HJI Inequality: For nonlinear systems, frequency-domain concepts like transfer functions are no longer directly applicable. The problem must be reformulated in the time domain using the concept of L2-gain [91].

$$\begin{cases} \dot{X} &= f(x) + g_1(x)w + g_2(x)u \\ Z &= h_1(x) + k_{12}(x)u \end{cases} \tag{2.5}$$

The objective is to find a state-feedback control law '$u = \alpha(x)$' such that the L2-gain of the closed-loop system from 'w' to 'z' is less than or equal to 'γ'. This problem is linked to the theory of dissipative systems and can be solved by finding a "storage function" '$V(x) \geq 0$' that satisfies the following Hamilton-Jacobi-Isaacs (HJI) partial differential inequality [90]:

$$\frac{\partial V}{\partial X} f(x) + \left(\frac{1}{2\gamma^2}\right)\left(\frac{\partial V}{\partial X}\right) g_1(x) g_1^T(x) \left(\frac{\partial V}{\partial X}\right)^T + \left(\frac{1}{2}\right) h_1^T(x) h_1(x) - \left(\frac{1}{2}\right) L_{g_2}H(x) R^{-1}(x) L_{g_2}H(x)^T \leq 0 \tag{2.6}$$

Where $L_{g_2}H(x) = \left(\frac{\partial H}{\partial x}\right) g_2(x)^T$ is the Lie derivative of a Hamiltonian-like function 'H' along the vector field '$g_2(x)$', and 'R' is a weighting matrix.

A more direct and common form of the HJI inequality expresses the differential game nature of the problem, where the controller 'u' seeks to minimize the system energy while the disturbance 'w' seeks to maximize it. For a state-feedback law '$u(x)$', the HJI inequality that must be satisfied is [92]:

$$\sup_{\{w\neq 0\}} \left\{ \frac{\partial V}{\partial X}(f(x) + g_1(x)w + g_2(x)u(x)) + \|Z(x, u(x))\|_2^2 - \gamma^2 \|w\|_2^2 \right\} \leq 0 \tag{2.7}$$

The solution for the worst-case disturbance 'w^*' and the optimal control 'u^*' can be found by taking the gradients of the Hamiltonian function associated with this inequality, yielding:

$$\begin{cases} w^* &= \left(\frac{1}{\gamma^2}\right) g_1^T(x) \left(\frac{\partial V}{\partial X}\right)^T \\ u^* &= -R^{-1}(x) g_2^T(x) \left(\frac{\partial V}{\partial X}\right)^T \end{cases} \tag{2.8}$$

Substituting these back into the HJI inequality gives the final form that must be solved for '$V(x)$' :

$$\frac{\partial V}{\partial X} f(x) + h_1^T(x)h_1(x) + \left(\frac{1}{\gamma^2}\right)\left(\frac{\partial V}{\partial X}\right) g_1(x)g_1^T(x)\left(\frac{\partial V}{\partial X}\right)^T - \left(\frac{\partial V}{\partial X}\right) g_2(x)R^{-1}(x)g_2^T(x)\left(\frac{\partial V}{\partial X}\right)^T \leq 0 \quad (2.9)$$

C. Challenges and Practical Implications: While the HJI inequality offers a complete theoretical solution, solving this partial differential inequality for a general nonlinear system is a formidable task. Analytical solutions are rare and exist only for very simple, low-dimensional systems. Higher-order systems face the "curse of dimensionality," where the computational complexity of finding a solution '$V(x)'$ grows exponentially with the number of states [93].

This intractability is the primary limitation of applying nonlinear H_∞ control in practice and motivates the exploration of alternative methods to find approximate solutions :

1. *Successive Approximation*: An iterative procedure where the nonlinear problem is successively approximated and solved.

2. *State-Dependent Riccati Equation (SDRE)*: This technique factors the nonlinear dynamics into a pseudo-linear form, where an algebraic Riccati equation is solved at each time step.

3. *Sum-of-Squares (SOS) Programming*: If the system dynamics and storage function can be represented as polynomials, the HJI inequality can be converted into a convex semidefinite program (SDP) that can be solved efficiently [90–93].

In the context of this thesis, the inherent difficulty in deriving an exact, global solution for the H_∞ controller for complex nonlinear systems is a critical point. Although effective, these controllers can be conservative as they are designed for the absolute worst-case disturbance, a condition that may never occur, potentially leading to sluggish or suboptimal performance under normal, fault-free conditions. This trade-off between robustness and nominal performance, coupled with the profound difficulty of solving the HJI inequality, creates a compelling case for investigating intelligent, adaptive, and learning-based FTC strategies, which are the subject of the subsequent chapter.

2.2.1.2 Sliding Mode Control (SMC)

Sliding Mode Control (SMC) is a powerful nonlinear control strategy known for its exceptional robustness, finite-time convergence, and inherent invariance to a specific class of uncertainties and disturbances, making it a natural paradigm for Passive Fault-Tolerant Control (PFTC). The core philosophy of SMC is not to cancel disturbances, but to alter the system's dynamics to become completely insensitive to their effects [94]. This is achieved by using a discontinuous,

high-speed switching control law to force the system's state trajectories onto a user-defined manifold called the sliding surface and maintain them there.
Figure 2.4 below represent a block diagram for sliding mode control approach.

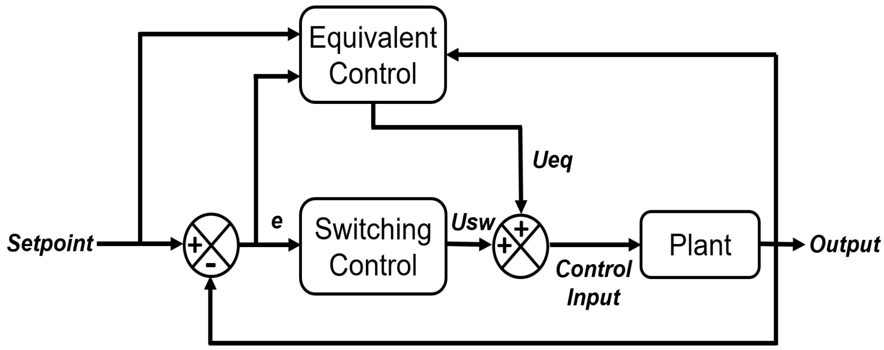

Figure 2.4: Blok Diagram of Sliding Mode control

The design of an SMC system is divided into two phases:

1. *Design of a Sliding Surface (σ)*: Defining a manifold in the state space such that when the system trajectory is confined to it, the system exhibits desired behaviour (e.g., stability, tracking).

2. *Design of a Control Law (u):* Synthesizing a control law that guarantees the state trajectory will reach the sliding surface in finite time (the "reaching phase") and remain on it thereafter (the "sliding phase"), despite matched uncertainties and faults [95].

A. Phase 1: Defining the Sliding Surface: The sliding surface is an algebraic constraint on the system's states, defined by '$\sigma(x) = 0$', where 'x' is the state vector. Its selection is the most critical design step as it dictates the controlled system's behaviour. For a tracking problem, where the goal is to make the system state 'x' follow a desired trajectory x_d, the error vector is defined as [96]:

$$e(t) = x(t) - x_d(t) \tag{2.10}$$

A common choice for the sliding variable σ is a weighted linear combination of the error and its derivatives. For an n-th order system, this can be expressed as :

$$\sigma(e,t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} e(t) \tag{2.11}$$

Where ($\lambda > 0$) is a user-defined positive constant. For a second-order system, this simplifies to:

$$\sigma = \dot{e} + \lambda e \tag{2.12}$$

Once the system is forced onto the surface ($\sigma = 0$), the error dynamics are governed by the stable, first-order linear differential equation:

$$\dot{e} = -\lambda e \tag{2.13}$$

The solution $e(t) = e(0)e^{-\lambda t}$ guarantees that the tracking error converges exponentially to zero. The dynamics during the sliding phase are of a lower order and are completely independent of the original system dynamics and any matched disturbances, which is the source of SMC's remarkable robustness [97].

The *Figure 2.5* below represent the Phase Plane Portrait of a Sliding Mode Control (SMC) system.

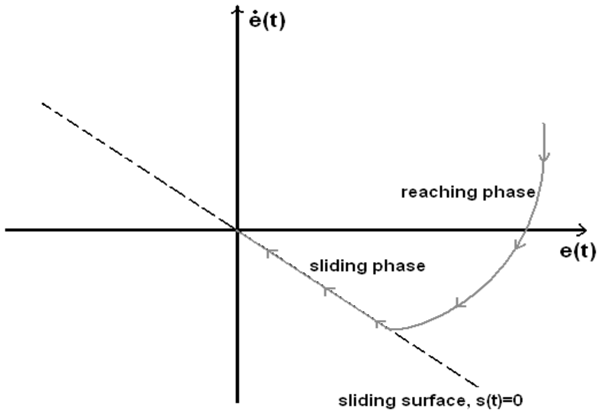

Figure 2.5: Phase Plane Portrait of Sliding Mode Control

B. Phase 2: The Control Law and Reaching Condition: The control law 'u' must be designed to drive the state trajectory to the surface ($\sigma = 0$) and keep it there. A sufficient condition to guarantee convergence in finite time is the "reaching condition," often based on a Lyapunov stability argument [96]:

$$\sigma \frac{d\sigma}{dt} \leq -\eta|\sigma| \tag{2.14}$$

Where 'η' is a strictly positive constant.

To derive the control law, consider a second-order affine nonlinear system with uncertainties and faults [95]:

$$\ddot{X} = f(x, \dot{x}) + b(x, \dot{x})u + d(t) \tag{2.15}$$

Here, $f(x, \dot{x})$ and $b(x, \dot{x})$ are the nominal system dynamics, and '$d(t)$' is a lumped term representing all uncertainties, disturbances, and faults, assumed to be bounded by a known constant: $|d(t)| \leq D_{\max}$.

The SMC law is typically composed of two parts :

$$u = u_{eq} + u_{sw} \tag{2.16}$$

The *equivalent control*, u_{eq}, is a continuous term representing the ideal control effort to keep the system on the sliding surface if there were no uncertainties ($d(t) = 0$) [94]. It is found by setting $d\sigma/dt = 0$.

$$u_{eq} = b^{-1}(x, \dot{x}) \cdot (-f(x, \dot{x}) + \ddot{x}_d - \lambda\dot{e}) \tag{2.17}$$

The *switching control*, u_{sw}, is a discontinuous term that drives the states to the surface and counteracts the uncertainties $d(t)$. A standard choice is a signum function :

$$u_{sw} = -b^{-1}(x, \dot{x}) \cdot k \cdot \mathrm{sgn}(\sigma) \tag{2.18}$$

Where 'k' is the switching gain. Combining these gives the full SMC law :

$$u = b^{-1}(x, \dot{x}) \cdot (-f(x, \dot{x}) + \ddot{x}_d - \lambda \dot{e} - k \cdot \mathrm{sgn}(\sigma)) \tag{2.19}$$

To satisfy the reaching condition, the gain 'k' must be chosen to be larger than the maximum possible magnitude of the lumped uncertainty: $k > D_{\max}$

C. SMC for Passive Fault Tolerance and the Chattering Problem: SMC provides passive fault tolerance by modelling faults as bounded contributions to the lumped uncertainty term $d(t)$. The fixed-gain switching controller ensures stability and performance without needing to detect or identify the fault [95]. However, this powerful invariance property applies primarily to matched uncertainties—those that enter the system dynamics through the same channel as the control input.

The most notorious practical challenge in SMC is the *chattering phenomenon*. The ideal, discontinuous '$\mathrm{sgn}(\sigma)$' function requires infinite-frequency switching to keep the state on the ($\sigma = 0$) manifold. In any real system, delays and inertia cause high-frequency oscillations around the sliding surface. Chattering is highly undesirable as it can [96]:

- Excite unmodeled high-frequency dynamics.
- Lead to excessive wear and damage to actuators.
- Result in low control accuracy and high energy consumption.

To mitigate chattering, the standard approach is to smooth out the control discontinuity within a thin boundary layer of thickness Φ around the sliding surface by replacing the $\mathrm{sgn}(\sigma)$ function with a continuous approximation, such as the saturation function $\mathrm{sat}(\sigma/\Phi)$ [97].

$$\mathrm{sat}\left(\frac{\sigma}{\Phi}\right) = \begin{cases} \sigma/\Phi & \text{if } |\sigma| \leq \Phi \\ \mathrm{sgn}(\sigma) & \text{if } |\sigma| > \Phi \end{cases} \tag{2.20}$$

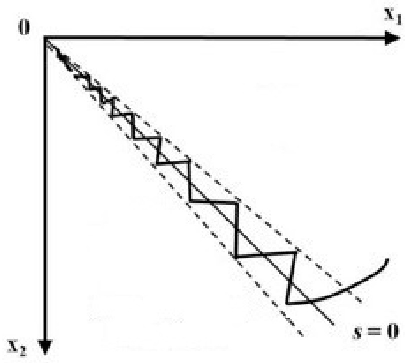

Figure 2.6: Mitigation of Chattering using a Boundary Layer

This modification comes at the cost of performance. Inside the boundary layer, the perfect invariance to matched disturbances is sacrificed, and the system is only guaranteed to remain within a small neighbourhood of the ideal surface. This introduces a trade-off between control action robustness and tracking precision. More advanced techniques, such as Higher-Order Sliding Mode Control (HOSM), have been developed to eliminate chattering while preserving the theoretical precision of the original SMC, albeit at the cost of increased design complexity. The persistence of the chattering trade-off and the limitation to matched uncertainties are primary motivators for exploring alternative FTC schemes [94–97].

2.2.2 Active FTC Approaches

In stark contrast to the static, robust-by-design philosophy of passive fault tolerance, Active Fault-Tolerant Control (AFTC) embodies a dynamic and adaptive paradigm. AFTC systems explicitly react to the occurrence of a fault by reconfiguring the control strategy in real-time. This is accomplished through a sophisticated architecture that integrates a control loop with a dedicated Fault Detection and Diagnosis (FDD) module [89]. The fundamental premise of AFTC is to first obtain explicit information about the fault's existence, location, and severity, and then use this information to make intelligent adjustments to the control law [86].

The operation of an AFTC system follows a distinct sequence of events, as illustrated in the general architecture shown in *Figure 2.7*.

1. ***Fault Occurrence***: A component or actuator deviates from its nominal behaviour.

2. ***Fault Detection***: The FDD module, continuously monitoring system signals, detects an anomaly or inconsistency between the expected and measured system behaviour.

3. ***Fault Diagnosis (Isolation and Identification)***: The FDD module analyses the fault symptoms (residuals) to isolate its location (e.g., which sensor or actuator has failed) and identify its characteristics (e.g., type of fault, magnitude, and time profile).

4. ***Control Reconfiguration***: Based on the diagnostic information provided by the FDD, the controller supervision block initiates a change in the control law to compensate for the fault's effect, aiming to restore system stability and recover performance as much as possible [87].

Figure 2.7 below, represent the block diagram of an active FTC approach [86].

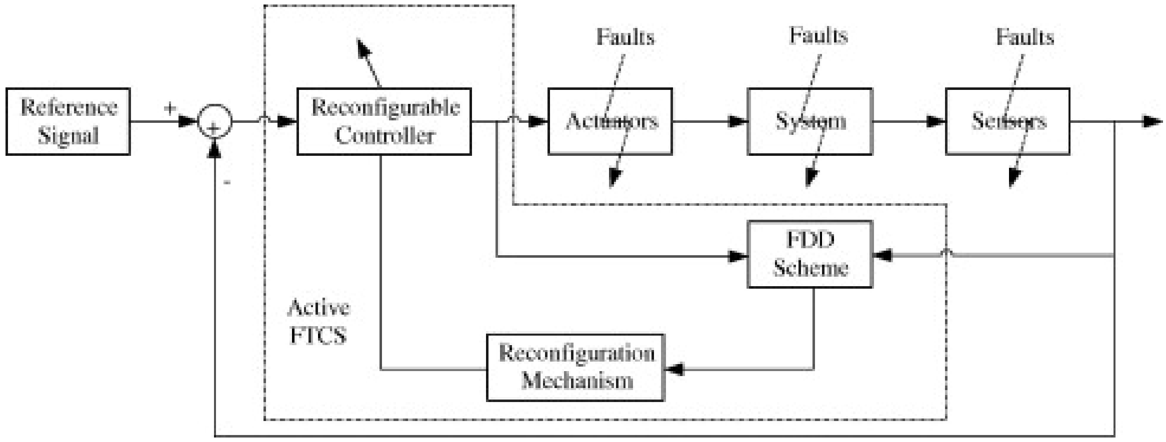

Figure 2.7: Diagram of AFTC

The primary advantage of this active approach is superior performance under both nominal and faulty conditions. Unlike a passive controller, which is often conservatively designed for a worst-case scenario, an active controller can operate optimally when the system is healthy and can adapt to the specific nature of a fault when one occurs. However, this performance comes at the cost of increased complexity. The AFTC system's effectiveness is critically dependent on the FDD module, which introduces a potential time delay between fault occurrence and accommodation and adds another layer of complexity that itself could be a source of failure [88].

The following subsections explore the key methodologies that constitute the core of active fault-tolerant strategies.

2.2.2.1 Control Reconfiguration Strategies

Control reconfiguration is the core of an Active Fault-Tolerant Control (AFTC) system. It uses information from a Fault Detection and Diagnosis (FDD) module to adapt the control strategy after a fault is diagnosed. The primary goal is to preserve stability and recover as much performance as possible. The choice of strategy involves a trade-off between effectiveness, complexity, and computational demand, ranging from simple parameter adjustments to a complete online redesign of the controller [98].

A. Control Trimming (Parameter Retuning)

Control trimming, or gain scheduling, is the simplest strategy. It maintains the controller's structure but adjusts its key parameters (gains) online in response to fault information [99]. This approach is predicated on the assumption that faults primarily affect the system's parameters.

A.1. Conceptual Framework and Application: When a fault occurs, such as a partial loss of actuator effectiveness, the FDD system estimates the new, faulty system parameters. The reconfiguration mechanism then uses a pre-computed mapping to find a new set of controller gains suitable for the post-fault dynamics [98].

For a system described by [98]:

$$\dot{x} = f(x) + B(\alpha)u \tag{2.21}$$

With $f(x)$ representing System Dynamics Function, and a fault changes the parameter vector from 'α_0' to 'α_f'. A common example is an actuator fault, where the effective control input becomes :

$$u_{eff} = (I - \beta)u \tag{2.22}$$

Here, u_{eff} represent the effective control input signal, β represents the percentage loss of effectiveness. The FDD provides an estimate $\hat{\beta}$. If the nominal control is:

$$u = -k(\alpha_0)x \tag{2.23}$$

The post-fault system would likely be unstable. The trimming strategy uses a mapping, $K_{\text{new}} = \Phi(\beta)$, to find an appropriate new gain. The reconfigured control law is [99]:

$$u = -k_{\text{new}} * x \tag{2.24}$$

A.2. Implementation and Challenges: The mapping 'Φ' can be a :

1. *Lookup Table*: Pre-designed gains for discrete fault levels. This is fast but inflexible.
2. *Parametric Function*: A continuous function that calculates gains based on the estimated fault parameters [100].

The primary limitation of control trimming is its scope; it is only effective for faults that do not fundamentally alter the system's structure.

The *Figure 2.8* below, represent the block diagram of control trimming via gain scheduling technique [99, 100].

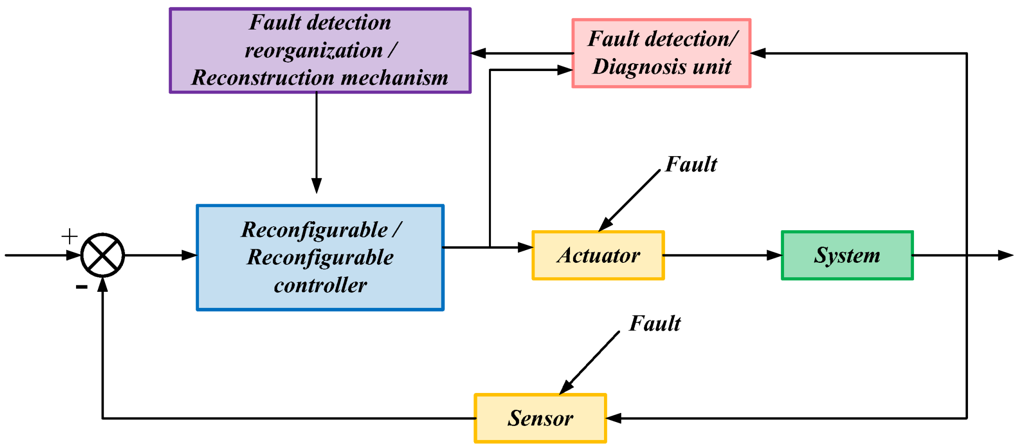

Figure 2.8: Block Diagram of Control Trimming via Gain Scheduling

2.2.2.2 Observer-Based FTC

Observer-based techniques are the bedrock of modern model-based Fault Detection and Diagnosis (FDD) and, by extension, the enabling technology for most Active Fault-Tolerant Control (AFTC) architectures. The core principle is rooted in analytical redundancy, where a mathematical model of the system (an observer or state estimator) runs in parallel to the actual plant. The observer uses the measured system inputs and outputs to generate an estimate of the system's internal states. The discrepancy between the system's actual behaviour and the observer's estimated behaviour, known as the residual, serves as a symptom of a fault [101].

The power of this approach lies in its ability to detect and diagnose faults without requiring costly physical redundancy. The entire AFTC process hinges on the ability of the observer to generate a residual signal that is maximally sensitive to faults while being minimally sensitive to noise, disturbances, and model uncertainties [101].

A. The Foundational Linear Case: The Luenberger Observer:

To understand the core mechanics, we first consider the standard linear time-invariant (LTI) case. Given a system described by the state-space model [54]:

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{cases} \tag{2.25}$$

A full-order Luenberger observer is constructed with the same structure, but with an added correction term proportional to the output estimation error [55]:

$$\begin{cases} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= C\hat{x}(t) \end{cases} \tag{2.26}$$

Where $\hat{x}(t)$ is the estimated state vector and 'L' is the observer gain matrix. The dynamics of the state estimation error, $e_x(t) = x(t) - \hat{x}(t)$, can be derived by subtracting *Eq. 2.26* from *Eq. 2.25* :

$$\dot{e}_x(t) = (A - LC)e_x(t) \tag{2.27}$$

If the pair (A, C) is observable, the observer gain 'L' can be chosen to place the eigenvalues of the matrix $(A - LC)$ at any desired location in the left-half of the complex plane, ensuring that the estimation error $e_x(t)$ converges to zero.

The residual signal $r(t)$, is defined as the output estimation error [56]:

$$r(t) = y(t) - \hat{y}(t) = Ce_x(t) \tag{2.28}$$

Under ideal, fault-free conditions, $r(t)$ converges to zero. Now, consider the presence of an actuator fault $f_a(t)$ and a sensor fault $f_s(t)$:

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) + Ef_a(t) \\ y(t) &= Cx(t) + Ff_s(t) \end{cases} \tag{2.29}$$

The residual dynamics become [57]:

$$r(t) = Ce_x(t) + Ff_s(t) \tag{2.30}$$

Where:

$$\dot{e}_x(t) = (A - LC)e_x(t) + Ef_a(t) - LFf_s(t) \tag{2.31}$$

The residual is now clearly driven by the fault signals. However, for nonlinear systems, this linear framework is wholly inadequate as the principle of superposition does not hold.

B. Observer Techniques for Nonlinear Systems:

The challenge in the nonlinear domain is that a simple error feedback with a constant gain 'L' cannot guarantee global error convergence. A variety of nonlinear observer design techniques have been developed to address this.

B.1. Extended Kalman Filter (EKF): The EKF is a direct extension of the Kalman Filter for nonlinear systems. It operates by performing a successive linearization of the nonlinear system dynamics around the current state estimate at each time step. For a system described by [58]:

$$\begin{cases} \dot{x} &= f(x,u) + w \\ y &= h(x) + v \end{cases} \tag{2.32}$$

Where 'w' and 'v' are process and measurement noise. The EKF algorithm proceeds in two steps:

1. *Predict [59]:*

$$\text{-State prediction: } \hat{X}(t) = f(\hat{x}(t-1), u(t-1)) \tag{2.33}$$

$$\text{-Covariance prediction: } P(t) = A(t)P(t-1)A^T(t) + Q \tag{2.34}$$

2. *Update [60]:*

$$\text{-Kalman Gain: } K(t) = P(t)C^T(t)\left[C(t)P(t)C^T(t) + R\right]^{-1} \tag{2.35}$$

$$\text{-State update: } \hat{X}(t) = \hat{x}(t) + K(t)\left[y(t) - h(\hat{X}(t))\right] \tag{2.36}$$

$$\text{-Covariance update: } P(t) = [I - K(t)C(t)]P(t) \tag{2.37}$$

Here, $A(t)$ and $C(t)$ are the Jacobian matrices of f and h evaluated at the current state estimate.

In FTC, the *innovation sequence [60, 61]*:

$$r(t) = y(t) - h(\hat{X}(t)) \tag{2.38}$$

Serves as a statistically meaningful residual. It is nominally a zero-mean, white-noise process. A fault will cause this innovation to acquire a non-zero mean and lose its white-noise property, which can be detected by statistical tests (e.g., a chi-squared test).

B.2. Sliding Mode Observer (SMO): Drawing on the robustness properties of Sliding Mode Control, the SMO is a powerful tool for FDD in uncertain nonlinear systems. An SMO uses a discontinuous switching function to force the observer's estimation error to zero in finite time [62].

Consider an uncertain nonlinear system [63]:

$$\begin{cases} \dot{X} &= Ax + B[u + \varphi(x,u) + f_a(t)] \\ y &= Cx \end{cases} \tag{2.39}$$

Where φ represents model uncertainties and f_a is a matched actuator fault, An SMO can be designed as :

$$\dot{\hat{X}} = A\hat{x} + B[u + \varphi(\hat{x},u)] + L(y - \hat{y}) - Bv \tag{2.40}$$

The term 'v' is a high-gain, discontinuous switching function designed to reject the uncertainties and drive the output error ($e_y = y - \hat{y}$) to zero.

The key insight for FDD is that when the sliding motion is achieved ($e_y = 0$), the *equivalent output injection signal* v_{eq}, which represents the average value of the switching term, must be equal to the uncertainties and faults it is cancelling out. By low-pass filtering the high-frequency switching signal 'v', one can reconstruct an estimate of the fault [64, 65]:

$$\hat{f}_a(t) \approx v_{eq}(t) \tag{2.41}$$

This provides not just fault detection but also a direct estimate of the fault's magnitude.

C. The FDD Process: From Residual to Reconfiguration

The observer's output is only the first step. A complete observer-based FDD system performs a sequence of tasks. *figure 2.9* bellow llustrates the Complete Observer-Based FDD Process, where the flow from raw system data through residual generation, evaluation (detection), isolation, and finally, providing fault information to the FTC supervision [66].

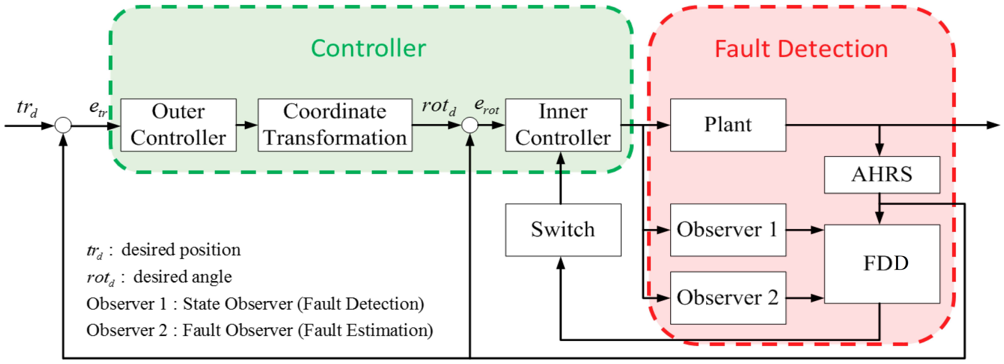

Figure 2.9: The Complete Observer-Based FDD Process.

1. *Residual Generation*: This is the role of the chosen observer (Luenberger, EKF, SMO, etc.).

2. *Residual Evaluation and Fault Detection*: The raw residual $r(t)$ is processed to make a binary decision: fault or no fault. This is typically done by comparing its norm to a predefined threshold J_{th}

$$\begin{cases} \|r(t)\| > J_{th} \implies \text{Fault Detected} \\ \|r(t)\| \leq J_{th} \implies \text{No Fault Detected} \end{cases} \tag{2.42}$$

The choice of J_{th} is a critical trade-off. A low threshold increases sensitivity to small faults but also increases the rate of *false alarms* due to noise. A high threshold reduces false alarms but may lead to *missed detections* of incipient or minor faults [51, 52].

Figure 2.10 below, represent the complete process for residual evaluation and fault detection, where The residual r(t) is small under normal conditions (due to noise). And a fault causes it to grow and cross the threshold J_{th}. The trade-off between false alarms and missed detections is determined by the threshold's level relative to the noise band [53].

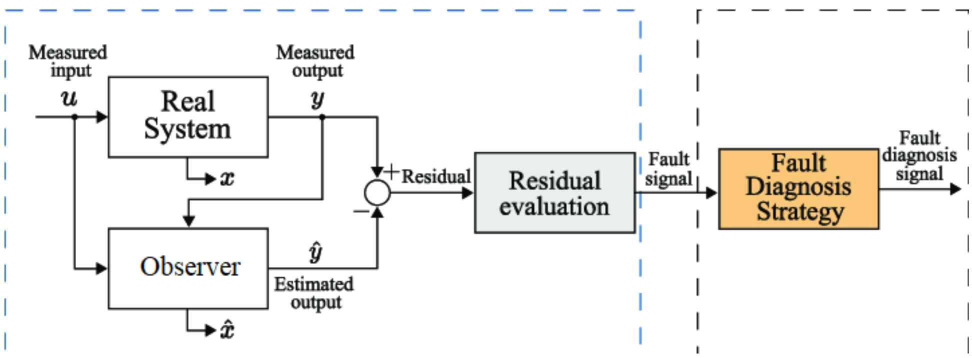

Figure 2.10: Residual Evaluation and Fault Detection Process

3. *Fault Isolation:* Once a fault is detected, the next step is to determine its location and type. This is often achieved using an observer bank. A bank of observers is designed where each observer is specifically insensitive to a particular fault. When a fault fk occurs, all

residuals will become non-zero except for the residual from observer Ok, allowing the fault to be isolated [51].

Figure 2.11 below, represent the complete process for fault isolation using an observer bank, where ach observer is designed for a different fault hypothesis. The pattern of activated residuals in the "Decision Logic" block.

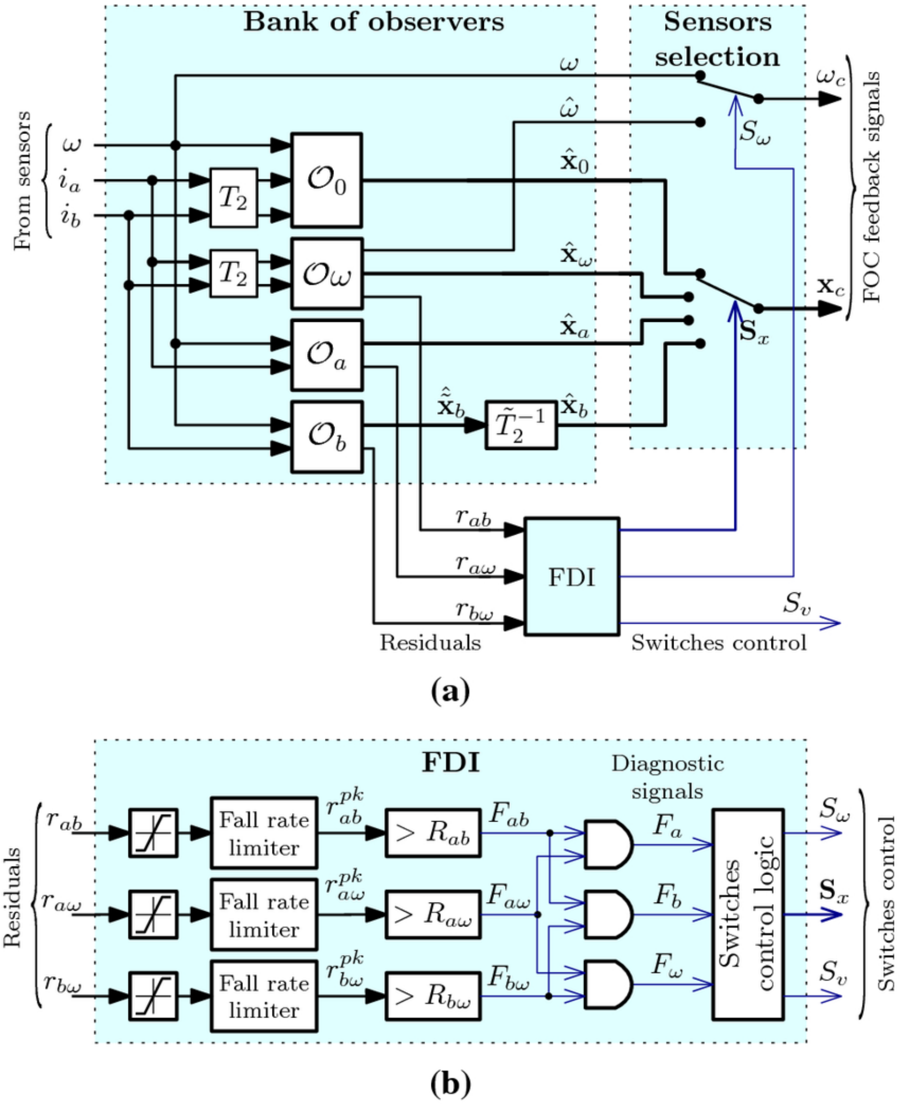

Figure 2.11: Fault Isolation using an Observer Bank

The major challenge of observer-based FTC in nonlinear systems is the difficulty of decoupling the effects of faults from the effects of model uncertainty, external disturbances, and the system's own nominal nonlinear behaviour.

2.2.3 Hybrid FTC Approaches

The distinct separation between passive and active FTC philosophies, while useful for classification, presents a practical design dilemma. Passive FTC (PFTC) offers guaranteed robustness and stability for a predefined set of faults from the moment of their occurrence, but often at the cost of conservative, degraded nominal performance. Conversely, Active FTC (AFTC) promises high performance under both nominal and faulty conditions by adapting to the specific fault, but it introduces significant complexity, potential time delays associated with diagnosis, and a critical dependence on the reliability of the Fault Detection and Diagnosis (FDD) module. Hybrid FTC approaches emerge from a pragmatic desire to mitigate these respective drawbacks by synergistically combining the two paradigms. The overarching goal is to create a control system that leverages the guaranteed stability and reliability of a passive design while harnessing the superior performance recovery capabilities of an active one [102, 103]. The specific architecture of this combination, however, can vary significantly depending on the system's performance requirements and safety criticality, leading to distinct strategic designs.

2.2.3.1 Passive Baseline with Active Compensation

This architecture is an intuitive and widely investigated approach to hybrid fault-tolerant control. The core philosophy is to prioritize safety and guaranteed stability above all else, while still striving for enhanced performance post-fault. The system is built upon a foundation of inherent robustness provided by a passive FTC controller that is *always active*. This baseline PFTC system is designed to single-handedly ensure the stability of the plant and maintain a minimum acceptable level of performance in the presence of a predefined set of faults, effectively creating a "guaranteed stability envelope" [104].

The active component functions as a secondary, intelligent layer, driven by a Fault Detection and Diagnosis (FDD) module. When the system is healthy, this component is dormant. Upon the detection and diagnosis of a fault, this component is activated. Its role is not to stabilize the system—that task is already handled by the passive baseline—but to generate a supplementary, corrective control signal to counteract the diagnosed fault's effects and recover lost performance. This elegantly overcomes the critical latency period inherent in pure AFTC systems [105].

A. Mathematical Framework and Design Principles:

Let us consider a nonlinear system subject to faults, described in an affine-control form [105]:

$$\dot{X} = f(x) + B(x)u_{eff} \tag{2.43}$$

Where u_{eff} is the effective control input. Two common fault classes are:

1. *Actuator Faults*: A loss of effectiveness in the actuators, modelled by [106]:

$$u_{eff} = (I - \beta)u \tag{2.44}$$

   Where β is a diagonal matrix representing the percentage effectiveness loss.

2. *Additive Component Faults*: These represent faults that act as external disturbances, such

as biases or failures in other components. The model becomes :

$$\dot{X} = f(x) + B(x)u + E(x)f_c(t) \tag{2.45}$$

Where $f_c(t)$ is the fault signal and $E(x)$ is its distribution matrix.

A.1. Design of the Passive Baseline Controller (upassive): The primary objective is robust stability. It is designed assuming it is the only controller acting on the plant. A robust control methodology such as H_∞ H or Sliding Mode Control (SMC) is used to synthesize upassive to guarantee that for all faults within a predefined set, the closed-loop system remains stable and key performance variables remain bounded [107].

- *If using* H_∞: The faults are modelled as a generalized disturbance vector 'w'. The controller is designed to satisfy $\|T_{zw}\|_\infty < \gamma$, ensuring the energy of the performance output is attenuated relative to the fault/disturbance energy.

- *If using SMC*: The controller is designed with a switching term large enough to overcome the effects of any *matched faults*. This can provide complete invariance to such faults but is limited by the matching condition and the *chattering phenomenon* [106, 107].

Regardless of the method, the passive controller provides a hard guarantee of stability, forming the system's safety net.

A.2. Design of the Active Compensator (u_{active})***:*** The active compensator is dormant $(u_{active} = 0)$ until the FDD module provides a reliable fault estimate. Its goal is to actively cancel the diagnosed fault's effect, thereby restoring the system's behaviour towards that of the healthy plant operating under the passive controller [105].

Let the fault estimate provided by the FDD be $\hat{\beta}$ and $\hat{f}_c$. The ideal behavior, dictated by the passive controller, would be [104]:

$$\dot{X}_{\text{ideal}} = f(x) + B(x)u_{\text{passive}} \tag{2.46}$$

The actual system dynamics with the full hybrid control law are :

$$\dot{X}_{\text{actual}} = f(x) + B(x)(I - \beta)(u_{\text{passive}} + u_{\text{active}}) + E(x)f_c \tag{2.47}$$

To achieve perfect compensation (i.e., $\dot{X}_{\text{actual}} \approx \dot{X}_{\text{ideal}}$), the active controller must be designed such that [102]:

$$B(x)u_{\text{passive}} \approx B(x)(I - \hat{\beta})(u_{\text{passive}} + u_{\text{active}}) + E(x)\hat{f}_c \tag{2.48}$$

Solving for u_{active}, assuming $B(x)$ is invertible in the operating range, yields the active compensation law [103]:

$$u_{\text{active}}(t) = (I - \hat{\beta})^{-1}\left[\hat{\beta}u_{\text{passive}}(t) - B^{-1}(x)E(x)\hat{f}_c(x)\right] \tag{2.49}$$

This equation clearly shows that the active compensation is a function of the fault estimates ($\hat{\beta}$, $\hat{f}_c$) and the current value of the passive control signal u_{passive}. It highlights the critical dependence on the FDD module's output [102].
The figure *Figure 2.12* below, represent the block diagram for a Hybrid FTC with a Passive Baseline.

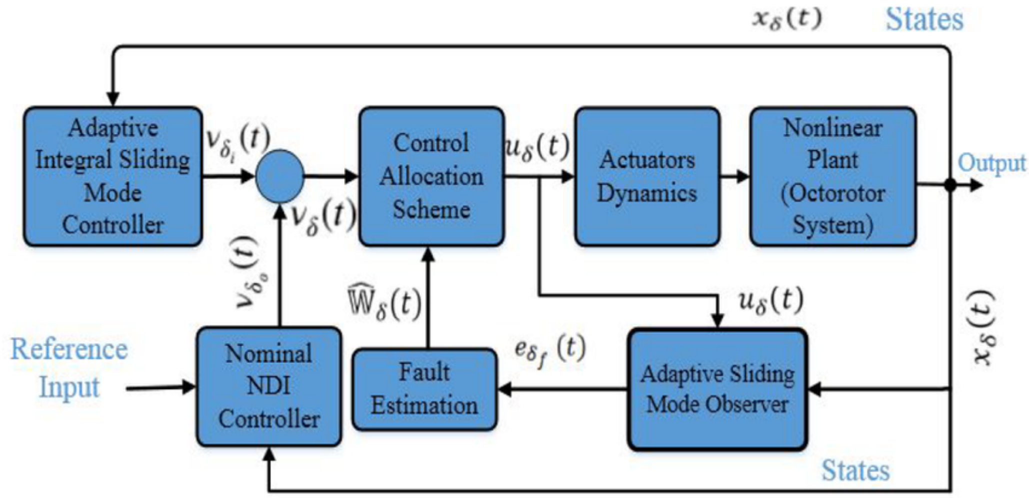

Figure 2.12: Architecture for a Hybrid FTC with a Passive Baseline

B. Critical Analysis: Advantages and Inherent Limitations:

B.1. Key Advantages:

-*Handling of FDD Latency:* This is the architecture's most significant advantage. There is no window of vulnerability between fault occurrence and its detection; the system remains stable throughout the FDD's processing time.

-*Modular and Verifiable Design:* The separation of tasks (stability vs. performance recovery) simplifies the design and verification process. The stability of the system can be proven by analysing the passive loop alone.

-*Reduced FDD Burden:* The FDD does not need to be instantaneous. A temporary failure or incorrect diagnosis is not catastrophic; it would simply result in the system reverting to the (sub-optimal but stable) performance level provided by the passive baseline [103].

B.2. Inherent Limitations and Challenges:

-*Conservative Performance Envelope:* The system's performance is fundamentally constrained by the passive baseline controller. Even with perfect active compensation, the nominal, fault-free performance is perpetually compromised to maintain robustness against faults that may never occur.

-*Actuator Saturation and Control Authority:* The total control command is the sum (up + ua). Both signals can be large, and their sum can easily exceed the physical limits of the actuators. This saturation can lead to significant performance degradation and potentially destabilize the "guaranteed" stable system [104].

-*Dependence on FDD Accuracy:* While the system is robust to FDD latency, it is not robust to FDD inaccuracy. As seen in *Eq. 2.49*, any error in the fault estimates will result in an imperfect cancellation signal, which acts as a new disturbance to the passive loop.

In summary, the passive baseline with active compensation architecture is a highly pragmatic and safety-conscious solution. It trades optimal performance for guaranteed reliability and predictable behaviour. Its primary value lies in applications where the cost of instability is prohibitively high. However, its inherent conservatism and significant challenges with control authority demonstrate that it is not a panacea, thus motivating the investigation of alternative hybrid structures and AI-based techniques.

2.2.3.2 Active Control with Passive Fallback (Safety-Net Approach)

This hybrid architecture represents a philosophical inversion of the passive baseline approach, prioritizing high performance and optimality during normal operation. The system operates by default under a sophisticated Active Fault-Tolerant Control (AFTC) scheme, designed for optimal performance and to provide intelligent accommodation for all anticipated faults. The "hybrid" nature lies in the inclusion of a secondary, dormant controller: a simple, robust, and extremely reliable passive controller that serves as a "safety-net" or "lifeboat" [105].

The critical distinction of this approach is the trigger for reconfiguration. The switch to the passive controller is not initiated by a fault in the plant itself, but rather by a detected *failure or loss of confidence in the supervisory control and diagnosis system.* This strategy is paramount in safety-critical applications where the complex AFTC system, while powerful, is acknowledged as a potential point of failure [106].

A. System Architecture and Switching Logic:

The architecture is a supervised switching system with two distinct control pathways and a high-level supervisory unit that decides which path is active.

The two control pathways are :

1. *The Primary AFTC Path ($\delta = 0$)*: This is the nominal operating mode. It includes a high-performance controller (e.g., an LQR, MPC, or adaptive controller) and a sophisticated FDD module. This path is responsible for achieving optimal performance metrics (e.g., fast tracking, disturbance rejection) and for detecting and accommodating plant faults through a dedicated reconfiguration mechanism (as described in *Subsection 2.2.2.1*).

2. *The Fallback PFTC Path ($\delta = 1$)*: This path contains a simple, fixed-gain, robust controller (e.g., a low-order H_∞ or a chattering-free SMC). Its design objective is not performance, but *guaranteed stability preservation* under a wide range of conditions [105].

The decision to switch is governed by the Supervisor and Safety Monitoring Unit. This unit continuously assesses the health and reliability of the primary AFTC path. The switching trigger condition $\delta \to 1$ is a logical function of several factors [104]:

- *FDD Health Diagnostics*: The FDD module runs self-tests and can issue a "loss of confidence" flag if it detects internal inconsistencies.

- *State Trajectory Monitoring*: The supervisor monitors the system state $x(t)$. If the trajectory begins to diverge towards the boundary of a pre-defined "safe region," a failure is declared.

- *AFTC Output Monitoring*: If the AFTC module demands an infeasible control action or if its internal states become unstable, the supervisor can trigger the fallback.

B. Critical Implementation Challenge: The Bumpless Transfer:

The transition from the high-performance active controller to the simple passive controller is a point of significant vulnerability. The internal states of the two controllers are generally different, which can lead to a large, abrupt discontinuity—a "bump"—in the control signal at the moment of the switch [102]. This bump can induce severe transients that may destabilize the very system the switch was intended to save.

To ensure a bumpless transfer, the state of the dormant passive controller must be “conditioned” or “kept hot.” This involves continuously updating the passive controller’s state, x_{cp}, such that its output would match the active controller’s output if it were to be engaged. A common conditioning scheme is to propagate the passive controller’s state using [103]:

$$\dot{x}_{\text{cp}} = A_p \cdot x_{\text{cp}} + B_p \cdot y + L_c \left(u_{\text{active}}(t) - u_{\text{passive}}(t)\right) \tag{2.50}$$

With:

- n_p: number of states in the passive controller, m: number of control inputs (actuators), p: number of measured outputs (sensors),
- $x_{cp} \in (n_p * 1)$: The Conditioned Passive Controller State,
- $y \in (p * 1)$: The Plant Measurement Vector,
- $A_p \in (n_p * n_p)$: The Passive Controller State Matrix,
- $B_p \in (n_p * p)$: The Passive Controller Input Matrix,
- $L_c \in (n_p * m)$: Is the Conditioning Feedback Gain, this forces u_{passive} to track u_{active}, ensuring that at the moment of the switch, the control signal $u(t)$ remains continuous.

C. Analysis of Advantages and Disadvantages:

C.1. Key Advantages:

-*Optimal Nominal Performance:* The system is not burdened by a conservative passive controller during normal operation and can be tuned for high efficiency and fast response.

-*Ultimate Fail-Safe for the Control System:* It provides a robust, verifiable failure mode for the most complex part of the system—the AFTC module itself. This is critical for certification in aerospace and autonomous systems.

-*Handles Unanticipated Faults:* The fallback mechanism provides a safety net against completely unforeseen events that might confuse or cause the FDD logic to fail [104].

C.2. Inherent Limitations and Challenges:

-*Complexity of the Supervisor:* The supervisor is a complex piece of software in itself and introduces a new and critical point of failure.

-*Performance "Cliff":* The transition from the high-performance active mode to the low-performance passive mode is abrupt and results in a significant degradation of performance, which may be unacceptable in mission-critical phases.

-*Verification and Validation ($V\&V$):* The $V\&V$ of this hybrid system is extremely challenging, requiring exhaustive testing of both control paths, the supervisor's switching logic, and the bumpless transfer mechanism [106].

In summary, the active control with passive fallback architecture is tailored for systems where high performance is the norm, but catastrophic failure is not an option. It acknowledges the fallibility of complex diagnostic and control software and provides a robust, albeit low-performance, safety net. Its utility is highest in applications where a controlled degradation to a simple, stable state is a far better outcome than the risk of losing the entire system due to the failure of its complex "brain."

2.3 AI-Based FTC Methods for Nonlinear Systems

This section explores the application of Artificial Intelligence (AI) as a powerful alternative framework for designing the next generation of FTC systems. In stark contrast to conventional methods that require an explicit mathematical representation of the system and its faults, AI techniques leverage principles of computational intelligence to learn from data, approximate complex functions, and make decisions under uncertainty. This offers a promising pathway to directly address the key limitations of the conventional paradigm. For instance, the universal approximation capabilities of neural networks can capture complex nonlinear dynamics without recourse to first principles or linearization. The pattern recognition strengths of machine learning can enable more robust fault diagnosis from raw sensor data. The goal-directed learning of reinforcement learning allows for the discovery of optimal control policies in environments where a model is unknown or incomplete [72, 107].

The aim of this section is to investigate the principal methodologies within this emerging field. We will begin by exploring the use of expert knowledge and linguistic rules through *Fuzzy Logic Controllers*. We will then delve into the powerful function approximation and online learning capabilities of *Neural Network-based* FTC. Finally, we will examine the paradigm of *Reinforcement Learning*, where an agent learns an optimal fault-tolerant control policy through direct interaction with the system. Through this exploration, we will establish the foundational strengths of AI-based methods while also setting the stage for a critical analysis of their own unique advantages and challenges.

2.3.1 Fuzzy Logic Controllers for FTC

While the fundamental principles of Fuzzy Logic—including membership functions, inference engines, and defuzzification—were established in Chapter 01 as a diagnostic tool, their application in Fault-Tolerant Control (FTC) serves a different purpose: Decision Making and Control Reconfiguration.

In an FTC context, Fuzzy Logic Controllers (FLCs) are prized for their ability to encapsulate expert heuristic knowledge into the control loop. Unlike conventional controllers that require precise differential equations, an FLC can implement linguistic control strategies (e.g., "If the error is large, apply a strong corrective action"). This is particularly advantageous when the system model becomes uncertain due to a fault [107, 108].

2.3.1.1 Applications of Fuzzy Logic in Fault-Tolerant Control

A. Fuzzy Logic for Fault Diagnosis:

In this application, the FLC acts as an intelligent FDD system. The inputs are the *symptoms of the fault* (typically residuals from observers), and the outputs are the classification of the fault's type, location, and/or severity. The rule base is designed based on knowledge of the system's fault signatures [108]:

The rule base is designed based on knowledge of the system's fault signatures [107]:

- IF $residual_{01}$ is *High* AND $residual_{02}$ is *NearZero* THEN fault is *Actuator01_ Stuck*
- IF $residual_{01}$ is *SlowlyIncreasing* AND $residual_{03}$ is *Normal* THEN fault is *Incipient_ Sensor02_ Bias*

The key advantage is robustness to noise and ambiguity in the residuals compared to a crisp, threshold-based logic system.
The *Figure 2.13* below, represent the block diagram for a fuzzy logic-based fault diagnosis system [109].

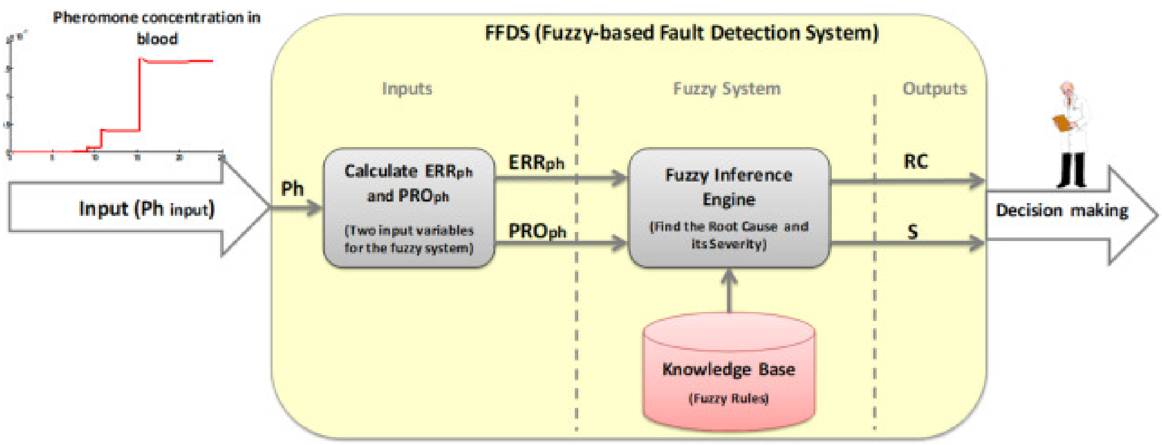

Figure 2.13: Fuzzy Logic-Based Fault Diagnosis System

B. Fuzzy Logic for Control Reconfiguration:

Here, the FLC acts as a high-level supervisor that intelligently modifies the control strategy based on diagnostic information. The inputs are the outputs from the FDD module, and the outputs are adjustments to the low-level controller's parameters or a signal to switch controllers. The rules encode a high-level control strategy [110, 111]:

- IF *fault* is *Minor_Actuator_Loss* AND *performance_error* is *Small* THEN *PID_gain_increase* is *Small*
- IF *fault* is *Severe_Sensor_Failure* THEN *switch_to_backup_estimator* is *TRUE*

This approach allows for a smooth, graded response to faults of varying severity, in contrast to a binary switching logic.

2.3.2 Neural Network-Based FTC

While the fundamental architecture of Artificial Neural Networks (ANNs) was defined in **Chapter 01 (Section 3.2)** as a diagnostic tool, their application in Fault-Tolerant Control (FTC) leverages a distinct theoretical property: *the Universal Approximation Theorem.* This theorem states that a feedforward network with sufficient hidden neurons can approximate any continuous nonlinear function to an arbitrary degree of accuracy.

In the context of FTC, we move beyond the static pattern recognition used in diagnosis and utilize ANNs for their dynamic learning capabilities. Rather than relying on fixed weights trained offline, Neural Network-based FTC often employs Adaptive Control schemes where the network weights are updated online to track changes in system dynamics caused by faults [112].

2.3.2.1 Neural Networks for Fault-Tolerant Control

NNs can be integrated into the control loop itself, typically in one of two ways: for system identification or for direct control adaptation.

A. NNs for System Identification and Model-Based FTC:

Instead of deriving a first-principles model, an NN can be trained to learn a model of the plant directly from its input-output data, predicting the next state or output [113]:

$$\hat{x}(k+1) = f_{\mathrm{NN}}(x(k), u(k)) \tag{2.51}$$

- n: Number of system states, m: Number of control inputs, k: The Discrete Time Step,
- $\hat{x}(k+1) \in (n * 1)$: The Predicted Next State Vector,
- $x(k) \in (n * 1)$: The Current State Vector,
- $u(k) \in (m * 1)$: The Control Input Vector,
- $f_{NN}() \in (\mathbb{R}^{n+m} \rightarrow \mathbb{R}^n)$: The Neural Network Mapping Function.

This learned NN model can then be used within a conventional model-based control framework, such as serving as the prediction model inside an MPC optimization loop or learning the nonlinear functions for a Feedback Linearizing controller.

In an FTC context, multiple NNs can be trained, one for the healthy plant and others for different fault conditions. The FDD module would then select the appropriate NN model for the controller to use after a fault is diagnosed [113, 114].

B. Online Learning and Adaptive Control with Neural Networks:

This is the most powerful application. Here, the NN is used to approximate the unknown or uncertain parts of the system dynamics *online*, and its weights are adapted in real-time. This is a direct extension of the conventional adaptive control concepts.

Consider a nonlinear system with uncertainties [112]:

$$\dot{x} = f(x) + g(x)u \tag{2.52}$$

Where $f(x)$ and $g(x)$ are unknown nonlinear functions. Our goal is to make the system track a reference model x_m. The control law is structured to cancel the nonlinearities [111]:

$$u = g^{-1}(x) * [-f(x) + v] \tag{2.53}$$

Where 'v' is a new control input. If $f(x)$ and $g(x)$ were known, this would linearize the system. Since they are not, we replace them with NN approximations, $\hat{f}(x|W_f)$ and $\hat{g}(x|W_g)$, where W_f and W_g are the NN weights.

The *Figure 2.14* below, represent the block diagram for a direct adaptive NN control for fault tolerant control [110].

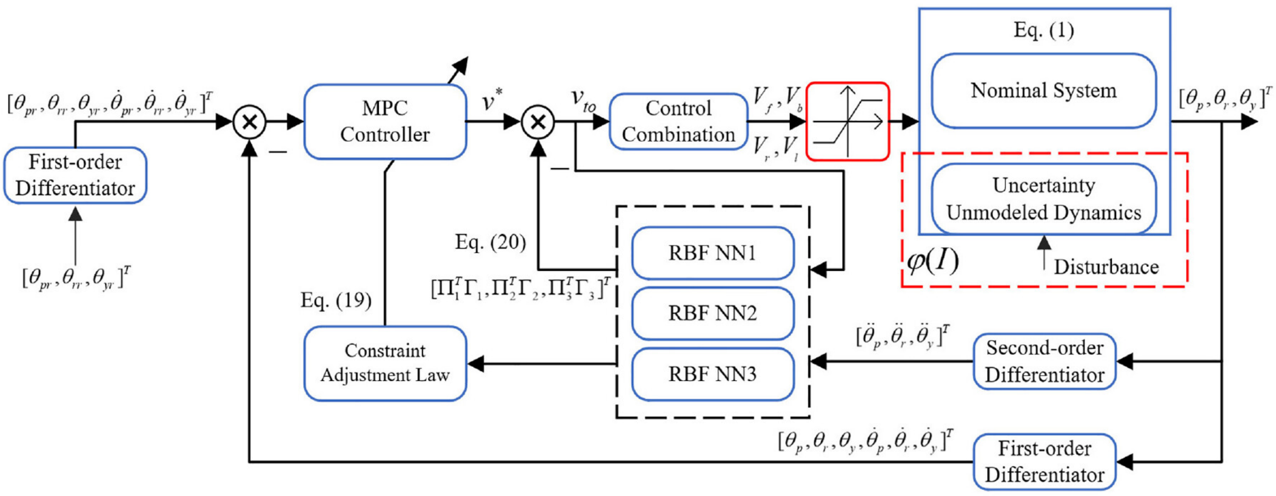

Figure 2.14: Direct Adaptive NN Control for Fault Tolerant Control

The key is to derive online adaptation laws for the NN weights W_f and W_g to ensure that the tracking error ($e = x - x_m$) converges to zero. This is typically achieved using a Lyapunov stability analysis. A typical resulting adaptation law for the weights $\tilde{W}$ of a generic

NN approximator is [110]:

$$\frac{d\tilde{W}}{dt} = -\Gamma \cdot e \cdot \sigma(\tilde{V}^T x) \tag{2.54}$$

Where 'Γ' is an adaptation gain matrix, 'e' is the tracking error, and $\sigma(\tilde{V}^T x)$ is the output of the NN's hidden layer (the regressor).

When a fault occurs, it abruptly changes the system's true dynamics $f(x)$ and $g(x)$. This causes an immediate increase in the tracking error e, which, through the adaptation law, drives the NN weights to new values that create a new approximation $\hat{f}$ and $\hat{g}$ that accurately represents the *new, faulty* system. This allows the controller to automatically compensate for the fault without an explicit FDD module [109].

2.4 Comparative Analysis: Conventional vs. AI for FTC

This section provides a direct comparative analysis of conventional and Artificial Intelligence-based approaches to Fault-Tolerant Control (FTC). The objective is to juxtapose the two paradigms along critical philosophical and operational axes to illuminate the fundamental trade-offs a control engineer faces. The core of the analysis reveals a fundamental dichotomy: the rigorous, model-based, and verifiable determinism of conventional methods versus the adaptive, data-driven, and powerfully expressive—yet often opaque—nature of AI-based systems.

2.4.1 Juxtaposition of Core Philosophies

The most profound difference between the two paradigms lies in their philosophical underpinnings—how they approach representing knowledge and making decisions.

-*Conventional Paradigm*: Conventional methods are predicated on the belief that a system's behavior can be captured by a set of mathematical equations derived from physical laws. Knowledge is explicit, formal, and encoded in state-space matrices, transfer functions, and differential inequalities. The design process is one of analytical deduction and formal proof. An H_∞ controller, for example, is the result of solving a set of Riccati equations, providing a mathematically provable guarantee of performance [91, 95].

-*AI Paradigm: The World as a Pattern*: AI-based methods operate on the premise that useful knowledge can be extracted empirically from data, even when the underlying physical laws are unknown. Knowledge is implicit, often distributed across millions of neural network weights or encoded in a set of linguistic rules. The design process is one of induction and empirical validation. An NN controller learns a control surface by observing outcomes, while an RL agent discovers a policy by optimizing a behavioral objective. Its power stems from this ability to learn and adapt without a formal model [107, 113].

This philosophical divergence leads to fundamentally different strengths and weaknesses, which are synthesized in *Table 2.2*.

Table 2.2: High-Level Philosophical and Operational Comparison of FTC Paradigms

Criterion	Conventional FTC Paradigm	AI-Based FTC Paradigm
Design Philosophy	Deductive & Analytical. Based on mathematical models and first principles.	Inductive & Empirical. Based on learning from data, experience, or interaction.
Knowledge Source	Mathematical Models. State-space, transfer functions, differential equations.	Raw Data & Heuristics. System I/O data, expert linguistic rules, interaction rewards.
Handling of Nonlinearity	Challenging. Often requires linearization, local analysis, or leads to intractable problems (e.g., HJI).	Inherent Strength. Universal approximators (NNs) can represent complex nonlinear functions directly.
Adaptation Mechanism	Pre-programmed & Parametric. Relies on gain scheduling, controller switching, or parameter tuning within a fixed structure.	Structural & Emergent. Can adapt internal model structure (NNs) or discover entirely new policies (RL).
Fault Diagnosis Approach	Analytical. Relies on observers and residual analysis based on a nominal model. Sensitive to model-reality mismatch.	Pattern Recognition. Learns fault signatures from data. Can be more robust to noise but depends on data quality.
Performance Guarantees	Primary Strength. Provides formal, provable guarantees of stability and performance bounds (e.g., H_∞ norm, Lyapunov stability).	Primary Weakness. Generally offers no formal guarantees; performance is validated empirically.
Flexibility to Novelty	Low. Cannot handle unanticipated faults outside the predefined uncertainty set or fault library.	High (Potentially). Online learning NNs and RL can potentially adapt to novel situations not seen in training.

2.4.2 Analysis of the Core Trade-Off: Verifiable Robustness vs. Adaptive Performance

The analysis illuminates the central trade-off in modern FTC design. This is not a choice between algorithms, but between two fundamentally different sets of priorities.

Conventional methods excel where safety, reliability, and predictability are the primary, non-negotiable requirements. For a civil aircraft, a mathematical proof that the controller will keep the system within a safe flight envelope is far more valuable than the ability to learn a slightly more fuel-efficient policy. The "brittleness" and conservatism of conventional methods are accepted because they create a bounded, analyzable problem space for certifiable safety [90, 94].

AI-based methods excel where the system is too complex to model, the environment is uncertain, and high performance in the face of novelty is the primary objective. For an autonomous rover exploring an unknown environment, the ability of an RL agent to discover novel strategies for unforeseen terrain is more valuable than a formal proof of stability for a simplified model. The opacity and lack of guarantees are accepted because the alternative—creating a conventional controller for every possible contingency—is simply impossible [114].

2.4.3 Mapping Conventional Challenges to AI-Based Solutions

To make this comparison more concrete, *Table 2.3* directly maps the most significant challenges of conventional FTC to the specific AI-based techniques that offer a potential solution.

Table 2.3: Mapping Conventional FTC Challenges to Potential AI-Based Solutions

Challenge in Conventional FTC	**Potential AI-Based Solution & Rationale**
Difficulty of modelling complex nonlinear plants.	Neural Network System Identification. The NN learns the input-output dynamics $f_{\text{NN}}(x, u)$ directly from data, providing a high-fidelity model for use in model-based control (e.g., MPC) without first-principles derivation.
Analytical intractability of solving nonlinear control problems (e.g., HJI inequality).	Reinforcement Learning Policy Optimization. The RL agent bypasses the need to solve the analytical problem entirely, instead discovering the optimal control law $\pi^*(s)$ through direct, goal-oriented interaction with the environment (or a simulation).
Fragility of observer-based FDD to noise and model uncertainty.	Fuzzy Logic or NN-Based Classifiers. These methods treat FDD as a pattern recognition problem. They can learn robust mappings from noisy residual signals to fault classes, often outperforming threshold-based logic in ambiguous situations.
The "chattering" phenomenon in Sliding Mode Control.	Fuzzy Logic Smoothing. An FLC can be used to implement the switching law. The overlapping nature of the fuzzy membership functions naturally smooths the control signal around the sliding surface, mitigating chattering while preserving robustness.
Inflexibility and combinatorial explosion of pre-defined controller libraries for active FTC.	Adaptive NN Control or Reinforcement Learning. Instead of switching between a finite set of controllers, an adaptive NN modifies its internal structure to create the appropriate control law online. An RL agent learns a single, complex policy that implicitly contains the logic for handling different fault modes.
Difficulty of designing controllers for systems with significant, unknown parameter variations.	Online Adaptive NN Control. The NN controller's weights are adapted online using Lyapunov-based update laws, allowing it to track slow or abrupt changes in the plant's parameters (i.e., incipient or sudden faults) automatically.

2.4.4 Conclusion of Comparative Analysis

This comparative analysis reveals that the choice between conventional and AI-based FTC is not a matter of one being universally superior. Rather, they represent two distinct and complementary toolkits. Conventional methods provide a foundation of rigor, reliability, and verifiable safety. AI methods provide a new frontier of adaptability, performance, and autonomy for systems of previously intractable complexity.

The critical insight, however, is that the trajectory of modern engineering systems—towards greater autonomy, higher complexity, and operation in more uncertain environments—is pushing ever harder against the boundaries of what is feasible with purely conventional techniques. This provides a powerful and undeniable motivation for a deep investigation into the AI-based paradigm. The challenge is to harness the immense potential of these learning-based methods while simultaneously developing techniques to instill them with the levels of safety, reliability, and trustworthiness required for real-world deployment.

2.5 Conclusion

This chapter has provided a comprehensive review of the state-of-the-art in Fault-Tolerant Control, identifying a fundamental dichotomy between established conventional techniques and emerging Artificial Intelligence paradigms. The investigation revealed that while conventional methods—whether Passive or Active—offer mathematical rigor and verifiable stability, they remain inherently constrained by their dependence on accurate system models. Conversely, AI-based approaches demonstrate exceptional adaptability to nonlinearities and unforeseen faults but introduce significant challenges regarding interpretability and the lack of formal safety guarantees.

Ultimately, this review crystallizes the core trade-off driving modern research: the conflict between verifiable robustness and adaptive performance. Bridging this gap requires a hybrid perspective that harnesses the learning power of intelligent systems while retaining the reliability of control theory. However, the successful implementation of any such advanced control law relies heavily on the accuracy of the underlying system representation. Consequently, Chapter 03 will address this foundational prerequisite by exploring the methodologies for the Modelling of Nonlinear Systems.

Chapter 3

Modelling of Nonlinear Systems: Conventional and AI-based Methods

3.1 Introduction

The vast majority of real-world physical and engineering systems, particularly those relevant to modern control and diagnostic applications, exhibit inherent nonlinear behavior. Unlike their linear counterparts, the dynamics of these systems are governed by complex differential equations capable of exhibiting phenomena such as multiple equilibria and chaos [115]. Consequently, the development of accurate and robust mathematical representations is a foundational prerequisite for the advanced Fault-Tolerant Control (FTC) and diagnosis strategies discussed in previous chapters.

This chapter systematically explores the methodologies for modelling nonlinear systems, charting a path from traditional physical derivation to modern data-driven identification. To provide a coherent analysis of this broad field, the discussion is structured around the fundamental "transparency" of the modelling approach.

The investigation commences by establishing a classification framework, distinguishing between White-box models derived from first principles, Black-box models inferred solely from data, and Grey-box hybrids. With this context established, the chapter first examines Conventional Modelling Approaches. This includes a review of deductive techniques based on physical laws (e.g., Lagrangian mechanics) and classical System Identification methods, such as NARX and Volterra series, which structure empirical data into mathematical forms.

Recognizing the limitations of conventional methods in the face of high complexity and uncertainty, the discussion then transitions to AI-Based Modelling. This section details how computational intelligence techniques, specifically Neural Networks and Fuzzy Logic (including Takagi-Sugeno models and ANFIS), utilize their universal approximation capabilities to learn nonlinear dynamics directly from operational data.

Moreover, the chapter synthesizes these findings through a formal Comparative Analysis, evaluating the trade-offs between physical interpretability and data-driven flexibility. The discussion concludes by explicitly connecting these modelling concepts to their practical application in Diagnosis and FTC, demonstrating how the choice of model directly influences the generation of residuals for fault detection and the design of stable control laws.

In conclusion, *Chapter 03* provides the essential Model-Based Foundation upon which the subsequent design and experimental validation of the novel Fault-Tolerant Control strategy (*Chapter 04*) will rely. It critically examines the tools necessary to capture the nonlinear behaviour of the UAV quadrotor, ensuring that the control system is designed upon a sound and accurate representation of the physical dynamics.

3.1.1 Importance of Modelling for Diagnosis and Control

A high-fidelity model of a system is indispensable for the development of effective fault diagnosis and fault-tolerant control (FTC) systems. In the context of diagnostics, a model provides a benchmark of the system's normal behaviour, against which real-time operational data can be compared to detect, isolate, and identify potential faults. An accurate model enables the generation of residuals—signals that are ideally zero in the absence of faults—which are crucial for robust fault detection [116].

For fault-tolerant control, the role of a system model is even more pronounced. An FTC system is designed to maintain system stability and acceptable performance not only under normal operating conditions but also in the presence of component or sensor faults. This often requires the control system to be dynamically reconfigured based on the information provided by the fault diagnosis scheme [117]. The model is used to predict the system's future behaviour under different fault scenarios and to synthesize control laws that can accommodate these faults, thereby ensuring the overall safety and reliability of the system. The effectiveness of any model-based diagnostic or FTC strategy is, therefore, fundamentally limited by the accuracy of the underlying system model [118]. This chapter will delve into the various methodologies available for modelling nonlinear systems, laying the groundwork for the subsequent development of intelligent diagnostic and fault-tolerant control techniques.

3.1.2 Classification of Models

The approaches to modelling nonlinear systems can be broadly categorized into three distinct classes: white-box, grey-box, and black-box models. This classification is based on the degree of prior knowledge about the system that is incorporated into the model.

a) ***White-box models***, also known as first-principle models, are based entirely on fundamental physical laws and established engineering principles. These models are derived from a deep understanding of the system's internal workings and are expressed through mathematical equations that describe the underlying physical phenomena. While white-box models offer the highest level of insight and extrapolation capabilities, their development can be exceedingly complex and time-consuming, especially for highly intricate nonlinear systems [118].

b) ***Black-box models***, in contrast, are data-driven and do not assume any prior knowledge about the system's internal structure. These models treat the system as an opaque box and aim to establish a mathematical relationship between the system's inputs and outputs based on observed data. Techniques such as neural networks and NARMAX (Nonlinear Autoregressive Moving Average with exogenous inputs) models fall into this category.

   While often easier to develop, the purely empirical nature of black-box models can limit their interpretability and their ability to generalize to operating conditions not present in the training data [119].

c) ***Grey-box models*** , represent a hybrid approach that combines elements of both white-box and black box modelling. These models incorporate partial theoretical knowledge about the system, such as its basic structure or some of its physical parameters, while using data-driven methods to identify the unknown or more complex parts of the model. Grey-box modelling seeks to leverage the strengths of both approaches, aiming for a balance between model accuracy, interpretability, and development effort. This approach is particularly valuable when a complete first-principle model is infeasible, yet some prior knowledge can be exploited to create a more robust and insightful model than a purely black-box approach would allow [120].

3.2 Conventional Modelling Approaches for Nonlinear Systems

Historically, the modelling of nonlinear systems has been approached from two primary perspectives: those grounded in fundamental physical laws and those based on empirical data from system observations. These conventional methods, predating the widespread adoption of artificial intelligence techniques, form the bedrock of system modelling and continue to be highly relevant. This section delves into these classical approaches, categorizing them into first-principle modelling and system identification techniques [117]. While first-principle models offer deep physical insight, system identification provides a structured methodology for constructing models from data when underlying principles are not fully known or are too complex to model explicitly.

3.2.1 First-Principle Modelling

First-Principle Modelling (FPM), also known as white-box or physics-based modelling, is a fundamental method for mathematically describing an engineering system. In electrical engineering, this approach relies on foundational laws like Kirchhoff's Current and Voltage Laws ((KCL)/(KVL)), Maxwell's equations, and Lorentz forces, as well as the principles governing semiconductor devices and electromechanical energy conversion. The goal of FPM is to create a model that reflects the system's intrinsic physical reality using its geometry, material properties, and the principles of electricity and magnetism, rather than empirical data [115].

The development of a first-principle model is a rigorous, systematic process that involves:

1) Defining the System Topology: Identifying all components (e.g., resistors, inductors, capacitors, semiconductor switches, rotating machines) and their interconnections.

2) Applying Fundamental Physical Laws: This step requires writing the differential and algebraic equations that describe the behaviour of the components and the energy flow between them. For circuits, this means applying KVL and KCL.

3) Incorporating Constitutive Relations: Material-specific and device-specific equations are introduced to define component behaviour. These relations, such as the $v - i$ characteristics of a diode or the flux-current $(\lambda - i)$ relationship in a saturable inductor, are often the main sources of nonlinearity [116].

4) Model Formulation and State-Space Representation: The derived equations are assembled into a coherent mathematical framework, typically the state-space form, which is well-suited for modern control and estimation theory [116, 118]:

$$\begin{cases} \dot{x} &= f(x, u, p) \\ y &= h(x, u, p) \end{cases} \tag{3.1}$$

Here, 'x' represents the state variables (e.g., inductor currents, rotor speed), 'u' is the external inputs (e.g., applied voltages), 'p' denotes the physical parameters (e.g., resistance, inductance), and 'y' is the measurable outputs. The nonlinear vector functions 'f' and 'h' capture the system's dynamics.

3.2.1.1 Illustrative Example: The Permanent Magnet Synchronous Motor (PMSM)

The Permanent Magnet Synchronous Motor (PMSM), a nonlinear system common in high-performance applications like electric vehicles and robotics, serves as an excellent illustration of the FPM process. Its nonlinear behaviour stems from the interaction between its electrical and mechanical subsystems and the sinusoidal nature of its physical construction [121].

To simplify the model, the three-phase stationary a-b-c reference frame is converted to a rotating d-q (direct-quadrature) reference frame aligned with the rotor's magnetic flux. This Park transformation converts time-varying sinusoidal variables into DC quantities in a steady state, aiding analysis and control design. However, the resulting equations remain inherently nonlinear [122]. *Figure 3.1* illustrates this modelling framework.

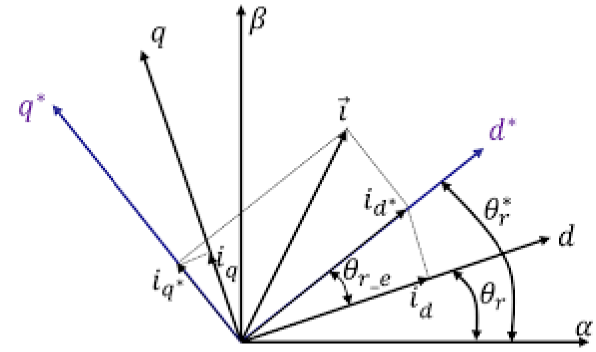

Figure 3.1: PMSM d-q Reference Frame

A. Electrical Subsystem Model: Applying Kirchhoff's Voltage Law to the stator windings and then transforming the resulting equations into the d-q frame yields the following voltage equations [121]:

$$\begin{cases} V_d &= R_s i_d + \frac{d\lambda_d}{dt} - \omega_e \lambda_q \\ V_q &= R_s i_q + \frac{d\lambda_q}{dt} + \omega_e \lambda_d \end{cases} \tag{3.2}$$

Where V_d, V_q are the d-q axis stator voltages; i_d, i_q are the d-q axis stator currents; R_s is the stator resistance; λ_d, λ_q are the d-q axis stator flux linkages; and ω_e is the electrical rotational speed of the rotor. The terms $(\omega_e \lambda_q)$ and $(\omega_e \lambda_d)$ are the "cross-coupling" or "speed-voltage" terms. They represent the electromotive force (EMF) induced in one axis due to the flux and rotation of the other axis. These product terms are a primary source of nonlinearity, coupling the electrical dynamics with the rotor speed [122].

$$\begin{cases} \lambda_d &= L_d i_d + \lambda_{pm} \\ \lambda_q &= L_q i_q \end{cases} \tag{3.3}$$

Here, L_d and L_q are the d-q axis inductances, and λ_{pm} is the constant flux linkage established by the rotor's permanent magnets. Substituting *eq.3.3* into *eq.3.2* and rearranging to isolate the current derivatives gives the state equations for the electrical subsystem [123]:

$$\begin{cases} \frac{di_d}{dt} &= \left(\frac{1}{L_d}\right)(V_d - R_s i_d + \omega_e L_q i_q) \\ \frac{di_q}{dt} &= \left(\frac{1}{L_q}\right)(V_q - R_s i_q + \omega_e L_d i_d - \omega_e \lambda_{pm}) \end{cases} \tag{3.4}$$

B. Mechanical Subsystem Model:

The mechanical dynamics are described by applying Newton's second law for rotation [123]:

$$J\frac{d\omega_m}{dt} = T_e - T_L - B\omega_m \tag{3.5}$$

Where 'J' is the combined moment of inertia of the rotor and load, ω_m is the mechanical rotor speed, T_e is the electromagnetic torque produced by the motor, T_L is the external load torque, and 'B' is the viscous friction coefficient. The electrical speed ω_e is related to the mechanical speed ω_m by the number of pole pairs 'p' [121]:

$$\omega_e = p * \omega_m \tag{3.6}$$

C. Electromechanical Coupling: Torque Equation

The crucial link between the electrical and mechanical subsystems is the electromagnetic torque T_e. For a PMSM, it is derived from the co-energy principle and expressed in the d-q frame as [122]:

$$T_e = \left(\frac{3}{2}\right) p(\lambda_d i_q - \lambda_q i_d) \tag{3.7}$$

Substituting the flux linkage equations *(3.3)* into *(3.7)* reveals the full expression for torque [124]:

$$T_e = \left(\frac{3}{2}\right) p\left[(L_d - L_q)i_d i_q + \lambda_{pm} i_q\right] \tag{3.8}$$

This equation highlights two distinct torque components: the magnetic torque ($\lambda_{pm} i_q$) and the reluctance torque ($(L_d - L_q)i_d i_q$), which arises if $L_d \neq L_q$ (as in Interior PMSMs). The reluctance torque term, being a product of two state variables (i_d and i_q), is another significant source of nonlinearity. Combining equations *(3.4)*, *(3.5)*, and *(3.6)* yields a third-order nonlinear state-space model for the PMSM. This model is fundamental for simulating motor behaviour and designing high-performance control strategies like Field-Oriented Control (FOC) [124, 125].

3.2.2 System Identification Techniques

When the complexity and cost of First-Principle Modelling (FPM) are prohibitive, System Identification (SysID) provides a powerful, data-driven alternative. This empirical methodology constructs mathematical models by observing a system's input-output data, treating it as a "black box" or "grey box" and using statistical methods to characterize its behaviour System Identification Techniques [117].

The SysID process is iterative and involves these key steps:

1) Data Acquisition: Designing an experiment to collect informative input-output data. The input signal must be exciting enough to reveal the system's dynamics.

2) Model Structure Selection: Choosing a mathematical model structure that determines the model's flexibility and complexity.

3) Parameter Estimation: Estimating the model's unknown parameters by fitting its predictions to the measured data. This is an optimization problem focused on minimizing a cost function, like the sum of squared errors.

4) Model Validation: Evaluating the model's accuracy on a separate dataset and analysing the statistical properties of the prediction errors (residuals) [119].

Selecting a model structure is especially challenging for nonlinear systems where superposition does not apply. Many nonlinear model structures exist; this section focuses on two influential families: Volterra and Wiener models, which use a functional expansion view of non-linearity, and NARMAX/NARX models, which provide a flexible discrete-time representation.

3.2.2.1 Volterra Series / Wiener Models

The Volterra series provides a foundational framework for representing nonlinear systems with memory, acting as a generalization of the Taylor series for static functions and the convolution integral for linear systems. It expresses the output $y(t)$ as an infinite series of multi-dimensional convolution integrals of the input $u(t)$, capturing higher-order dynamic effects [126].

For a causal, time-invariant, single-input single-output (SISO) system, the continuous-time Volterra series is given by [126]:

$$\begin{aligned} y(t) = h_0 &+ \int_0^{\infty} h_1(\tau)u(t-\tau)d\tau + \iint_0^{\infty} h_2(\tau_1,\tau_2)u(t-\tau_1)u(t-\tau_2)d\tau_1 d\tau_2 + \dots \\ &+ \underbrace{\int_0^{\infty} \dots \int_0^{\infty}}_{\text{n-integrals}} h_n(\tau_1,\dots,\tau_n)\prod_{i=1}^{n} u(t-\tau_i)d\tau_i + \dots \end{aligned} \tag{3.9}$$

Here, $h_n(\tau_1,\dots,\tau_n)$ is the n-th order Volterra kernel.

- The zeroth-order kernel, h_0, is simply the DC offset of the output.
- The first-order kernel, $h_1(\tau)$, is identical to the familiar impulse response of a linear system.
- The second-order kernel, $h_2(\tau_1,\tau_2)$, characterizes the quadratic nonlinear interactions of the input at different time lags, and so on for higher-order kernels.

While theoretically powerful, the direct identification of Volterra kernels is challenging due to the "curse of dimensionality." The number of parameters required grows exponentially with the order of the nonlinearity (n) and the system's memory length. This "parameter explosion" makes high-order Volterra models computationally intractable for most practical systems [127].

To overcome this, structured models like the Wiener model were developed. A Wiener model *Figure 3.2* is a block-oriented model with a linear time-invariant (LTI) dynamic block followed by a static nonlinear block.

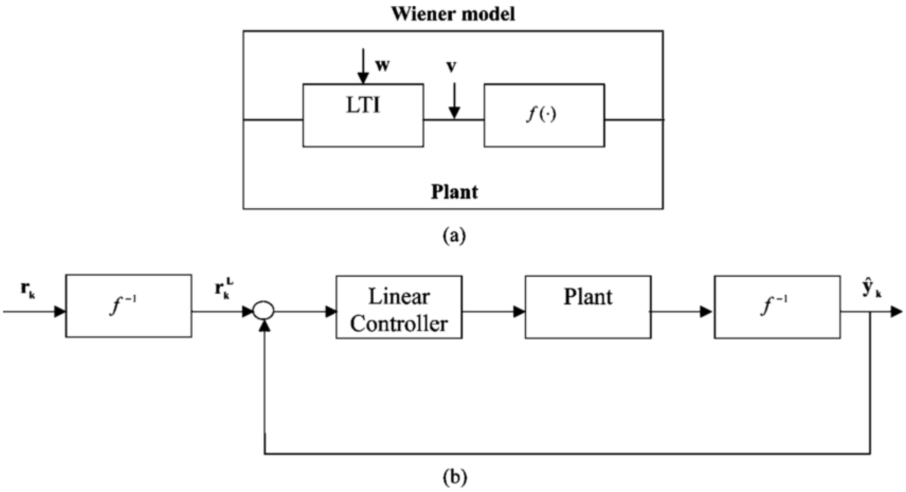

Figure 3.2: Block Diagram of a Wiener Model

The intermediate signal $z(t)$ is given by the convolution [127]:

$$z(t) = g(t) * u(t) = \int_{0}^{\infty} g(\tau)u(t-\tau)d\tau \tag{3.10}$$

- τ: The integration variable (or time lag/delay),
- $z(t)$: The Intermediate Signal (This is the output of the linear dynamic block and the input to the nonlinear block),
- $g(t)$: The Impulse Response Function of the Linear Time-Invariant (LTI) subsystem.,
- $f()$: The Static Nonlinear Function. It maps the intermediate signal $z(t)$ to the output $y(t)$.

The final output is a nonlinear function of this signal [125]:

$$y(t) = f(z(t)) \tag{3.11}$$

The Wiener model structure significantly reduces complexity by separating the dynamic and static aspects of the system's behaviour. Identification is simplified, particularly when the input is Gaussian white noise, as cross-correlation techniques can be used to identify the linear and nonlinear parts separately. Its dual, the Hammerstein model, consists of a static nonlinear block followed by an LTI dynamic block [126, 127]. However, these block-oriented models impose a rigid structure that may not be representative of more complex, intertwined nonlinear dynamics.

3.2.2.2 NARMAX / NARX Models

A more general, flexible representation for discrete-time nonlinear systems is the Nonlinear Autoregressive Moving Average with exogenous inputs (NARMAX) model. It represents the current system output $y(k)$ as a nonlinear function of its past outputs, inputs, and prediction errors, making it a direct nonlinear extension of the ARMAX model [128].

The general NARMAX model is expressed as [128]:

$$y(t) = F\left[y(k-1), \ldots, y(k-n_y), u(k-d), \ldots, u(k-d-n_u), e(k-1), \ldots, e(k-n_e)\right] + e(k) \tag{3.12}$$

Where:

- $y(k)$, $u(k)$, and $e(k)$ are the system output, input, and prediction error at discrete time step k, respectively.
- n_y, n_u, and n_e are the maximum lags for the output, input, and error terms.
- 'd' is the system time delay.
- $F[\cdot]$ is a general, unknown nonlinear function.

The inclusion of error terms allows the model to represent stochastic dynamics, but estimating parameters in their presence is computationally demanding.

A very common and practical simplification is the nonlinear autoregressive with exogenous inputs (NARX) model, which omits the moving average terms [129]:

$$y(t) = F\left[y(k-1), \ldots, y(k-n_y), u(k-d), \ldots, u(k-d-n_u)\right] + e(k) \tag{3.13}$$

Here, $e(k)$ is treated as an uncorrelated, zero-mean residual. The key advantage of the NARX formulation is that its regressors (the arguments of $F[\cdot]$) are all known from past measurements.

The power of this approach lies in representing the function $F[\cdot]$, which is typically approximated using a series expansion, like a polynomial. For example, a second-degree polynomial NARX model might look like [130]:

$$\hat{y}(t) = c_1 y(k-1) + c_2 u(k-1) + c_3 y(k-1)^2 + c_4 u(k-1)^2 + c_5 y(k-1)u(k-1) \tag{3.14}$$

This model is nonlinear in its variables but, crucially, linear in its parameters (c_1 to c_5), which allows for efficient parameter estimation using linear least-squares algorithms once the model structure is determined [130].

Consequently, the central challenge in NARX/NARMAX modelling is structure selection, not parameter estimation. A model with too few terms will be inaccurate (*under-fitting*), while too many terms will generalize poorly (*over-fitting*). The most established solution is the Orthogonal Forward Regression (OFR) algorithm (or FROLS), which iteratively selects the most significant terms from a large set of candidates, adding the one that provides the greatest reduction in unexplained variance at each step [128].

While NARX models are effective in representing many nonlinear systems, their "black-box" nature offers limited physical insight, and their accuracy depends heavily on the quality of the identification data [129]. These limitations highlight the boundaries of conventional system identification and motivate the search for alternative modelling paradigms.

3.3 AI-Based Modelling Approaches for Nonlinear Systems

Conventional modelling techniques present a fundamental gap: First-Principle Models (FPM), while physically transparent, are often intractable for complex systems and brittle to unmodeled dynamics. Conversely, classical System Identification (SysID) methods produce opaque "black-box" models with poor generalization, limiting their use in fault diagnosis [116, 117]. This necessitates a paradigm shift towards more flexible and adaptive modelling frameworks.

This section introduces Artificial Intelligence (AI)-based modelling as a compelling solution. AI-based approaches, particularly those inspired by computational intelligence, use data-driven learning algorithms to automatically discover complex nonlinear relationships within data. Their core strength lies in their inherent capacity as universal function approximators, meaning they can approximate any continuous nonlinear function to an arbitrary degree of accuracy, allowing them to capture subtle dynamics that conventional models neglect [131].

This section will focus on two influential AI paradigms: Artificial Neural Networks (ANNs) and Fuzzy Logic (FL) systems.

- **Artificial Neural Networks**, inspired by the brain, excel at learning complex patterns from vast amounts of data. We will explore architectures like the Multi-Layer Perceptron (MLP), Radial Basis Function (RBF) networks, and Recurrent Neural Networks (RNNs).

- **Fuzzy Logic**, provides a framework for reasoning with imprecise information, mirroring human expert knowledge to create more interpretable "grey-box" models.

Furthermore, we will examine Neuro-Fuzzy Systems, which combine these two fields. By exploring these techniques, this section establishes the foundation for this thesis's central premise: that AI-based models provide a superior foundation for robust fault diagnosis and fault-tolerant control in complex nonlinear systems.

3.3.1 Neural Network Models

3.3.1.1 Multi-Layer Perceptron (MLP)

The Multi-Layer Perceptron (MLP), whose fundamental architecture and training mechanisms were established in **Chapter 01 (Section 3.2)**, serves as a universal approximator. However, in the context of nonlinear system modelling, a standard static MLP is insufficient because dynamic systems possess memory—their future state depends on past states and inputs [131].

To adapt the MLP for system identification, it is typically embedded within a NARX (Nonlinear AutoRegressive with eXogenous inputs) structure. As illustrated in the concept of "Black-Box" modelling, the input vector to the MLP is augmented with a Tapped-Delay Line (TDL) [128, 131].

To model dynamic systems, a static MLP is typically embedded within a NARX-like structure, as previously introduced in *eq.(3.13)*. The MLP acts as the powerful nonlinear function

$F[\cdot]$ [129]:

$$\hat{y}(k) = F_{\text{MLP}}\left[y(k-1), \dots, y(k-n_y), u(k-d), \dots, u(k-d-n_u)\right] \tag{3.15}$$

This tapped-delay line of past inputs and outputs provides the network with a "memory," allowing it to model dynamic behaviour [128]. The MLP's advantage over polynomial NARX models is its superior ability to approximate highly complex nonlinearities without parameter explosion.

3.3.1.2 Radial Basis Function (RBF) Networks

RBF networks are feedforward networks with a different operational principle. An RBF neuron's output is based on the distance between the input vector and a prototype vector called the neuron's "centre," resulting in a localized response.

The RBF architecture is simpler than an MLP's, typically with an input layer, a single hidden layer of RBF neurons, and a linear output layer [132], *Figure 3.3* below represent its architecture.

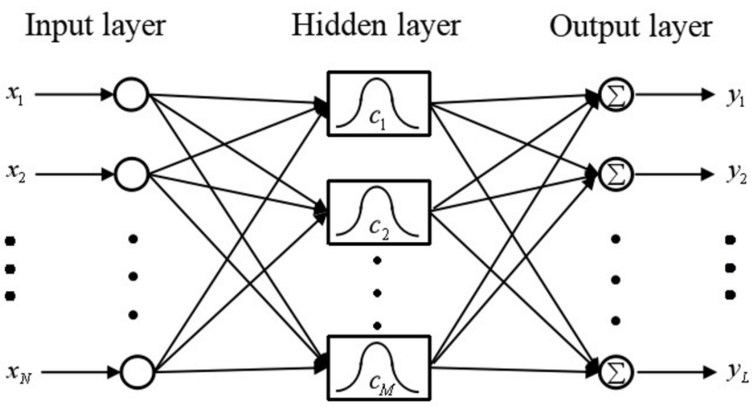

Figure 3.3: Architecture of a Radial Basis Function Network

The activation of a hidden RBF neuron is given by a radial basis function, most commonly a Gaussian kernel [133]:

$$\varphi_i(x) = \exp\left(\frac{-\|x - c_i\|^2}{2\sigma_i^2}\right) \tag{3.16}$$

Where:

- 'x' is the input vector.
- 'c_i' is the centre of the i-th neuron, which acts as a prototype for a region in the input space.
- 'σ_i' is the width or spread of the Gaussian function, controlling the size of the receptive field.
- '$\|\cdot\|$' denotes the Euclidean norm.

The network's output is then a simple linear combination of the activations of the hidden neurons [131]:

$$\hat{y} = \sum_{i=1}^{H} w_i \varphi_i(x) + w_0 \tag{3.17}$$

Where 'H' is the number of hidden neurons (centres), w_i are the weights of the output layer, and 'w_0' is a bias term.

A. Hybrid Learning Process:

RBF networks have a faster, more robust training process, often performed in two stages:

A.1) Unsupervised Learning of Hidden Layer Parameters: The centres $'c'$ and widths $'\sigma'$ are determined from the input data distribution, often using a clustering algorithm like K-Means.

A.2) Supervised Learning of Output Layer Weights: With the centres and widths fixed, the optimal output weights w can be found efficiently by solving a simple linear least-squares problem, which has a unique global solution [131].

3.3.1.3 Recurrent Neural Networks (RNN)

While feedforward networks can model dynamic systems using an external tapped-delay (NARX) structure, RNNs provide a more natural solution by introducing feedback loops into their architecture. This feedback endows the network with an internal memory, or "state," that evolves over time [76, 132].

A simple RNN processes a sequence of inputs. At each time step t, the activation of the hidden layer 'h_t' depends on both the current input 'x_t' and the hidden state from the previous time step 'h_{t-1}'. *Figure 3.4* bellow represents a "Folded" and "Unfolded" Recurrent Neural Network, where the left side shows the compact, folded representation of an RNN, with a loop indicating recurrence [77,112]. The right side shows the network "unfolded in time," illustrating how a single network processes a sequence by passing its hidden state from one time step to the next.

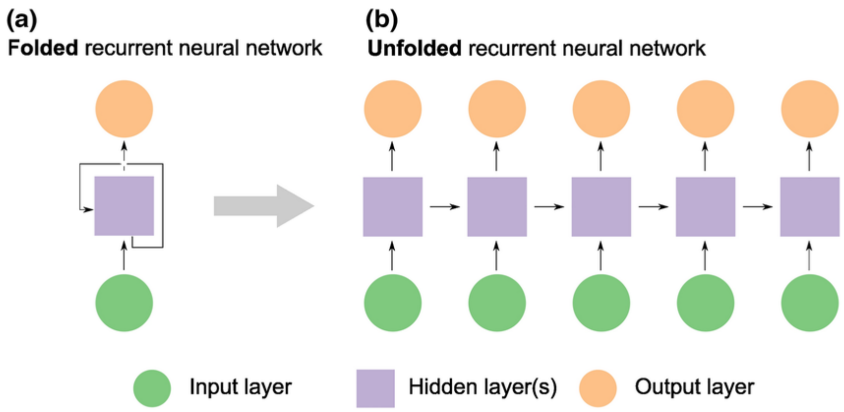

Figure 3.4: A "Folded" and "Unfolded" Recurrent Neural Network.

The governing equations for a simple RNN are [78]:

$$h_t = f_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{3.18}$$

$$\hat{y}_t = f_y(W_{hy}h_t + b_y) \tag{3.19}$$

Where:

- 'h_t' is the hidden state vector at time 't', serving as the network's memory.
- 'x_t' is the input vector at time 't'.
- W_{xh}, W_{hh}, W_{hy} are the input-to-hidden, hidden-to-hidden (recurrent), and hidden-to-output weight matrices, respectively.
- 'b_h', 'b_y' are the corresponding bias vectors.

This recurrent structure allows the network to learn temporal dependencies and model the internal dynamics of a system directly. Training is performed using a modified version of backpropagation called Backpropagation through Time (BPTT).

3.3.2 Fuzzy Logic Models

While the fundamental mechanisms of Fuzzy Logic (fuzzification, inference, defuzzification) were detailed in Chapter 01 for diagnosis and applied in Chapter 02 for control, their application in System Modelling requires a specific architecture: the Takagi-Sugeno (T-S) Model.

3.3.2.1 Limitations of Mamdani Models for Identification

The Mamdani inference system (discussed in previous chapters) is excellent for capturing expert linguistic heuristics. However, for nonlinear system identification, it suffers from a lack of mathematical tractability and difficulty in parameter optimization. It produces a "step-like" output that is often insufficient for modeling smooth dynamic systems like a UAV [31, 72].

3.3.2.2 Takagi-Sugeno (T-S) Fuzzy Models

To address the limitations of Mamdani models in regression and modeling tasks, Takagi and Sugeno proposed a hybrid structure. Unlike Mamdani rules which use fuzzy sets in the consequent (e.g., THEN Output is High), T-S rules use deterministic functions (typically linear) in the consequent [134].

A typical T-S model rule for a dynamic system is expressed as [134]:

$$\text{IF } z_1 \text{ is } M_1 l \ldots \text{ THEN } \dot{x} = A_i x(t) + B_i u(t) \tag{3.20}$$

This allows the nonlinear system to be represented as a weighted interpolation of local linear models, with:

- r: Total number of fuzzy rules,
- z_1: The Premise Variable (or Scheduling Variable),
- M_{1l}: The Fuzzy Set (or Membership Function),
- i: The Rule Index,
- A_i: The Local System Matrix (or State Matrix) for the $i-th$ rule,
- B_i: The Local Input Matrix for the $i-th$ rule.

A. Structure and Representation:

The Takagi-Sugeno (T-S) fuzzy model, introduced by Takagi and Sugeno (1985), represents a shift from the purely linguistic framework of the Mamdani model to a quasi-linear structure. This hybrid approach provides a formal and mathematically tractable way to represent complex nonlinear systems. Its significance lies in its computational efficiency and its implications for system analysis, state estimation, and controller synthesis, which are crucial for robust diagnosis and fault-tolerant control [135].

A.1. The Anatomy of a Takagi-Sugeno Fuzzy Rule: The fundamental building block of the T-S model is the IF-THEN rule, which establishes a local input-output relationship. A deep understanding of its two constituent parts—the premise and the consequent—is essential for appreciating its capabilities [134].

-The Premise (Antecedent): A Fuzzy Partition of the Operating Space

The premise, or antecedent, of a T-S rule serves to define a fuzzy region in the operational space of the system. Its function is to partition the complex, global behaviour of the system into a set of simpler, localized operating regimes. The general form is [136]:

$$\text{IF } z_1(k) \text{ is } A_{1i} \text{ AND } z_2(k) \text{ is } A_{2i} \text{ AND } \dots \text{ AND } z_p(k) \text{ is } A_{pi} \tag{3.21}$$

-*Premise Variables* $z(k)$: These are the variables upon which the partitioning is based, which can be external inputs $u(k)$, measurable system outputs $y(k)$ or states $x(k)$, or estimated states, are selected to be the most indicative of the system's current operating regime.

-*Linguistic Terms and Membership Functions* A_{ji}: Each premise variable $z_j(k)$ is associated with a fuzzy set A_{ji} (e.g., 'Low Speed', 'High Torque') defined by a membership function (MF), $\mu_{A_{ji}}(z_j)$ [136, 137].

-The Consequent: A Crisp Functional Mapping

The T-S model's consequent replaces the Mamdani model's fuzzy sets with a precise mathematical function, defining the system's local behaviour within the fuzzy region specified by the premise.

-*First-Order Affine Consequent*: This is the most common form, using a first-order affine function of the system's input variables [137].

$$y_i = a_{i0} + a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = [1, x_1, x_2, \dots, x_n] * [a_{i0}, a_{i1}, \dots, a_{in}]^T \tag{3.22}$$

The parameters a_{ij} are the consequent parameters of the i-th rule, and a_i is the vector of these parameters. This equation defines a local hyperplane that approximates the nonlinear system's behaviour in the region defined by the premise. The term a_{i0} is the offset or bias of this local model. This affine structure is highly effective for approximating a wide class of nonlinearities with a reasonable number of rules [138].

-*Zero-Order (Singleton) Consequent*: A simpler special case where the consequent is a constant value:

$$y_i = a_{i0} \tag{3.23}$$

This can be seen as a T-S model where all linear coefficients $(a_{i1}, \ldots, a_{in})$ are zero. It is computationally simpler but less expressive, as it only approximates the nonlinear function with a piecewise-constant surface [137].

A.2. The T-S Inference Engine: A Weighted Average Interpolator, The T-S inference engine aggregates the outputs of individual rules into a single, crisp global output through a weighted average.

-Calculating the Rule Firing Strength w_i:

For a given premise input vector z', the firing strength of each rule R_i is calculated by applying a t-norm operator to the membership degrees of the inputs in the antecedent fuzzy sets [139].

$$w_i(z') = T(\mu_{A_{1i}}(z'_1), \mu_{A_{2i}}(z'_2), \ldots, \mu_{A_{pi}}(z'_p)) \tag{3.24}$$

The choice of t-norm is a design decision with important practical implications. Common t-norm operators include the minimum (min) and product operators. The product operator is often preferred in data-driven systems as it results in a smoother model surface and is suitable for gradient-based optimization [138].

-The Global Output as a Convex Combination:

The final output $\hat{y}$ is the weighted average of all individual rule consequents y_i [140].

$$\hat{y} = \frac{\sum_{i=1}^{R} w_i(z') * y_i}{\sum_{i=1}^{R} w_i(z')} \tag{3.25}$$

This can be expressed using the normalized firing strength [139]:

$$\overline{w}_i(z') = \frac{w_i(z')}{\sum_{j=1}^{R} w_j(z')} \tag{3.26}$$

The overall output is then a convex combination of the local model outputs [141]:

$$\hat{y} = \sum_{i=1}^{R} \overline{w}_i(z') * y_i \tag{3.27}$$

This formulation is computationally elegant and reveals the T-S model's true nature: it is a smooth interpolator.

B. Identification Methods (Clustering, Optimization):

Constructing a T-S model from experimental data involves two interdependent problems: structure identification and parameter identification.

1. *Structure Identification*: This involves defining the fuzzy rule base, including the number of rules and the parameters of the antecedent membership functions.

2. *Parameter Identification*: This involves estimating the numerical values of the consequent parameters for each local model.

A common practice is to decouple these problems, first determining the structure using clustering algorithms and then estimating the parameters using optimization techniques [141,142].

B.1. Structure Identification via Data Clustering: Clustering algorithms are used to partition the system's operational space. The Fuzzy C-Means (FCM) algorithm is a standard method for structure identification in T-S modelling. FCM partitions a data set $X = \{x_1, \ldots, x_N\}$ into a specified number of clusters by minimizing the following objective function [142]:

$$J_m = \sum_{k=1}^{N} \sum_{i=1}^{C} u_{ik}^m \|x_k - c_i\|^2 \tag{3.28}$$

Where:

- 'C' is the pre-specified number of clusters (and thus, the number of fuzzy rules).
- 'x_k' is the k-th data point.
- 'c_i' is the center of the i-th cluster.
- 'u_{ik}' is the degree of membership of 'x_k' in the i-th cluster.
- '$(m > 1)$' is the weighting exponent or "fuzzifier."

The minimization of J_m is performed through an iterative process that alternates between updating the membership values u_{ik} and the cluster centres c_i :

1. *Initialization*: Initialize the cluster centres c_i (e.g., randomly or using a subset of data points).

2. *Membership Update*: Update the membership u_{ik} of each data point in each cluster:

$$u_{ik} = \frac{1}{\sum_{j=1}^{C} \left(\frac{\|x_k - c_i\|}{\|x_k - c_j\|} \right)^{\frac{2}{(m-1)}}} \tag{3.29}$$

3. *Centre Update*: Update the cluster centres 'c_i' as the weighted average of the data points:

$$c_i = \frac{\sum_{k=1}^{N} u_{ik}^m x_k}{\sum_{k=1}^{N} u_{ik}^m} \tag{3.30}$$

4. *Termination*: Repeat steps 2 and 3 until the change in the cluster centres or the objective function J_m between iterations is below a small tolerance 'ϵ' [141].

The identified cluster centres become the centres of the Gaussian membership functions. A drawback is that the number of clusters must be known in advance. Subtractive Clustering can be used to estimate the number of clusters and their initial locations [137].

B.2. Parameter Identification via Optimization: Once the antecedent structure is fixed, the consequent parameters can be identified. The T-S model output is linear with respect to these parameters, which can be solved using well-established optimization methods. For a set of N data points, the output can be written as [134]:

$$y(k) = \sum_{(i=1)}^{R} \overline{w}_i(k) * (a_{i0} + a_{i1}x_1(k) + \cdots + a_{in}x_n(k)) + e(k) \tag{3.31}$$

Where $e(k)$ is the modelling error. This can be expressed in a linear regression format [135]:

$$Y = \Phi\Theta + E \tag{3.32}$$

Where:

- Y is the vector of measured outputs.
- E is the vector of errors.
- Θ is the vector containing all unknown consequent parameters.
- Φ is the regression matrix.

The batch Least Squares Estimation (LSE) method is suitable for offline modelling. For online adaptation, Recursive Least Squares (RLS) can be used. This decoupling of identification is suboptimal. Advanced techniques, like those in neuro-fuzzy systems, perform a global optimization by simultaneously tuning both premise and consequent parameters [138].

3.3.3 Neuro-Fuzzy Systems for Modelling

AI-based modelling presents a dichotomy: Artificial Neural Networks (ANNs) are powerful "black-box" function approximators that are difficult to interpret, whereas Fuzzy Logic (FL) systems like the Takagi-Sugeno (T-S) model are interpretable "grey-box" structures but often rely on suboptimal, two-stage design processes [139].

Neuro-Fuzzy Systems integrate these paradigms, creating a system with the transparent structure of a fuzzy system and the learning capability of a neural network. This fusion allows fuzzy inference systems to be cast into a network architecture, enabling the use of ANN learning algorithms for automatic parameter identification from data [141].

The most prominent example is the Adaptive Neuro-Fuzzy Inference System (ANFIS), which is functionally equivalent to a T-S fuzzy inference system. ANFIS uses a hybrid learning procedure to simultaneously tune both premise (membership function) and consequent (local linear model) parameters, enabling systematic, data-driven construction of high-performance T-S models.

3.3.3.1 The ANFIS Architecture: A Multi-Layer Representation of a T-S Model

ANFIS employs a five-layer feedforward network that maps the computational flow of a first-order Sugeno-type fuzzy system. Each layer performs a specific function in the fuzzy inference process, and the model's adaptability stems from tunable parameters in certain layers.

For a simple T-S fuzzy system with two inputs, x_1 and x_2, and one output, y, a two-rule system would be [135]:

$$\text{Rule 1: IF } x_1 \text{ is } A_1 \text{ AND } x_2 \text{ is } B_1 \text{ THEN } y_1 = p_1 x_1 + q_1 x_2 + r_1 \tag{3.33}$$

$$\text{Rule 2: IF } x_1 \text{ is } A_2 \text{ AND } x_2 \text{ is } B_2 \text{ THEN } y_2 = p_2 x_1 + q_2 x_2 + r_2 \tag{3.34}$$

with:

- x_1, x_2: The Crisp Input Variables,
- A_1, A_2: The Fuzzy Sets associated with input x_1,
- B_1, B_2: The Fuzzy Sets associated with input x_2,
- p_1, p_2: The coefficients (weights) corresponding to input x_1 for Rule 1 and Rule 2, respectively,
- q_1, q_2: The coefficients (weights) corresponding to input x_2 for Rule 1 and Rule 2, respectively,
- r_1, r_2: The Bias Terms (or Constants) for Rule 1 and Rule 2.

The equivalent ANFIS architecture is detailed below in *figure 3.5*:

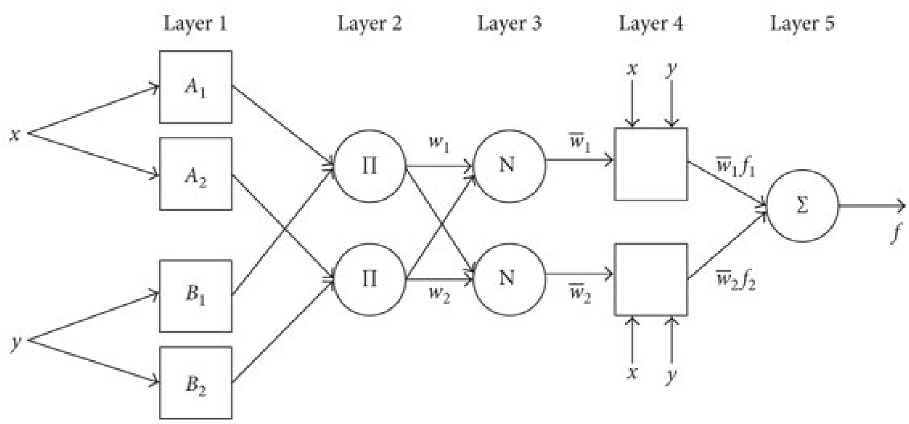

Figure 3.5: Five-Layer ANFIS Architecture

Layer 1: The Fuzzification Layer (Premise Parameter Layer): Every node 'i' in this layer is an adaptive node whose function is to compute the membership degree of an input to a fuzzy set. The node function is a membership function (MF) [136].

$$O_{1,i} = \mu_{A_i}(x) \tag{3.35}$$

The key feature is that the MFs are parameterized. For example, a generalized bell-shaped MF can be used [136]:

$$\mu_{A_i}(x) = \frac{1}{1 + \left|\frac{x-c_i}{a_i}\right|^{2b_i}} \tag{3.36}$$

Where

- $\{a_i, b_i, c_i\}$: is the set of premise parameters. These parameters, which define the shape and position of the membership functions, are adaptive and will be tuned by the learning algorithm,
- $O_{1,i}, O_{2,i}, O_{3,i}, O_{4,i}O_{5,i}$: Are the Outputs of layers 1,2,3,4 and 5 consecutively,
- w_i: The Firing Strength (or weight) of the $i-th$ rule,
- μ_{Ai}: The Membership Function (MF).

Layer 2: The Rule Antecedent Layer (t-norm Layer): Each fixed node in this layer implements the fuzzy AND operation by multiplying incoming signals, calculating the *firing strength* of each rule [137].

$$w_i = O_{2,i} = \mu_{A_i}(x_1) * \mu_{B_i}(x_2) \tag{3.37}$$

Layer 3: The Normalization Layer: Every fixed node in this layer computes the ratio of the i-th rule's firing strength to the sum of all rules' firing strengths, yielding the normalized firing strength [138].

$$\overline{w}_i = O_{3,i} = \frac{w_i}{(w_1 + w_2)} = \frac{w_i}{\sum_j w_j} \tag{3.38}$$

Layer 4 (Consequent Parameter Layer): Every node in this adaptive layer calculates the output contribution of a single T-S rule. The output is the product of the normalized firing strength and a first-order polynomial of the inputs [137].

$$O_{4,i} = \overline{w}_i * y_i = \overline{w}_i(p_i x_1 + q_i x_2 + r_i) \tag{3.39}$$

The parameters $\{p_i, q_i, r_i\}$ are the second set of adaptive parameters in the network.

Layer 5 (The Output Layer): This final layer has only a single fixed node labelled Σ. It computes the overall output of the ANFIS model by simply summing all the incoming signals from Layer 4 [139].

$$\hat{y} = O_{5,i} = \sum_i O_{4,i} = \sum_i \overline{w}_i * y_i = \frac{\sum_i w_i * y_i}{\sum_i w_i} \tag{3.40}$$

This fifth structure provides a precise, feedforward network representation of a T-S fuzzy model, enabling the application of network training algorithms for model identification.

3.3.3.2 ANFIS for Nonlinear System Identification: The Modelling Procedure

While the ANFIS architecture provides the learning framework, its application to modelling dynamic systems is a multi-step process. To model a dynamic system, the static ANFIS mapper must be embedded within an autoregressive framework, such as a NARX structure [140].

A. Step 1 (Data Acquisition and Preparation): During this step we collect a rich set of input-output data from the system. This data is divided into two sets:

- *Training Data Set*: Used by the learning algorithm to tune parameters.
- *Checking (or Validation) Data Set*: Used to evaluate the model's generalization ability and detect overfitting. The data is typically normalized before use to improve learning efficiency [141].

B. Step 2 (Defining the Model Structure (Regressor Selection)): Define the inputs to the ANFIS network, which are the past inputs and outputs (regressors) assumed to influence the current output. This establishes a NARX model structure [139]:

$$\hat{y}(k) = F_{\text{ANFIS}}\left[y(k-1), \ldots, y(k-n_y), u(k-d), \ldots, u(k-d-n_u)\right] \tag{3.41}$$

Here, $\hat{y}(k)$ is the one-step-ahead prediction of the system output. The vector $x_{\text{ANFIS}} = [y(k-1), \ldots, u(k-d-n_u)]$ forms the input vector to the ANFIS network.

C. Step 3 (Initializing the ANFIS Structure): Before training can begin, the initial structure of the fuzzy inference system within ANFIS must be specified. This involves:

- *Generating the Initial FIS*: A common approach is grid partitioning, where the user specifies the number and type of membership functions (MFs) for each input variable in x_{ANFIS}.
- *Alternative Initialization (Clustering)*: To avoid the high number of rules from grid partitioning, Subtractive Clustering can be applied to the data to estimate an appropriate number of rules and initial MF centres, leading to a more parsimonious model [142].

D. Step 4: Training the ANFIS Model: The ANFIS hybrid learning algorithm is executed for a set number of epochs. In each epoch:

1. The forward pass calculates the optimal consequent parameters $\{p_i, q_i, r_i\}$ via Least Squares Estimation (LSE) for the fixed premise parameters.
2. The backward pass propagates the error and updates the premise parameters $\{c_i, \sigma_i\}$ via gradient descent.

During this process, the error between the model output $\hat{y}(k)$ and the true output $y(k)$ on the training data is progressively minimized [141].

E. Step 5 (Model Validation and Refinement): A model that perfectly fits the training data is not necessarily a good model; it may have simply memorized the data (overfitting). Validation is the crucial final step to ensure the model has learned the underlying system dynamics and can generalize [142].

- *Performance Evaluation*: After each epoch, the model's performance (e.g., Root Mean Squared Error) is evaluated on the independent checking data set [134].

$$\text{RMSE} = \sqrt{\left(\frac{1}{N_{\text{check}}}\right) \sum_{k=1}^{N_{\text{check}}} \left(y_{\text{check}}(k) - \hat{y}_{\text{check}}(k)\right)^2} \tag{3.42}$$

  With:

  - N_{check}: The Total Number of Samples in the checking (validation) dataset,
  - k: The Sample Index,
  - $y_{check}(k)$: The Actual (Target) Output,
  - $\hat{y}_{check}(k)$: The Predicted (Estimated) Output.

- *Overfitting Detection*: The training and checking errors are plotted against epochs. Overfitting occurs when the checking error begins to rise while the training error continues to fall. Training should be stopped at the point of minimum checking error [135].

3.4 Comparative Analysis: Conventional vs. AI for Modelling

This chapter has introduced two distinct paradigms for modelling nonlinear systems: conventional, physics-based approaches and AI, computation-based approaches. This section provides a high-level, point-by-point comparison to clarify the fundamental trade-offs between them, justifying the methodological choices made later in this thesis.

The choice of modelling methodology is a critical design decision with significant implications for the performance and reliability of any subsequent diagnostic or control system.

3.4.1 Thematic Comparison Criteria

The comparison will be structured around the following key themes, which encapsulate the entire lifecycle and utility of a model for advanced engineering applications.

3.4.1.1 Basis of Knowledge and Model Interpretability

This criterion examines the origin of the model's information and the degree to which its internal structure is comprehensible to a human engineer.

-*Conventional Methods:* This paradigm is split. First-Principle Models (FPM) are "white-box," with every parameter corresponding to a physical quantity, which is invaluable for diagnosis. In contrast, System Identification (SysID) models like NARX are largely "black-box," as their polynomial coefficients lack physical meaning [117, 128].

-*AI-Based Methods:* Interpretability varies. Artificial Neural Networks (ANNs) are fundamentally "black-box" models; how they work is hidden in their complex structure. Conversely, T-S Fuzzy Logic systems (identified via ANFIS) are "grey-box." Their human-readable IF-THEN rules and local linear models can often be related to specific operating regimes, bridging data-driven accuracy with human insight [131].

3.4.1.2 Approximation Capability and Handling of Complexity

This theme addresses the model's ability to accurately represent complex, real-world nonlinear phenomena.

-*Conventional Methods:* FPMs are limited by simplifying assumptions made during their derivation and may fail to capture complex effects. Polynomial NARX models are constrained by their degree and suffer from the "curse of dimensionality," making them prone to overfitting [118].

-*AI-Based Methods:* This is a primary strength. As universal function approximators, AI models have a vast capacity to represent complexity. ANNs learn intricate mappings, while T-S fuzzy models blend simple local models to approximate any arbitrary function, capturing subtle dynamics often missed by conventional methods [132].

3.4.1.3 Data Requirements and Generalization

This criterion examines the reliance on experimental data and the model's ability to perform beyond its training set.

-*Conventional Methods:* FPM requires minimal input-output data for its structure, relying on physical laws. Its strength is extrapolation, making reasonable predictions for conditions outside the validation range. SysID methods are entirely data-dependent [119].

-*AI-Based Methods:* These models are data-hungry, with performance limited by the quality and richness of the training data. A key weakness is poor extrapolation. An AI model is an expert interpolator but can be unreliable outside its training domain, a critical safety concern for fault-tolerant control [133].

3.4.1.4 Development Effort and Process

This theme compares the nature and magnitude of the human effort required to construct the model.

-*Conventional Methods:* The effort is front-loaded and knowledge-intensive. FPM requires deep domain expertise and theoretical derivation. SysID demands careful experimental design and statistical analysis [115].

-AI-Based Methods: The effort shifts to data management and computational experimentation. Tasks include data acquisition, pre-processing, and an iterative process of designing the model architecture and tuning hyper parameters. Expertise is required more in machine learning and software engineering than in physics [112].

3.4.1.5 Suitability for Diagnosis and Formal Analysis

This final criterion directly assesses the utility of the models for the ultimate goals of this thesis. -*Conventional Methods*: FPM excels at physical diagnosis due to its "white-box" nature and is suitable for formal analysis. SysID models are less suited for diagnosis but can be used for fault detection.

-*AI-Based Methods*: Here, the distinction between ANNs and fuzzy systems is paramount.

- *ANNs*: are generally poor for diagnosis (beyond anomaly detection) due to their opacity. Their black-box nature makes formal stability analysis extremely difficult [77].
- *T-S Fuzzy Models*: Exceptionally well-suited for both diagnosis and control. Their "grey-box" structure allows for diagnostic monitoring of local model parameters. Critically, their quasi-linear, convex structure is amenable to rigorous stability analysis using Lyapunov theory and Linear Matrix Inequalities (LMIs), allowing for performance guarantees [135].

3.4.2 Summary Table of Comparative Analysis

Table 3.1 provides a concise summary of this comparative analysis, contrasting the key modelling paradigms discussed throughout this chapter.

3.4.3 Concluding Remarks on the Comparative Analysis

This analysis reveals a clear landscape of trade-offs, with no single methodology being universally superior. FPM is the gold standard for physical insight but is often impractical. SysID provides a baseline but lacks transparency. Within AI, a critical trade-off exists between the raw power of ANNs and the interpretable, tractable structure of T-S fuzzy systems.

For the objectives of this thesis—designing robust fault diagnosis and fault-tolerant control systems—interpretability and the potential for formal analysis are essential. Therefore, this analysis strongly motivates a focus on the grey-box neuro-fuzzy approach. It offers the most promising compromise, harnessing data-driven learning within a structured, rule-based framework that allows for physical interpretation and rigorous mathematical analysis. Subsequent chapters will build upon this conclusion, using T-S fuzzy models as the core technology.

Table 3.1: Holistic Comparison of Modelling Methodologies for Diagnosis and Control

Criterion	**First-Principle Modelling (FPM)**	**Conventional SysID (NARX)**	**Artificial Neural Networks (ANNs)**	**Neuro-Fuzzy Systems (T-S/ANFIS)**
Paradigm / Model Type	White-Box	Black-Box	Black-Box	Grey-Box
Basis of Knowledge	Fundamental physical laws, geometry, material properties.	Empirical input-output data.	Empirical input-output data.	Empirical data guided by a rule-based structure.
Interpretability	***Excellent***: Parameters are physically meaningful.	***Poor***: Polynomial coefficients lack physical meaning.	***Very Poor***: Connection weights are entirely opaque.	***Good***: Rule base is linguistic; local models can be analysed.
Approximation Capability	***Limited*** by simplifying assumptions; struggles with unmodeled effects.	***Moderate***; limited by polynomial degree.	***Excellent***: Universal function approximator.	***Excellent***: Universal function approximator.
Data Requirement	***Low*** (for structure); data needed for validation/tuning.	***High***: Requires rich, persistently exciting data.	***Very High***: Requires large, comprehensive datasets.	***High***: Also requires rich data but can incorporate prior knowledge.
Suitability for Physical Diagnosis	***Excellent***: Direct link between parameters and faults.	***Poor***: Limited to residual-based detection.	***Poor***: Limited to anomaly detection.	***Good***: Faults can be linked to changes in local model parameters.

3.5 Application of models in Diagnosis and FTC

Having established a taxonomy of modelling methodologies, this chapter now addresses their practical application. A high-fidelity model is the foundational prerequisite for designing intelligent systems capable of monitoring their own health (Fault Diagnosis) and responding to anomalies (Fault-Tolerant Control). This section bridges modelling theory with the engineering domains of Fault Diagnosis (FD) and Fault-Tolerant Control (FTC), demonstrating how a mathematical representation of a system is leveraged to enhance safety and performance.

3.5.1 Model-Based Fault Diagnosis

Model-based fault diagnosis relies on the analytical redundancy present in a system's mathematical model. The core principle is to compare the behaviour predicted by the model with the actual measured behaviour of the physical plant. A significant discrepancy is interpreted as a fault symptom. This process involves two main tasks: residual generation and residual evaluation [46].

3.5.1.1 Residual Generation: The Heart of Fault Detection:

A residual is a signal that is ideally near zero when the system is operating normally but deviates significantly in the presence of a fault. The most common method for generating a residual is through an output observer or a one-step-ahead predictor. The model runs in parallel with the actual plant, and the residual is the error between the measured output, $y(k)$, and the model's estimated output, $\hat{y}(k)$ [51, 101]:

$$r(k) = y(k) - \hat{y}(k) \tag{3.43}$$

Under ideal conditions, $r(k) \approx 0$ for a healthy system. When a fault alters the plant's dynamics, its output $y(k)$ diverges from the healthy-state model's prediction $\hat{y}(k)$, causing the residual to become non-zero [102].

Figure 3.6 below, represent the residual generation scheme using an output observer [101].

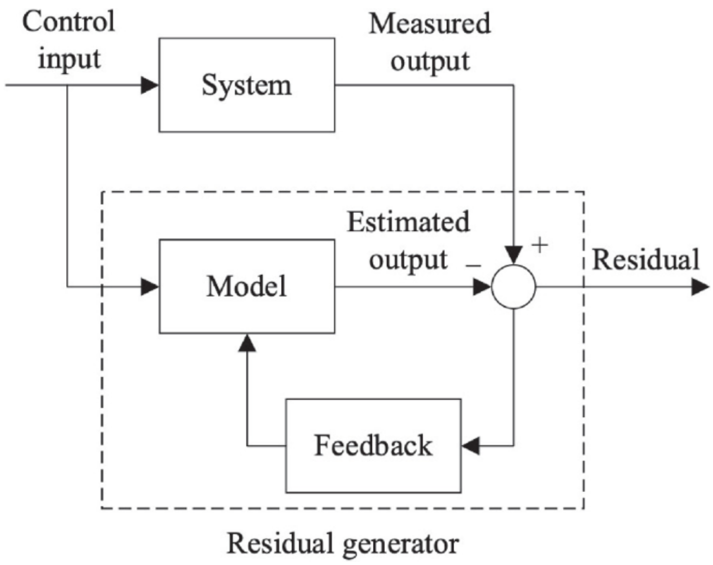

Figure 3.6: The Residual Generation Scheme using an Output Observer

3.5.1.2 Residual Evaluation and Fault Isolation

Once a residual signal is generated, it must be evaluated to make a decision.

Fault Detection: This is the task of deciding if a fault has occurred, typically by comparing the residual's magnitude against a pre-defined threshold. If $|r(k)| > J_{\text{th}}$, an alarm is triggered. The choice of J_{th} is a critical trade-off between sensitivity to faults and robustness against noise.

Fault Isolation: This is the task of determining the location and type of the fault. To isolate faults, a set of structured residuals is often required, typically achieved using a bank of models (or observers), where each is designed to be sensitive to certain faults and robust to others. By analysing which residual deviates from zero, the specific fault type can be deduced. The choice of model paradigm impacts this process; a bank of grey-box T-S models can provide deeper physical insight than black-box ANN models [55,136].

3.5.2 Model-Based Fault-Tolerant Control

Fault-Tolerant Control (FTC) refers to the ability of a control system to maintain stability and an acceptable level of performance in the presence of faults. While passive FTC achieves this through pre-designed robustness, *Active FTC*, which is the focus here, explicitly reacts to the fault. Active FTC systems rely on a model to understand the effect of the fault and to systematically compute a new control strategy to counteract it [80,86].

An active FTC system typically operates in a loop, as shown in *Figure 3.7*.

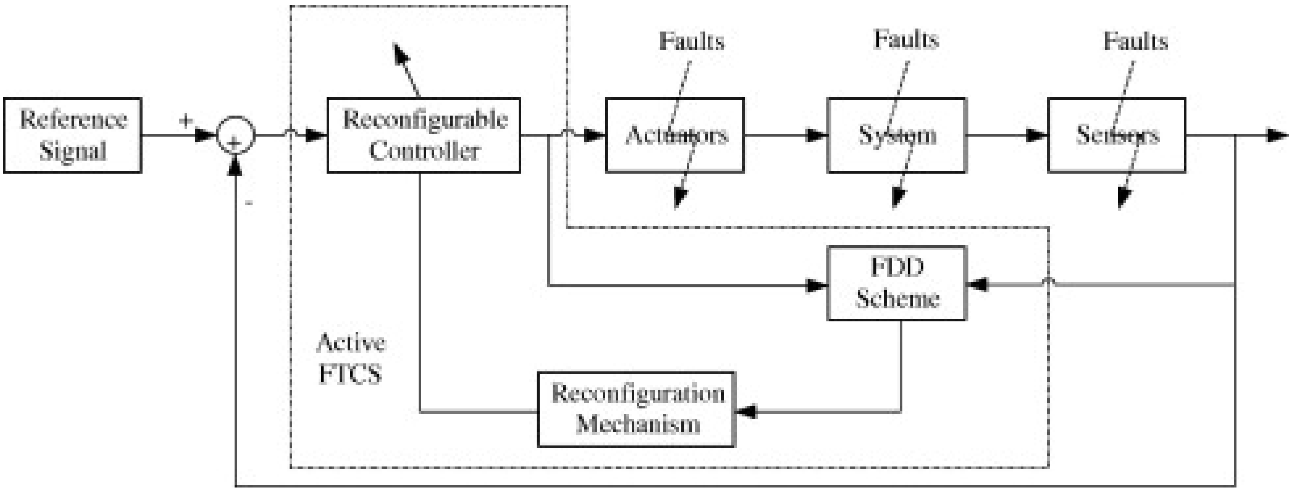

Figure 3.7: General Architecture of an Active Fault-Tolerant Control System

The model plays a central role in both the FDI and the Control Reconfiguration stages. Once the FDI unit has declared a fault, the model is used in one of several ways to achieve tolerance [87].

3.5.2.1 Control Law Rescheduling

This is the most straightforward approach to active FTC. During the offline design phase, a model of the plant is used to design a set of different controllers [88]:

- C_0: The high-performance controller for the healthy system.
- C_1: A controller specifically designed to handle Fault Type 1.
- C_2: A controller specifically designed to handle Fault Type 2.
- ...and so on.

These controllers are stored in a "bank of controllers." When the model-based FDI unit detects and isolates a specific fault (e.g., Fault 2), the reconfiguration mechanism is simply a supervisor that switches the active controller from C_0 to C_2. The model's primary role here is in the offline design of the contingent control laws [88,103].

3.5.2.2 Online Control Re-design and Re-tuning

A more sophisticated approach is to use the model to re-design the control law online after a fault has been identified. This is particularly effective for gradual degradation. The process is as follows [89, 143]:

1. The model-based FDI unit provides an online estimate of the faulty parameter, $\hat{\theta}(k)$.
2. This estimate $\hat{\theta}(k)$ is fed to the reconfiguration mechanism.
3. This mechanism uses an algorithm to re-calculate the controller's gains or structure based on the new plant parameter $\hat{\theta}(k)$ [105, 144].

This creates an adaptive system that continuously adjusts itself to the current health of the plant.

In conclusion, the model is an active and indispensable agent in both diagnosis and fault-tolerant control. The model's fidelity determines the FDI system's reliability, while its structure and interpretability dictate the sophistication of the possible FTC strategy—from simple controller switching to fully adaptive online re-design.

3.6 Conclusion

This chapter has completed the theoretical triptych of the thesis by analyzing the methodologies for representing nonlinear systems. The investigation highlighted that model selection is not merely a technical choice but a strategic trade-off between the physical transparency of First-Principle (White-box) derivation and the structural flexibility of Data-Driven (Black-box) identification. While conventional methods provide the mathematical rigor necessary for formal stability proofs, AI-based techniques—particularly Neural Networks and Takagi-Sugeno Fuzzy models—offer indispensable capabilities for capturing complex, unmodeled dynamics where physical laws become intractable.

Ultimately, accurate system modelling remains the central pillar of fault management, serving as the mathematical basis for both generating diagnostic residuals and tuning control parameters. With the foundations of diagnosis (*Chapter 01*), control theory (*Chapter 02*), and robust modelling (*Chapter 03*) now established, the research shifts from theory to application. *Chapter 04* will integrate these disciplines into a practical Case Study, applying the developed Hybrid Fault-Tolerant Control strategy to the challenging, highly nonlinear dynamics of a UAV Quadrotor.

Chapter 4

Case Study: Fault-Tolerant Control of UAV Quadrotors

4.1 Introduction

The emergence and rapid advancement of Unmanned Aerial Vehicles (UAVs) have revolutionized various sectors, with rotorcraft configurations playing a pivotal role. Among these, the quadrotor—a Vertical Take-Off and Landing (VTOL) aircraft characterized by its four independently controlled rotors—has garnered significant attention for its exceptional agility and mechanical simplicity [145], However, this popularity comes with inherent challenges; quadrotors are under-actuated, highly nonlinear, and open-loop unstable systems. Consequently, ensuring their operational reliability, particularly in safety-critical applications, necessitates the implementation of sophisticated Fault-Tolerant Control (FTC) strategies [146].

This chapter serves as the culmination of the theoretical research presented in previous chapters, providing the practical validation platform for the proposed Hybrid Fault-Tolerant Control (HFTC) strategy. The objective is to apply the novel control framework to the rigorous demands of a quadrotor system, demonstrating its ability to maintain stability and performance under severe actuator faults.

The investigation begins by establishing the physical context. It details the Quadrotor System Architecture, explaining the differential thrust mechanism and the imperative for fault tolerance in autonomous operations. Building on the modelling principles established in Chapter 03, the discussion then proceeds to derive a rigorous Nonlinear Dynamic Model. This derivation encompasses coordinate frames, kinematics, and the Newtonian mechanics governing the translational and rotational dynamics, ultimately formulating the complete state-space representation required for control design.

With the plant model established, the chapter defines the specific Fault Scenarios that challenge the system's integrity. Special emphasis is placed on modeling realistic actuator anomalies, such as Loss of Effectiveness (LOE) and rotor failures, as well as sensor degradations like bias and drift. These scenarios form the adversarial environment against which the proposed controller is tested.

The core technical contribution is then presented in the FTC System Design and Implementation. This section details the synthesis of the hybrid architecture, which uniquely integrates a verifiable Sliding Mode Control (SMC) law—enhanced by the novel Triple Power Reaching Law (ETPRL)—with an intelligent reconfiguration layer. It explains how Particle Swarm Optimization (PSO) is utilized offline to pre-compute an optimal "Bank of Parameters," and how an online Extended Kalman Filter (EKF) drives the real-time adaptation mechanism.

Finally, the efficacy of this framework is rigorously validated through Simulation and Analysis. The system is subjected to distinct, challenging flight maneuvers—including helical and square trajectories—under cumulative fault conditions. The resulting quantitative metrics regarding tracking error, chattering attenuation, and control effort provide the empirical evidence necessary to confirm the superiority of the proposed HFTC-ETPRL strategy over conventional benchmarks.

4.1.1 Quadrotor System Architecture and Core Components

A quadrotor Unmanned Aerial Vehicle (UAV) is a multi-rotor helicopter characterized by four identical rotors symmetrically arranged around a central rigid body or frame. This frame, often constructed from lightweight yet rigid materials like carbon fibre or aluminium, serves as the structural backbone, housing and protecting all on-board components while minimizing overall mass [147].

Each rotor consists of a brushless DC (BLDC) motor driving a fixed-pitch propeller. These motors are precisely controlled by Electronic Speed Controllers (ESCs), which translate low-power control signals from the central Flight Management Unit (FMU) into the necessary power to regulate motor speeds. Typically, two propellers rotate clockwise (CW) and two rotate counter-clockwise (CCW) in an 'X' or '+' configuration. This counter-rotating arrangement is crucial for neutralizing net aerodynamic torque, thereby simplifying attitude control and enhancing overall flight stability [148].

The FMU, or autopilot, is the computational core of the quadrotor, executing complex flight control algorithms and processing real-time sensor data. Critical to the FMU's operation is a comprehensive suite of on-board sensors. An Inertial Measurement Unit (IMU), with its accelerometers and gyroscopes, provides essential data for estimating the quadrotor's attitude and angular velocity. This is often augmented by a magnetometer for heading determination and a barometer for altitude estimation. For outdoor navigation, a Global Positioning System (GPS) module supplies absolute position and velocity data. Depending on mission requirements, additional sensors such as Lidar or ultrasonic sensors for obstacle avoidance, or optical flow cameras for vision-based navigation in GPS-denied environments, may be incorporated [145, 146]. The entire system is typically powered by a high-energy-density Lithium Polymer (LiPo) battery, while a robust communication module facilitates data exchange with a ground control station or remote pilot.

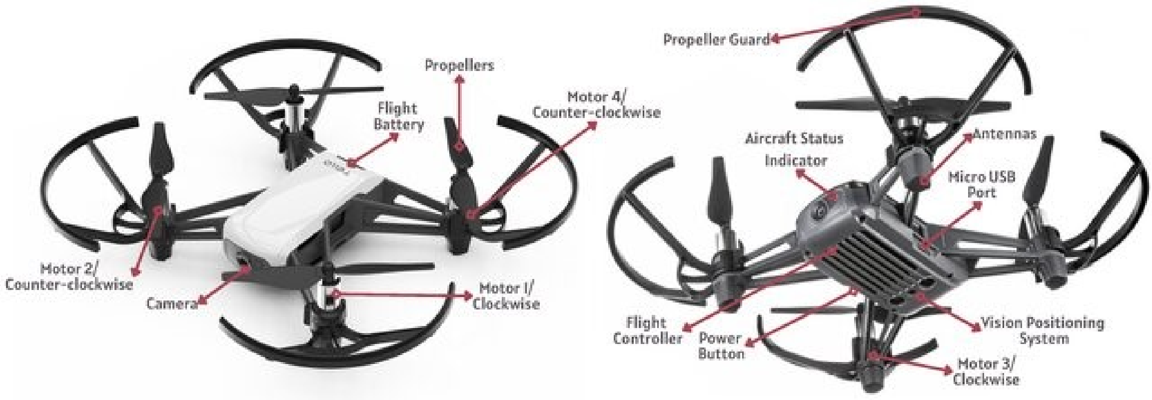

Figure 4.1: Structural description of quadrotor

4.1.2 Applications and the Imperative for FTC of UAV Quadrotors

The inherent advantages of quadrotors, including Vertical Take-Off and Landing (VTOL) and superior manoeuvrability, have led to their widespread adoption across diverse and safety-critical sectors. Applications range from commercial infrastructure inspection and precision agriculture to vital public safety roles like search and rescue (SAR) and urgent medical supply delivery [149]. This extensive deployment in high-stakes environments raises significant concerns regarding their operational safety and reliability.

The flight stability of a quadrotor is critically dependent on the continuous and precise control of multiple thrust-generating units. Consequently, any performance degradation, whether from internal malfunctions—such as actuator or sensor faults—or external disturbances, can rapidly lead to instability and catastrophic failure. Standard control systems, which assume perfect health, are ill-equipped to handle such anomalies [150].

This vulnerability underscores the paramount importance of Fault-Tolerant Control (FTC) systems, which are specifically engineered to detect, diagnose, and accommodate faults. The primary objectives of FTC are to enhance safety, increase reliability, maintain performance, ensure operational resilience, and reduce costs associated with accidents. Given the complex dynamics of quadrotors, designing effective FTC systems is a significant research challenge. This chapter, therefore, aims to present and validate advanced FTC strategies, particularly leveraging artificial intelligence, to ensure the dependable operation of UAVs in their myriad applications [148].

Figure 4.2 below, represent the logical flow for Fault-Tolerant Control (FTC) in UAV Quadrotor Applications [146].

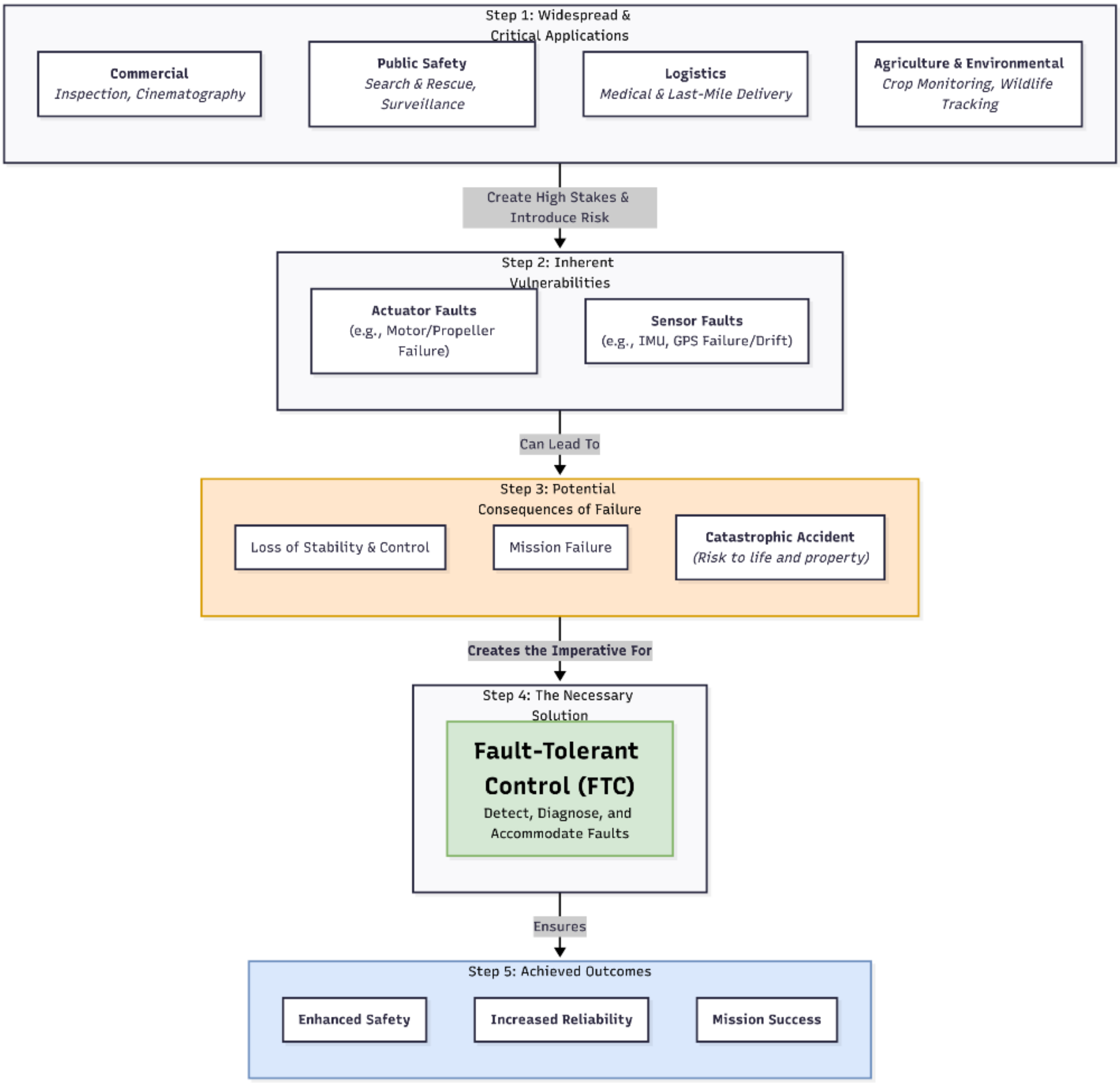

Figure 4.2: Logical Flow Illustrating the Imperative for Fault-Tolerant Control (FTC) in UAV Quadrotor Applications

4.2 Nonlinear Dynamic Model of the UAV Quadrotor

An accurate and comprehensive mathematical model is an indispensable prerequisite for the design, analysis, and implementation of effective control and fault diagnosis strategies. Given the quadrotor's inherent nonlinearities, state coupling, and under-actuated nature, a detailed dynamic model is essential for developing robust fault-tolerant control (FTC) [151].

The primary objective of this section is to derive the equations of motion that describe the quadrotor's position, orientation, and their rates of change in response to control inputs and external disturbances. This model, derived using the Newton-Euler formalism, will serve as:

- A baseline for control system design.

- A simulation platform for fault injection and fault tolerant control.
- A foundation for performance evaluation of the proposed AI-based FTC.

4.2.1 Coordinate Frames and Kinematics

To accurately describe the quadrotor's motion, two primary coordinate frames are defined, as depicted in *Figure 4.1*.

1. ***Earth-Fixed (Inertial) Frame*** $\{E\}$: This frame, denoted by (x_e, y_e, z_e), is considered inertial. Its origin is typically fixed at the ground, with the x_e-axis pointing North, y_e-axis pointing East, and z_e-axis pointing Down (NED convention) or Up (ENU convention, often preferred in robotics). For consistency with common control literature, we adopt the ENU convention, where z_e upwards. The linear position of the quadrotor's centre of mass (CoM) is denoted by [149]:

$$\xi = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{4.1}$$

With:

- ξ: The Linear Position Vector of the quadrotor's Center of Mass (CoM),
- x: Position coordinate along the Inertial $x_e - axis$,
- y: Position coordinate along the Inertial $y_e - axis$,
- z: Position coordinate along the Inertial $z_e - axis$.

2. ***Body-Fixed Frame*** $\{B\}$: This frame, denoted by (x_b, y_b, z_b), is attached to the quadrotor's CoM. The x_b-axis typically points forward, the y_b-axis to the right, and the z_b-axis upwards. The orientation of the body frame with respect to the inertial frame is described by Euler angles: roll (ϕ), pitch (θ), and yaw (ψ). The angular velocity of the quadrotor in the body frame is denoted by [150]:

$$\omega = \begin{bmatrix} p & q & r \end{bmatrix}^T \tag{4.2}$$

With:

- ω: The Angular Velocity Vector. It describes the rate of rotation of the quadrotor expressed in the Body-Fixed Frame B (the frame attached to the drone itself),
- p: The Roll Rate, (The rate of rotation around the body's longitudinal axis (x_b-axis, pointing forward)),
- q: The Pitch Rate, (The rate of rotation around the body's longitudinal axis (y_b-axis, pointing right)),
- r: The Pitch Rate, (The rate of rotation around the body's longitudinal axis (z_b-axis, pointing up)).

A clear depiction of these coordinate frames and the positive directions of the Euler angles is presented in *Figure 4.3*.

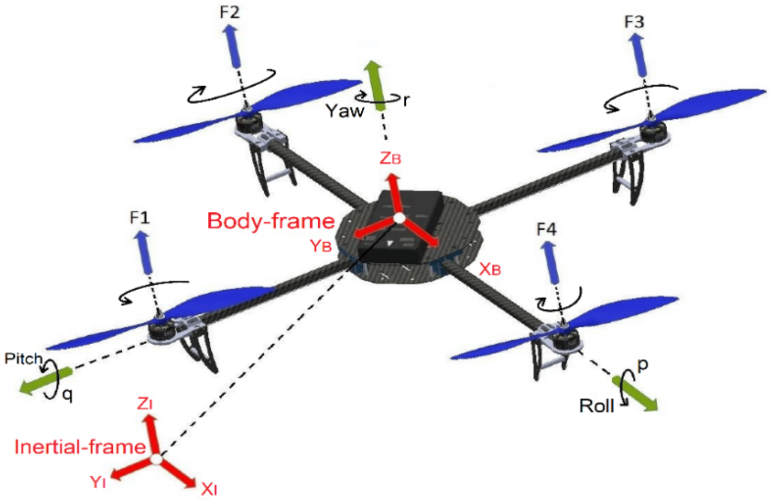

Figure 4.3: Structural coordinate system of quadrotor

The transformation from the body-fixed frame to the earth-fixed frame is accomplished via the rotation matrix $R_{E\leftarrow B}(\phi, \theta, \psi)$, which is a primary source of the system's kinematic nonlinearity [152]:

$$R_{E\leftarrow B} = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \tag{4.3}$$

The inverse transformation $R_{B\leftarrow E} = R_{E\leftarrow B}^T$

With:

- Φ: Is the roll angle, Θ: Is the pitch angle, Ψ: Is the yaw angle,
- $\dot{\Phi}$: Is the roll rate (rad/s), $\dot{\Theta}$: Is the pitch rate (rad/s), $\dot{\Psi}$: Is the yaw rate (rad/s),
- $C \equiv cos, S \equiv sin, t \equiv tang$: ,
- $R_{E\leftarrow B}$: The Rotation Matrix (Direction Cosine Matrix). It transforms a vector from the Body Frame B to the Earth/Inertial Frame E.
- η: The Euler Angle Vector (Orientation Vector),
- $\dot{\eta}$: The Euler Rate Vector.

The relationship between the body angular velocities $\omega = \begin{bmatrix} p & q & r \end{bmatrix}^T$ and the rates of change of Euler angles $\dot{\eta} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$ is given by the nonlinear transformation, or, conversely [151]:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{4.4}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & S_\phi t_\theta & C_\phi t_\theta \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi / C_\theta & C_\phi / C_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.5}$$

This matrix highlights the kinematic nonlinearities and the singularity at: $\theta = \pm\pi/2$

4.2.2 Forces, Moments and Control Inputs

The motion of the quadrotor is driven by forces and moments generated by its four rotors, as well as by external effects.

4.2.2.1 Rotor Thrust and Control Inputs

Each of the four rotors ($i = 1, 2, 3, 4$) generates a thrust force T_i and a reactive drag torque $M_{d,i}$. Assuming identical propellers and motors, the thrust T_i and drag torque $M_{d,i}$ are proportional to the square of the motor's angular velocity Ω_i [153]:

$$\begin{aligned} T_i &= k_T \Omega_i^2 \\ M_{d,i} &= k_M \Omega_i^2 = \gamma T_i \end{aligned} \tag{4.6}$$

Where k_T is the thrust coefficient, k_M is the drag moment coefficient, and $\gamma = k_M / k_T$ is a constant ratio. Rotors 1 and 3 typically rotate counter-clockwise (CCW), while rotors 2 and 4 rotate clockwise (CW), to cancel reactive yaw moments in hover. The total thrust $F_{\text{thrust},b}$ acts along the body z_b-axis :

$$F_{\text{thrust},b} = \begin{bmatrix} 0 \\ 0 \\ F_T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 T_i \end{bmatrix} \tag{4.7}$$

The control moments τ_{ctrl} generated by differential thrusts and reactive torques are defined in the body frame [154]. For a quadrotor with rotors at distance 'l' from the CoM, in an 'X' configuration [153]:

$$\begin{cases} U_1 = F_T = \sum_{i=1}^{4} T_i & \text{(Total thrust for vertical motion)} \\ U_2 = \tau_\phi = l(T_2 - T_4) & \text{(Roll torque)} \\ U_3 = \tau_\theta = l(T_1 - T_3) & \text{(Pitch torque)} \\ U_4 = \tau_\psi = M_{d,1} - M_{d,2} + M_{d,3} - M_{d,4} & \text{(Yaw torque)} \end{cases} \tag{4.8}$$

With τ_ϕ representing the roll moment around x_b-axis, τ_θ representing the pitch moment around y_b-axis, and τ_ψ representing the yaw moment around z_b-axis τ_{yaw}.

Here, the signs for yaw torque depend on the rotation direction of each rotor. If $M_{d,i}$ is positive for CW rotation and negative for CCW rotation, the expression should reflect this. Assuming $M_{d,i}$ is the magnitude of the drag torque, the signs are chosen to induce yaw [153].

The distribution of these forces and moments across the quadrotor body is visually summarized in *Figure 4.4*.

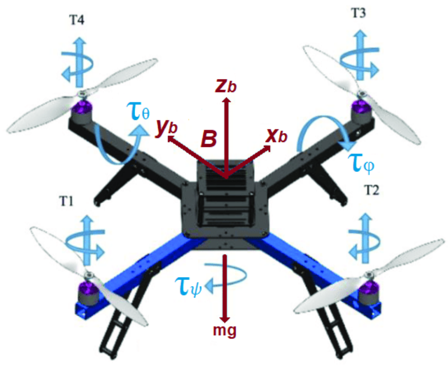

Figure 4.4: Representation of forces and moments acting on a quadrotor UAV in an 'X' configuration

4.2.2.2 External Forces and Moments

In addition to thrust and control moments, other forces and moments include:

- **Gravity**: Its equation in the inertial frame is [155]:

$$F_g = \begin{bmatrix} 0 & 0 & -mg \end{bmatrix}^T \tag{4.9}$$

  Where 'm' is the quadrotor mass and 'g' is the acceleration due to gravity.

- **Aerodynamic Drag**: For simplicity in this base model, often modelled as linear or quadratic in velocities. For translational motion [156]:

$$F_{d,e} = -K_v \dot{p} \tag{4.10}$$

  Where 'K_v' is a diagonal matrix of drag coefficients.

-For rotational motion, there is the following equation :

$$\tau_{d,b} = -K_\omega \omega \tag{4.11}$$

Where 'K_ω' is a diagonal matrix. More complex nonlinear drag models (e.g., quadratic) can be incorporated for higher fidelity.

- **Gyroscopic Moments**: From the rotating propellers, these moments act on the body when the quadrotor's attitude changes. The total gyroscopic moment can be approximated as [156]:

$$\tau_{\text{gyro}} = \sum_{i=1}^{4} J_r(k_i \Omega_i) \tag{4.12}$$

Where J_r is the propeller inertia, k_i is the unit vector along rotor axis, and Ω_i is the propeller speed. A simplified common form is [155]:

$$\tau_{\text{gyro},b} = J_r(q\Omega_y - r\Omega_z) \tag{4.13}$$

Where Ω_y and Ω_z are related to the sum of propeller speeds, considering their rotation directions.

4.2.3 Equations of Motion

The complete dynamic model of a quadrotor UAV is derived using the Newton-Euler formalism, which describes the translational motion based on Newton's second law and the rotational motion based on Euler's equations for rigid bodies. This approach allows for a systematic development of the equations of motion by considering all external forces and moments acting on the vehicle [157].

4.2.3.1 Translational Dynamics (Newton's Second Law)

The translational motion of the quadrotor's CoM in the inertial frame is governed by Newton's second law :

$$m\ddot{p} = F_E \tag{4.14}$$

Where 'F_E' is the total external force acting on the quadrotor in the inertial frame. This includes the transformed thrust force from the body frame, gravitational force, and aerodynamic drag [154]:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = R_{E \leftarrow B} \begin{bmatrix} 0 \\ 0 \\ F_T \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + F_{d,e} \tag{4.15}$$

Expanding the thrust component from $R_{E \leftarrow B}$ provides the full nonlinear equations for linear acceleration [152]:

$$\begin{cases} \ddot{x} &= \frac{F_T}{m}\left(C_\psi S_\theta C_\phi + S_\psi S_\phi\right) + \frac{F_{dx}}{m} \\ \ddot{y} &= \frac{F_T}{m}\left(S_\psi S_\theta C_\phi - C_\psi S_\phi\right) + \frac{F_{dy}}{m} \\ \ddot{z} &= \frac{F_T}{m}\left(C_\theta C_\phi\right) - g + \frac{F_{dz}}{m} \end{cases} \tag{4.16}$$

Where F_{dx}, F_{dy}, F_{dz} are components of $F_{d,e}$.

4.2.3.2 Rotational Dynamics (Euler's Equations)

The rotational motion of the quadrotor in the body-fixed frame is described by Euler's equations for a rigid body. Assuming a symmetrical quadrotor where the CoM aligns with the geometric centre and the inertia tensor 'I' is diagonal [158]:

$$I = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz}) \tag{4.17}$$

$$I\dot{\omega} + \omega \times (I\omega) = \tau_{\mathrm{total},b} \tag{4.18}$$

Expanding this equation for each axis [153]:

$$\begin{cases} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr &= \tau_\phi + \tau_{\mathrm{gyro},x} + \tau_{d,x} \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr &= \tau_\theta + \tau_{\mathrm{gyro},y} + \tau_{d,y} \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq &= \tau_\psi + \tau_{\mathrm{gyro},z} + \tau_{d,z} \end{cases} \tag{4.19}$$

Where $\tau_{\mathrm{gyro},x}, \tau_{\mathrm{gyro},y}, \tau_{\mathrm{gyro},z}$ are components of the gyroscopic moments and $\tau_{d,x}, \tau_{d,y}, \tau_{d,z}$ are components of the aerodynamic drag moments. The terms $(I_{zz} - I_{yy})qr$, $(I_{xx} - I_{zz})pr$, and $(I_{yy} - I_{xx})pq$ explicitly demonstrate the nonlinear coupling between angular velocities, a direct consequence of the body's rotation.

4.2.4 Complete Nonlinear Dynamic State-space Model

The aggregation of the kinematic and dynamic relationships forms the complete nonlinear model. The state of the system is typically defined by a vector of 12 variables [150]:

$$X = \begin{bmatrix} x & y & z & \phi & \theta & \psi & \dot{x} & \dot{y} & \dot{z} & p \\ q & r \end{bmatrix}^T \tag{4.20}$$

This model, while based on conventional physics, fully captures the nonlinear characteristics discussed—including trigonometric dependencies, strong state coupling, quadratic aerodynamic effects, and gyroscopic torques. It provides a robust and high-fidelity foundation for the development, implementation, and testing of the advanced AI-based fault diagnosis and fault-tolerant control methodologies presented in the subsequent sections of this chapter.

4.3 Fault Scenarios Definition

The reliable and safe operation of UAV quadrotors depends on the integrity of their actuators, sensors, and structural parts. These systems are susceptible to internal and external faults that can degrade performance, cause instability, or even lead to catastrophic failure. The development of robust fault diagnosis and fault-tolerant control (FTC) strategies requires a clear understanding of the types, characteristics, and effects of these potential fault scenarios [155].

This section categorizes and defines the primary fault types in quadrotor systems, which are the focus of the fault diagnosis and FTC methodologies in this chapter. These faults are generally classified as actuator and sensor faults, as they impact the system's ability to generate control inputs and perceive its state.

The selection of these fault scenarios is based on their prevalence in real-world UAV operations and their significant impact on safety-critical functions [156]. This systematic definition of fault scenarios provides the foundation for designing and evaluating the fault-tolerant control system, as discussed in *section 4*.

4.3.1 Actuator Faults

Actuator faults represent a critical class of malfunctions in quadrotor systems, directly affecting the vehicle's ability to generate commanded forces and moments. These faults can arise from various issues, including mechanical wear, motor controller failures, propeller damage, or power supply irregularities. Their immediate impact is a reduction or alteration of the expected thrust and/or torque produced by one or more rotors, directly leading to a discrepancy between the commanded control input and the actual force/moment exerted on the airframe [159]. *figure. 4.5* illustrates the nominal, healthy operation of a quadrotor, providing a baseline for comparison with faulty conditions.

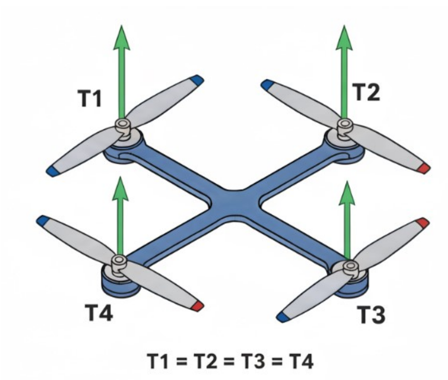

Figure 4.5: Nominal quadrotor operation (Hovering) where all the propellers produce equal thrust

This section details the most common and impactful actuator fault scenarios that will be considered for diagnosis and fault-tolerant control.

4.3.1.1 Loss of Effectiveness (LOE) Fault

A loss of effectiveness (LoE) fault is one of the most prevalent and challenging actuator malfunctions. It describes a situation where a rotor produces less thrust and/or torque than commanded, even when the motor is receiving the correct control signal. This can be caused by partial motor winding damage, reduced motor efficiency, propeller erosion, minor blade cracks, or issues within the electronic speed controller (ESC) that limit power delivery. The LoE fault is typically modelled as a multiplicative factor affecting the commanded thrust of a specific rotor [160].

For the i-th rotor, the actual thrust $T_{i,\text{actual}}$ in the presence of an LoE fault is given by [159]:

$$T_{i,\text{actual}} = \rho_i T_{i,\text{commanded}} \tag{4.21}$$

Where $T_{i,\text{commanded}}$ is the thrust commanded by the controller for rotor 'i', and $\rho_i \in [0, 1]$ is the effectiveness factor for rotor 'i'.

- If $\rho_i = 1$, the actuator is fully effective (nominal operation).
- If $\rho_i = 0$, the actuator has completely failed (total loss of thrust).
- If $0 < \rho_i < 1$, the actuator has a partial loss of effectiveness.

The same effectiveness factor ρ_i typically applies to the drag moment $M_{d,i}$ associated with that rotor, meaning $M_{d,i,\text{actual}} = \rho_i M_{d,i,\text{commanded}}$ [161]. *Figure 4.6* visually represents an LoE fault on a single rotor and its resulting impact on thrust distribution.

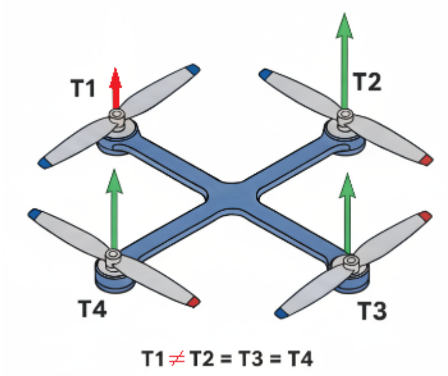

Figure 4.6: Quadrotor Suffering from LOE fault in motor 01

Impact on Dynamics: An LoE fault on a single rotor directly reduces the total thrust (if $\rho_i < 1$) and, more critically, creates an imbalance in the control moments. For instance, a partial LoE on Rotor 1 (part of the pitch control pair) will reduce the effective pitching moment generated, requiring other rotors to compensate [161]. This leads to :

- *Reduced Control Authority*: The maximum achievable force and torque are diminished.
- *Uncommanded Accelerations/Rotations*: The imbalance in forces/moments causes the quadrotor to drift or rotate in an uncommanded direction.

- *Steady-State Error*: Without compensation, the quadrotor may not be able to maintain its desired position or attitude, leading to persistent errors.
- *Increased Control Effort*: The remaining healthy actuators must work harder, potentially leading to saturation or increased power consumption.

4.3.1.2 Stuck Rotor Fault

A stuck rotor fault occurs when a motor or its propeller jams at a specific thrust level, becoming unresponsive to control commands. This can be due to mechanical obstruction, bearing failure, or a complete ESC malfunction. This fault represents a severe condition where the actuator is no longer controllable, operating at a fixed, potentially non-zero, thrust [145].

Mathematically, for a stuck rotor 'i', the actual thrust $T_{i,\text{actual}}$ and drag moment $M_{d,i,\text{actual}}$ are [146]:

$$T_{i,\text{actual}} = T_{i,\text{stuck}} \tag{4.22}$$

$$M_{d,i,\text{actual}} = M_{d,i,\text{stuck}} = \gamma T_{i,\text{stuck}} \tag{4.23}$$

Where $T_{i,\text{stuck}}$ is the constant thrust level at which the rotor is jammed. This value can range from zero (rotor completely off but jammed) to its maximum possible thrust.

Impact on Dynamics: A stuck rotor fault has a profound impact due to the loss of controllability for that actuator :

- *Severe Control Asymmetry*: The fixed thrust and drag moment from the stuck rotor create a constant disturbance that cannot be counteracted by commanding changes to that specific actuator.
- *Reduced Degrees of Control*: Effectively, the system loses one independent control input for attitude and thrust distribution.
- *Necessity for Redistribution*: The remaining healthy rotors must be reconfigured to compensate for the fixed force/moment, often requiring significant adjustments and potentially operating at their limits.
- *Instability*: If not promptly detected and accommodated, a stuck rotor fault can quickly lead to loss of stability, severe oscillations, and an unrecoverable flight condition [147].

4.3.2 Sensor Faults

Beyond actuator malfunctions, the reliable operation of a quadrotor heavily relies on accurate and timely information about its state, which is provided by its on-board sensor suite. Sensor faults introduce erroneous readings into the control loop, potentially leading to incorrect state estimation, flawed control actions, and ultimately, instability or mission failure, even if actuators are fully functional [148]. *figure. 4.7* illustrates a typical nominal sensor measurement, providing a baseline for comparison.

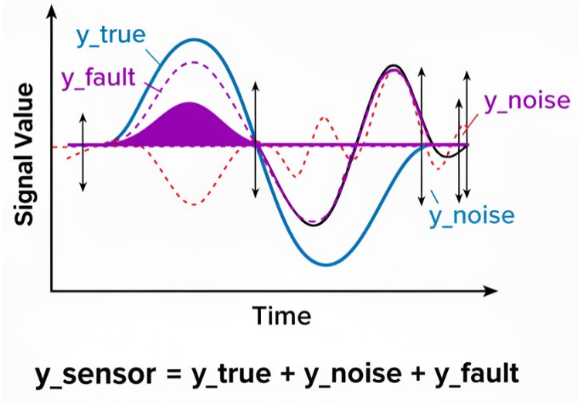

Figure 4.7: Nominal sensor measurement of a true signal ytrue corrupted by inherent random noise ynoise

This section describes the prevalent types of sensor faults considered in the context of this thesis. The readings from a sensor typically include a true physical value, measurement noise, and potentially a fault component [149]:

$$y_{\text{sensor}} = y_{\text{true}} + y_{\text{noise}} + y_{\text{fault}} \tag{4.24}$$

Where y_{true} is the actual physical value, y_{noise} represents the inherent random measurement noise (often modelled as Gaussian white noise), and y_{fault} is the additional component introduced by a fault.

4.3.2.1 Bias (Offset) Fault

A bias fault occurs when a sensor consistently outputs a reading that is offset by a fixed (or slowly varying) value from the true physical quantity. This can be caused by improper calibration, electronic component degradation, or environmental factors affecting the sensor's baseline. Bias faults are particularly insidious because they can appear as legitimate changes in the measured variable, leading the control system to react to non-existent conditions [150], *figure 4.8* below represent a Sensor measurement corrupted by a constant bias fault, showing a persistent offset from the true signal.

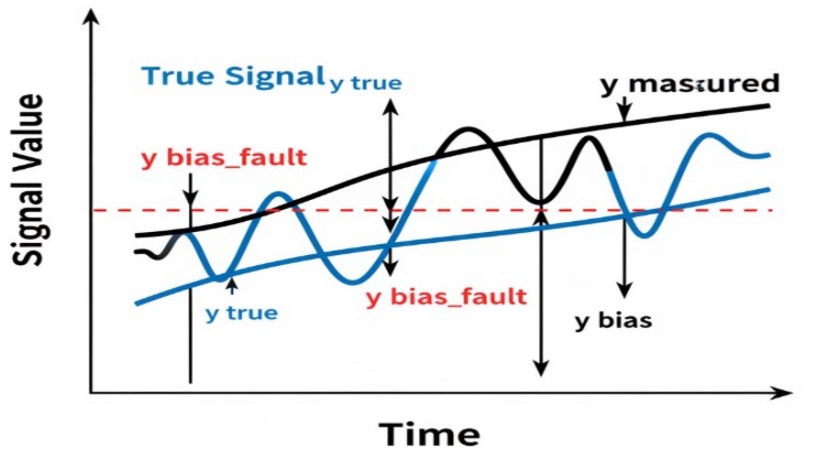

Figure 4.8: Sensor measurement corrupted by a constant bias fault.

For a sensor measuring quantity 'y', a bias fault 'b' is modelled as :

$$y_{\text{sensor}} = y_{\text{true}} + y_{\text{noise}} + b \tag{4.25}$$

Where 'b' is a constant or slowly varying offset.

Examples: A gyroscope consistently reporting a non-zero angular rate when the vehicle is stationary, or an accelerometer reading an offset from gravity along an axis. A barometer might report a constant altitude error.

Impact on Dynamics and Control: Bias faults directly corrupt the state estimation process.

- *Attitude Estimation Errors*: A biased gyroscope or accelerometer will lead to cumulative errors in attitude estimation (e.g., drift in roll, pitch, or yaw).
- *Position Estimation Errors*: A biased GPS or altimeter can cause the quadrotor to believe it is at an incorrect position or altitude, leading to *uncommanded manoeuvres.*
- *Uncommanded Control Actions*: The flight controller, relying on the faulty measurements, will generate incorrect control commands to "correct" for perceived errors that do not exist in reality, potentially causing oscillations, instability, or driving the system to its physical limits [151, 152].

4.3.2.2 Drift Fault

A drift fault is characterized by a gradual, continuous change in the sensor's output over time, independent of the true physical value. Unlike a fixed bias, drift indicates a progressive degradation. This type of fault is common in gyroscopes and magnetometers, where temperature changes or aging components can cause the baseline output to slowly shift [153]. *figure. 4.9* illustrates the characteristic progression of a drift fault, where the offset from the true signal gradually changes over time, leading to increasing errors .

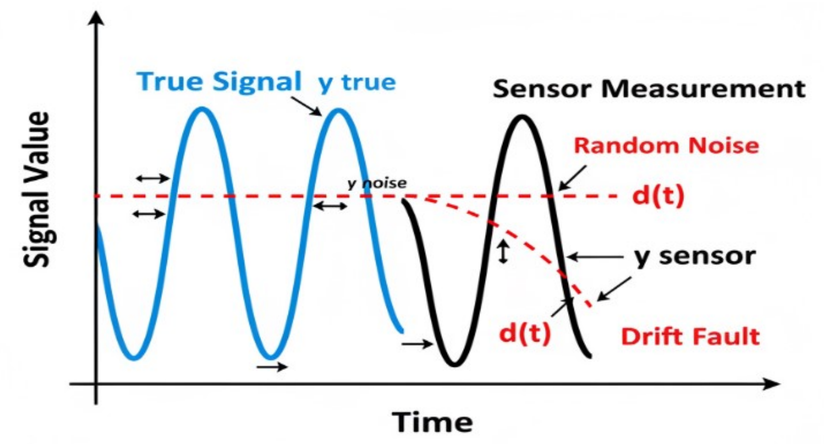

Figure 4.9: Sensor measurement exhibiting a drift fault.

A drift fault $d(t)$ can be modelled as a time-varying bias:

$$y_{\text{sensor}} = y_{\text{true}} + y_{\text{noise}} + d(t) \tag{4.26}$$

Where $d(t)$ is a function that changes gradually over time, often linearly or non-linearly.

Examples: A gyroscope's zero-rate output slowly increasing or decreasing, or a magnetometer's heading reference slowly wandering.

Impact on Dynamics and Control: Drift faults can be even more challenging than fixed biases because their effect is time-dependent and accumulative.

- *Accumulated Estimation Errors*: Continuous drift leads to progressively larger errors in state estimates, often resulting in significant deviations from the desired trajectory.
- *Aggressive Control Responses*: As the estimated error grows, the controller might attempt increasingly aggressive manoeuvres to compensate for a "phantom" state deviation.
- *Long-Term Instability*: If unaddressed, drift can eventually lead to such large estimation errors that the control system becomes unstable, especially during long-duration flights [154, 156].

4.3.2.3 Noise (Increased Variance) Fault

While all sensors exhibit some inherent noise, a fault can manifest as a significant increase in the variance or magnitude of this noise. This can be caused by electromagnetic interference, faulty wiring, component degradation, or environmental factors (e.g., strong vibrations causing IMU noise). This type of fault corrupts the signal-to-noise ratio, making the true signal harder to discern [155]. The visual impact of increased noise is presented in *figure. 4.10* which is characterized by significantly higher variance in the output compared to nominal operation, obscuring the true signal.

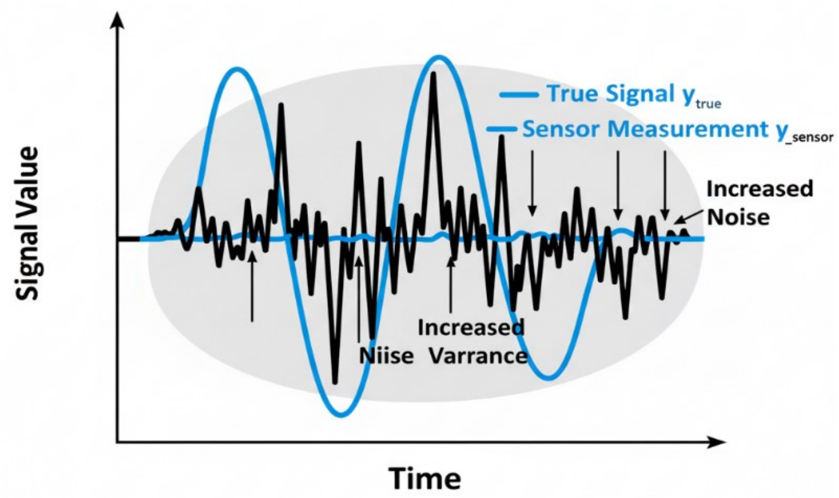

Figure 4.10: Sensor measurement corrupted by an increased noise fault.

For a sensor measuring quantity 'y', an increased noise fault is modelled by an increased variance σ^2_{fault} of the noise component :

$$y_{\text{sensor}} = y_{\text{true}} + N(0, \sigma^2_{\text{nominal}} + \sigma^2_{\text{fault}}) \tag{4.27}$$

Where $N(0, \sigma^2)$ represents Gaussian noise with zero mean and variance σ^2.

Impact on Dynamics and Control: Increased noise degrades the quality of sensor measurements, impacting filtering and estimation processes.

- *Degraded State Estimation*: Kalman filters or other state estimators will struggle to accurately estimate the quadrotor's state due to the high-frequency corruption, leading to noisy and less precise estimates.
- *Oscillatory Control Actions*: The controller may react to the noisy measurements by issuing rapid, oscillatory commands to the actuators, leading to "chattering" in control surfaces (or motor speeds) and increased wear, power consumption, and potentially exciting unmodeled dynamics.
- *Reduced Performance*: Overall tracking accuracy and stability margins will be reduced due to the uncertainty in state information [158, 159].

4.4 Fault-Tolerant Control System Design and Implementation

Building upon the dynamic model and fault scenarios established in the preceding sections, this chapter now transitions to the core challenge: the design and implementation of a robust fault-tolerant control system. This section presents a comprehensive framework for achieving resilience in the quadrotor's operation. It starts with a methodological discussion that justifies the selection of Sliding Mode Control (SMC) by comparing the strengths of conventional and AI-based approaches. Subsequently, a baseline PID controller is formulated to benchmark performance under nominal and faulty conditions. The primary focus is the rigorous development of the SMC-based FTC scheme, from its high-level architecture to the specific design of sliding surfaces and control laws. The section concludes by exploring prospective pathways for integrating artificial intelligence, thereby bridging the conventional method implemented herein with future research directions.

4.4.1 Justification of the Selected FTC Strategy for the UAV Quadrotor

In addressing the critical challenge of ensuring UAV quadrotor resilience, the choice of an FTC strategy must balance the trade-off between verifiable robustness and adaptive performance. For this investigation, we move beyond standard textbook implementations and propose a sophisticated, hybrid FTC architecture centred on an optimized Sliding Mode Controller. The justification for this specific design choice is rooted in its capacity to form a novel solution that synergistically combines the advantages of several techniques.

The foundational selection of Sliding Mode Control (SMC) is motivated by its proven robustness, a non-negotiable requirement for safety-critical aerial systems. However, our contribution lies in enhancing this foundation to overcome its practical limitations. We introduce a novel Enhanced Triple Power Reaching Law (ETPRL), chosen specifically for its documented superiority in accelerating state convergence and attenuating the high-frequency chattering that can compromise actuator health.

The most significant element justifying our approach is its hybrid philosophy of leveraging offline intelligence for rapid online fault adaptation. Instead of relying on computationally expensive online learning or adaptation, our method utilizes an innovative fault accommodation mechanism. This mechanism consists of a comprehensive library of controller parameters, each set being optimally tuned for a specific fault location and severity via Particle Swarm Optimization (PSO) offline. In real-time, an Extended Kalman Filter (EKF) performs precise fault detection and identification. Based on the EKF's estimation, the system executes an instantaneous and optimal control reconfiguration by selecting the appropriate parameters from the pre-computed library. This integrated strategy was deliberately chosen to deliver a robust, high-performance, and rapidly-acting FTC system, thereby establishing a powerful benchmark against which purely AI-driven solutions can be meaningfully evaluated.

4.4.2 FTC Design using Sliding Mode Control

Having established the performance limitations of the baseline PID controller under fault conditions, this section presents the core technical contribution of this chapter: the systematic design and development of a novel, hybrid Sliding Mode Control (SMC) based fault-tolerant control system. This approach is engineered not only to leverage the intrinsic robustness of

SMC but also to overcome its classical limitations through two primary innovations. First, a novel Enhanced Triple Power Reaching Law (ETPRL) is integrated into the control law to accelerate convergence and significantly attenuate chattering. Second, a unique fault accommodation paradigm is introduced, which combines an Extended Kalman Filter (EKF) for real-time fault diagnosis with a pre-computed library of controller parameters optimized offline via Particle Swarm Optimization (PSO) [159, 160]. The following subsections will detail the overall system architecture, the rigorous mathematical formulation of the sliding surfaces and the ETPRL-based control law, the integration of the FDD mechanism, and the resulting control reconfiguration strategy .

4.4.2.1 Overall Architecture of the SMC-Based Fault-Tolerant Control System

The proposed Fault-Tolerant Control (FTC) system is realized through a hybrid, multi-component architecture designed for rapid and precise response to actuator faults. The core philosophy is to synergize real-time fault diagnosis with a control strategy that reconfigures itself using pre-computed, optimal parameters. This approach circumvents the computational burden and potential convergence delays associated with purely online adaptive laws, making it highly suitable for safety-critical applications like UAVs [159].

The complete architecture, depicted in *figure. 4.11*, is composed of four primary, interacting subsystems:

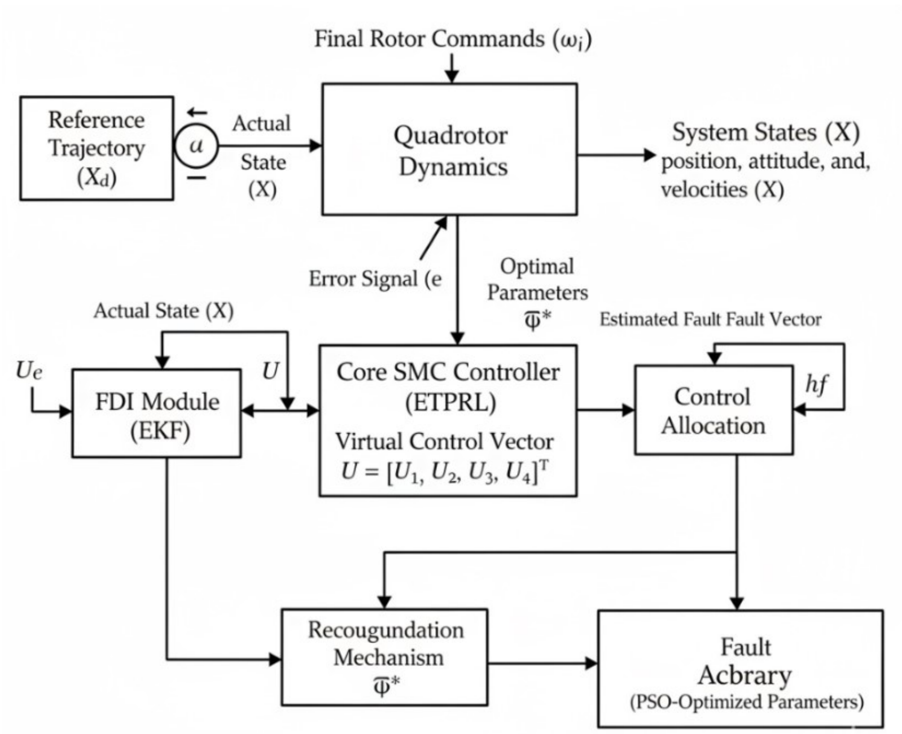

Figure 4.11: Overall Architecture of the Hybrid SMC-Based Fault-Tolerant Control System

1. *The Core SMC Controller*: The central control unit responsible for calculating the virtual control inputs (U_1, U_2, U_3, U_4) required to track the reference trajectory. Unlike a stan-

dard fixed-gain controller, its parameters are dynamically updated by the reconfiguration mechanism.

2. *The Fault Detection and Identification (FDI) Module*: An Extended Kalman Filter (EKF) that runs in parallel to the main controller. It processes the system's inputs (rotor commands) and outputs (sensor measurements) to provide a real-time estimate of the fault vector, $\hat{f}$, which contains the location and severity (e.g., Loss of Effectiveness percentage) of any actuator fault [157].

3. *The Fault Accommodation Library*: A pre-computed database or lookup table containing sets of optimal SMC parameters. Each parameter set in this library has been meticulously optimized offline using Particle Swarm Optimization (PSO) for a specific fault scenario (i.e., a specific fault location and severity).

4. *The Reconfiguration Mechanism*: This logic block acts as the bridge between the FDI module and the core controller. It takes the real-time fault estimate, $\hat{f}$, from the EKF and selects the corresponding optimal parameter set, Θ^*, from the Fault Accommodation Library. These parameters are then supplied to the Core SMC Controller, instantly tailoring its behaviour to the current health status of the quadrotor [158, 159].

A. Signal Flow and Operation:

During flight, the system operates as follows: The EKF continuously estimates the system's health. In a nominal state, $\hat{f}$ is zero, and the Reconfiguration Mechanism selects the baseline, non-faulty SMC parameters from the library. Upon the occurrence of a fault, the EKF detects, isolates, and quantifies it, updating $\hat{f}$ in real-time. The Reconfiguration Mechanism immediately uses this new $\hat{f}$ to index the accommodation library and retrieve the corresponding pre-optimized parameter set Θ^*. This new set is deployed to the SMC controller, which instantly adapts its control law to compensate for the fault, thereby maintaining stability and tracking performance [160].

The mathematical representation of the reconfiguration logic can be expressed as a mapping function, 'M', executed by the mechanism [156]:

$$M : \hat{f}(t) \to \Theta^*(t) \tag{4.28}$$

Where Θ^* is the set of optimal controller parameters (e.g., sliding surface gains, reaching law parameters) corresponding to the estimated fault $\hat{f}$. This event-driven, pre-optimized adaptation ensures a swift and precisely tailored response to system failures.

4.4.2.2 Design of Sliding Surfaces and Control Law

The design of the Sliding Mode Controller is a two-stage process. First, a sliding manifold is defined in the state space, which represents the desired system dynamics. Second, a control law is formulated to ensure the system states reach this manifold in finite time and remain on it thereafter, a condition that guarantees stability and tracking of the desired trajectory [162].

A. Sliding Surface Formulation:

The control objective is to drive the tracking error between the desired states 'x_d' and the actual states 'x' to zero. The state vector for control is defined as $x = [z, \phi, \theta, \psi]^T$. The tracking error vector is thus [163]:

$$e(t) = x_d(t) - x(t) \tag{4.29}$$

To ensure the convergence of this error, a set of linear sliding surfaces (or manifolds) is chosen, one for each controlled variable. The general form for a sliding surface 'S_i' corresponding to an error variable 'e_i' is [164]:

$$S_i(t) = \dot{e}_i(t) + C_i e_i(t) \tag{4.30}$$

where 'C_i' is a strictly positive design parameter that determines the rate of error convergence once the system is on the manifold ($S_i = 0$). A larger 'C_i' implies faster convergence. The sliding surfaces for the quadrotor's four degrees of freedom are therefore defined as [163]:

$$S_z = \dot{e}_z + C_z e_z \tag{4.31}$$

$$S_\phi = \dot{e}_\phi + C_\phi e_\phi \tag{4.32}$$

$$S_\theta = \dot{e}_\theta + C_\theta e_\theta \tag{4.33}$$

$$S_\psi = \dot{e}_\psi + C_\psi e_\psi \tag{4.34}$$

The objective of the control law is to enforce the condition ($S_i(t) = 0$) for all $i \in \{z, \phi, \theta, \psi\}$

B. SMC Control Law Derivation:

The standard SMC control law is composed of two parts: the equivalent control 'u_{eq}' and the discontinuous or reaching control 'u_{dis}' [163].

$$u(t) = u_{\text{eq}}(t) + u_{\text{dis}}(t) \tag{4.35}$$

The equivalent control, 'u_{eq}', is the continuous part of the control effort required to maintain the system state on the sliding surface once it has reached it ($\dot{S} = 0$). The discontinuous control, 'u_{dis}', is the component responsible for driving the system state towards the surface from any initial condition, ensuring the reaching condition is met [162].

B.1. Equivalent Control (u_{eq}): To derive the equivalent control, we use the invariance condition $\dot{S} = 0$. Differentiating the general sliding surface *Eq. 4.31* yields [159]:

$$\dot{S} = \ddot{e} + C\dot{e} = (\ddot{x}_d - \ddot{x}) + C\dot{e} \tag{4.36}$$

Substituting the general second-order nonlinear system dynamics $\ddot{x} = f(X) + g(X)u$, into *Eq. 4.36* gives [159]:

$$\dot{S} = \ddot{x}_d - [f(X) + g(X)u] + C\dot{e} \tag{4.37}$$

By setting $\dot{S} = 0$ and solving for the control input 'u', we obtain the equivalent control term, 'u_{eq}' [157]:

$$u_{\text{eq}}(t) = \frac{1}{g(X)} \left[\ddot{x}_d + C\dot{e} - f(X)\right] \tag{4.38}$$

Applying this formulation to the specific dynamics of the quadrotor (as derived in *Subsetion 4.2.3*), the equivalent control terms for altitude, roll, pitch, and yaw are [156]:

$$\begin{cases} u_{\text{eq},z}(t) = \dfrac{m}{C_\theta C_\phi}\left[\ddot{Z}_d + C_z\dot{e}_z - f_1(Z)\right] \\ u_{\text{eq},\phi}(t) = \dfrac{I_x}{l}\left[\ddot{\phi}_d + C_\phi\dot{e}_\phi - f_2(\phi)\right] \\ u_{\text{eq},\theta}(t) = \dfrac{I_y}{l}\left[\ddot{\theta}_d + C_\theta\dot{e}_\theta - f_3(\theta)\right] \\ u_{\text{eq},\psi}(t) = I_z\left[\ddot{\psi}_d + C_\psi\dot{e}_\psi - f_4(\Psi)\right] \end{cases} \tag{4.39}$$

Where f_1, f_2, f_3, f_4 represent the nonlinear terms and disturbances associated with each subsystem.

B.2. Discontinuous Control (u_{dis}) with Enhanced Triple Power Reaching Law (ETPRL): The discontinuous term is crucial for ensuring robustness and finite-time convergence to the sliding manifold. To overcome the limitations of conventional reaching laws, such as slow convergence and chattering, this work proposes a novel *Enhanced Triple Power Reaching Law (ETPRL)*. The reaching law is defined as [157]:

$$\dot{S} = -k_1|S|^\alpha \text{sgn}(S) - k_2|S|^\beta \text{sgn}(S) - k_3|S|^\gamma \text{sgn}(S) - k_4 S \tag{4.40}$$

Where k_1, k_2, k_3, k_4 are positive gains, and the exponents satisfy ($\alpha > 0$), ($0 < \beta < 1$), and ($\gamma > \alpha$).

This multi-stage structure synergistically combines the benefits of different power reaching laws :

- The term ($-k_1|S|^\alpha \text{sgn}(S)$) dominates when the state is far from the surface ($|S|$ is large), providing extremely rapid convergence.
- The term ($-k_2|S|^\beta \text{sgn}(S)$) governs the behavior during the intermediate phase.
- The term ($-k_3|S|^\gamma \text{sgn}(S)$) dominates when the state is close to the surface ($|S|$ is small), ensuring fast final convergence while mitigating chattering.
- The linear term ($-k_4 S$) provides continuous damping across the entire response [162,163].

The stability of this reaching law is proven via *Lyapunov analysis*, confirming that the condition $S\dot{S} \leq 0$ is satisfied, which guarantees the system states will reach the manifold in finite time. The discontinuous control effort, u_{dis}, is derived from this reaching law. The final forms for each control channel are [157, 165]:

$$u_{\text{dis},i}(t) = -k_{1,i}|S_i|^{\alpha_i}\text{sgn}(S_i) - k_{2,i}|S_i|^{\beta_i}\text{sgn}(S_i) - k_{3,i}|S_i|^{\gamma_i}\text{sgn}(S_i) - k_{4,i}S_i \tag{4.41}$$

For $i \in \{z, \phi, \theta, \psi\}$

Stability and Convergence Analysis

Théorème 4.1. *The states of the quadrotor system governed by the proposed Enhanced Triple Power Reaching Law (ETPRL) converge to the sliding manifold $S = 0$ within a fixed time and maintain asymptotic stability thereafter [165].*

Proof. The proof is conducted in two stages: first, the asymptotic stability is established via Lyapunov analysis; second, the finite-time convergence characteristic is derived through a phase-based analysis of the reaching law dynamics.

1) Lyapunov Stability Analysis To rigorously establish the stability of the closed-loop system and guarantee the reachability of the sliding manifold, we employ the direct method of Lyapunov. Consider the following positive-definite Lyapunov function candidate, V, representing the energy distance to the sliding surface [157]:

$$V = \frac{1}{2}S^2 \quad > 0, \quad \forall S \neq 0 \tag{4.42}$$

The time derivative of V along the system trajectories is given by [157]:

$$\dot{V} = \frac{d}{dt}\left(\frac{1}{2}S^2\right) = S\dot{S} \tag{4.43}$$

Substituting the discontinuous control dynamics defined by the ETPRL in Eq. 32 into the derivative yields [165]:

$$\dot{V} = S\left[-k_1|S|^{\alpha}\text{sgn}(S) - k_2|S|^{\beta}\text{sgn}(S) - k_3|S|^{\gamma}\text{sgn}(S) - k_4S\right] \tag{4.44}$$

Utilizing the property $x \cdot \text{sgn}(x) = |x|$, the expression is simplified by distributing S across the terms :

$$\dot{V} = -k_1|S|^{\alpha+1} - k_2|S|^{\beta+1} - k_3|S|^{\gamma+1} - k_4S^2 \tag{4.45}$$

Considering that the design gains k_1, k_2, k_3, k_4 are strictly positive, and the absolute magnitude $|S|$ is non-negative, the time derivative of the energy function satisfies [164]:

$$\dot{V} \leq 0 \tag{4.46}$$

Specifically, $\dot{V} = 0$ if and only if $S = 0$. This strictly negative definiteness ensures that the system trajectories are continuously driven towards the equilibrium point. Consequently, the sliding manifold is attractive, and the system is asymptotically stable.

2) Analysis of Fixed-Time Convergence To demonstrate the accelerated convergence properties, we analyze the reaching law behavior across three distinct phases based on the magnitude of the sliding variable $|S|$. For simplification, we assume symmetric gains $k_1 = k_2 = k_3 = k_4 = k$ and define the switching thresholds based on the initial state $|S_0|$. The convergence process is divided as follows [165]:

- **Phase I (Far Field):** When the system state is far from the manifold ($|S| > \omega_s$), the highest-order term dominates the dynamics. Since $\gamma > \alpha > \beta$, the term $|S|^\gamma$ provides the primary convergence force. The dynamics can be approximated as [157]:

$$\dot{S} \approx -k_3|S|^\gamma \mathrm{sgn}(S) - k_4 S \tag{4.47}$$

- **Phase II (Intermediate Field):** As the state approaches the vicinity of the manifold ($\omega_s \geq |S| > 1$), the influence of the highest power diminishes, and the term governed by α becomes the significant driver [157]:

$$\dot{S} \approx -k_1|S|^\alpha \mathrm{sgn}(S) - k_4 S \tag{4.48}$$

- **Phase III (Near Field):** In the final approach phase ($|S| \leq 1$), the fractional power term β ensures finite-time convergence without singularity, dominating the decay rate [163]:

$$\dot{S} \approx -k_2|S|^\beta \mathrm{sgn}(S) - k_4 S \tag{4.49}$$

By integrating the differential equations for each phase from the initial time t_0 to the final reaching time T, the evolution of the sliding surface can be expressed analytically [157]:

$$\begin{cases} S^{1-\gamma} = \left(S_0^{1-\gamma} + \dfrac{k_3}{k_4}\right) \exp^{-(1-\gamma)k_4 t} - \dfrac{k_3}{k_4} \\ S^{1-\alpha} = \left(1 + \dfrac{k_1}{k_4}\right) \exp^{-(1-\alpha)k_4 t} - \dfrac{k_1}{k_4} \\ S^{1-\beta} = \left(1 + \dfrac{k_2}{k_4}\right) \exp^{-(1-\beta)k_4 t} - \dfrac{k_2}{k_4} \end{cases} \tag{4.50}$$

Consequently, the upper bound of the convergence time for each stage (t_1, t_2, t_3) is derived as:

$$\begin{cases} t_1 = \dfrac{1}{(1-\gamma)k_4}\left[\ln\left(\gamma^{1-\gamma} + \dfrac{k_3}{k_4}\right) - \ln\left(S_0^{1-\gamma} + \dfrac{k_3}{k_4}\right)\right] \\ t_2 = \dfrac{1}{(1-\alpha)k_4}\left[\ln\left(1 + \dfrac{k_1}{k_4}\right) - \ln\left(\omega_s^{1-\alpha} + \dfrac{k_1}{k_4}\right)\right] \\ t_3 = \dfrac{1}{(\beta-1)k_4}\left[\ln\left(S^{1-\beta} + \dfrac{k_2}{k_4}\right) - \ln\left(1 + \dfrac{k_2}{k_4}\right)\right] \end{cases} \tag{4.51}$$

The total reaching time T required for the system states to converge from an arbitrary initial condition S_0 to the equilibrium is strictly bounded by the sum of the time intervals of these three phases [165]:

$$T < t_1 + t_2 + t_3 \tag{4.52}$$

This confirms that the proposed ETPRL guarantees finite-time convergence regardless of the initial system states. □

B.3. Final Control Law Synthesis: By combining the equivalent and discontinuous components, the final robust fault-tolerant control laws for the quadrotor's virtual inputs (U_1, U_2, U_3, U_4) are synthesized. For each channel 'i', the total control input is [157]:

$$U_i(t) = u_{\text{eq},i}(t) + u_{\text{dis},i}(t) \tag{4.53}$$

This formulation results in a robust and high-performance control law capable of forcing the system states to the specified sliding manifold and maintaining them there, even in the presence of uncertainties and faults, with an accelerated convergence rate and attenuated chattering due to the novel ETPRL. The parameters $\{C_i, k_{1,i}, k_{4,i}, \alpha_i, \beta_i, \gamma_i\}$ are the subject of the optimization and reconfiguration strategy discussed in the subsequent sections.

4.4.2.3 EKF-Based Fault Detection and Identification (FDI) Module

The efficacy of the proposed hybrid FTC architecture is critically dependent on the real-time delivery of accurate diagnostic information. This function is performed by a dedicated Fault Detection and Identification (FDI) module. The implemented FDI strategy is a multi-stage process that leverages the state estimation capabilities of an Extended Kalman Filter (EKF) to drive a logic-based fault analysis framework. The process is systematically designed to first detect and locate the fault, and then to estimate its magnitude.

A. State Estimation via Extended Kalman Filter (EKF):

The foundation of the FDI module is an EKF tasked with providing accurate, real-time estimates of the quadrotor's states from noisy sensor measurements. The EKF is a recursive algorithm that operates in two phases: prediction and correction [58].

A.1. Prediction Phase: The EKF projects the current state and error covariance estimates forward to obtain *a priori* estimates for the next time step [59].

$$\hat{X}_{k+1|k} = f(\hat{X}_{k|k}, U_k) \tag{4.54}$$
$$P_{k+1|k} = A * P_{k|k} * A^T + Q_k \tag{4.55}$$

A.2. Correction Phase: The EKF incorporates the new measurement (Y_{k+1}) to refine the *a priori* prediction into a more accurate *a posteriori* estimate [60].

$$K_{k+1} = P_{k+1|k} * H_{k+1}^T * \left(R_k + H_{k+1} * P_{k+1|k} * H_{k+1}^T\right)^{-1} \tag{4.56}$$
$$\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1} * \left[Y_{k+1} - \hat{Y}_{k+1|k}\right] \tag{4.57}$$
$$P_{k+1|k+1} = (I - K_{k+1} * H_{k+1}) * P_{k+1|k} * (I - K_{k+1} * H_{k+1})^T + K_{k+1} * R_k * K_{k+1}^T \tag{4.58}$$

Crucially, the primary output from the EKF for the FDI logic is the vector of estimated attitude angles, $[\hat{\phi}, \hat{\theta}, \hat{\psi}]^T$, which provides a clean signal essential for the subsequent diagnostic steps.

B. Fault Detection and Localization:

Fault detection is achieved by generating and monitoring a specialized *Fault Detection Signal (FDS)*. This signal is formulated based on the error between the EKF's estimated attitude 'i_{est}' and the desired attitude 'i_{des}' from the controller. For each attitude channel $i \in \{\phi, \theta, \psi\}$, the FDS is defined as [157]:

$$\text{FDS}_i = k_i(\dot{e}_i + c_i e_i) \tag{4.59}$$

Where '$e_i = i_{\text{est}} - i_{\text{des}}$', and '$k_i$' and '$c_i$' are tuning gains. A fault is formally detected if the absolute value of any FDS exceeds a predetermined threshold, 'T_i' :

$$|\text{FDS}_i| \geq T_i \Longrightarrow \text{faulty condition} \tag{4.60}$$

Once a fault is detected, its location is isolated by exploiting the unique signature of an actuator fault on the quadrotor's roll and yaw dynamics. The specific faulty actuator is pinpointed by analysing the signs (positive or negative variation) of the 'FDS_ϕ' and 'FDS_ψ' signals [157]. The logic for this localization is summarized in *Table 4.1*.

Table 4.1: *Logic of fault location*

Fault location	**Fault effect (FDS_ϕ variation)**	**Fault effect (FDS_ψ variation)**
Actuator 1	Negative FDS_ϕ variation	Positive FDS_ψ variation
Actuator 2	Positive FDS_ϕ variation	Negative FDS_ψ variation
Actuator 3	Positive FDS_ϕ variation	Positive FDS_ψ variation
Actuator 4	Negative FDS_ϕ variation	Negative FDS_ψ variation

C. Fault Severity Estimation:

Following detection and localization, the final step is to quantify the fault's magnitude, specifically the percentage of Loss of Effectiveness (LOE). This is accomplished using a pre-established look-up table that maps the magnitude of the yaw fault detection signal, $|\text{FDS}_\psi|$, to a corresponding LOE percentage [158].

This look-up table is generated offline through extensive simulation and tuning, where known LOE faults are injected and the resulting steady-state 'FDS_ψ' value is recorded. This creates a reliable mapping between the diagnostic signal and the physical fault severity.

The final output of the complete FDI module is an estimated fault vector, $\hat{f}$, containing the precise location (actuator 1-4) and severity (LOE %) of the anomaly. This vector is then passed to the Control Reconfiguration Strategy, providing the necessary information for a targeted and optimal response.

4.4.2.4 Control Reconfiguration and Allocation Strategy

Building upon the diagnostic information provided by the FDI module, the final stage of the FTC system is the control reconfiguration. The primary role of this strategy is to actively adjust the controller's output signals to compensate for the estimated Loss of Effectiveness (LOE), thereby maintaining vehicle stability and ensuring the quadrotor continues to follow its desired trajectory. The implemented strategy is based on an additive fault compensation approach, where the estimated control loss is explicitly reconstructed and added back to the control command [98].

A. Modelling the Faulty Actuator Dynamics: In the presence of an actuator fault, the quadrotor's dynamics are altered. The actual control input, '$U_{\text{act}}(t)$', that is effectively applied to the airframe is no longer equal to the optimal control, '$U_{\text{opt}}(t)$', computed by the SMC law. Instead, it is a degraded signal, which can be modelled as the optimal command minus a "control loss" term, '$U_f(t)$', that represents the effect of the LOE [147]:

$$U_{\text{act}}(t) = U_{\text{opt}}(t) - U_f(t) \tag{4.61}$$

The core challenge for the reconfiguration unit is to calculate a new, compensated control command, '$U_{\text{reconfigured}}$', such that it counteracts the effect of '$U_f(t)$'.

B. Quantifying and Reconstructing the Control Loss: The key to this strategy lies in accurately quantifying the control loss, '$U_f(t)$', using the output from the FDI module. The FDI provides the estimated fault amplitude vector, $\hat{f} = [\hat{f}_1, \hat{f}_2, \hat{f}_3, \hat{f}_4]^T$, where each element $\hat{f}_i$ corresponds to the estimated LOE percentage of the i-th rotor [148].

The control loss vector $U_f(t)$ can be expressed as a function of this fault vector and the current rotor speeds, Ω_i. This relationship is derived from the fundamental control allocation principles [160]:

$$U_f = \begin{bmatrix} u_{f,1} \\ u_{f,2} \\ u_{f,3} \\ u_{f,4} \end{bmatrix} = \begin{bmatrix} \hat{f}_1 b & \hat{f}_2 b & \hat{f}_3 b & \hat{f}_4 b \\ 0 & -\hat{f}_2 bl & 0 & \hat{f}_4 bl \\ \hat{f}_1 bl & 0 & -\hat{f}_3 bl & 0 \\ -\hat{f}_1 d & \hat{f}_2 d & -\hat{f}_3 d & \hat{f}_4 d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{4.62}$$

This equation provides a direct, model-based estimation of the reduction in total thrust ($U_{f,1}$) and the parasitic torques ($U_{f,2}, U_{f,3}, U_{f,4}$) caused by the fault.

C. The Additive Reconfiguration Law: The reconfiguration law is formulated to reconstruct the desired control signals by adding the estimated loss back to the actual (degraded) control signal. Therefore, the new, reconfigured control vector, $U_{\text{reconfigured}}$, is calculated as [163]:

$$U_{\text{reconfigured}}(t) = U_{\text{act}}(t) + U_f(t) \tag{4.63}$$

By substituting *Eq. 4.51* into *Eq. 4.50*, it becomes clear that this strategy aims to restore the original, optimal control effort :

$$U_{\text{reconfigured}}(t) \approx U_{\text{opt}}(t) \tag{4.64}$$

In practice, the implementation involves directly computing the reconstructed virtual control signals, $[U_1, U_2, U_3, U_4]^T_{\text{reconfigured}}$, on a component-wise basis [157]:

$$\begin{cases} U_{1,\text{reconf}} = U_{\text{act},1} + b(\hat{f}_1\Omega_1^2 + \hat{f}_2\Omega_2^2 + \hat{f}_3\Omega_3^2 + \hat{f}_4\Omega_4^2) \\ U_{2,\text{reconf}} = U_{\text{act},2} + bl(\hat{f}_4\Omega_4^2 - \hat{f}_2\Omega_2^2) \\ U_{3,\text{reconf}} = U_{\text{act},3} + bl(\hat{f}_1\Omega_1^2 - \hat{f}_3\Omega_3^2) \\ U_{4,\text{reconf}} = U_{\text{act},4} + d(-\hat{f}_1\Omega_1^2 + \hat{f}_2\Omega_2^2 - \hat{f}_3\Omega_3^2 + \hat{f}_4\Omega_4^2) \end{cases} \tag{4.65}$$

This reconfigured control vector, '$U_{\text{reconfigured}}$', replaces the original SMC output and is then passed to the control allocation block to calculate the final, compensated motor speed commands. This closes the FTC loop, turning the fault diagnosis into a direct, corrective control action that maintains the quadrotor's desired path under fault.

4.4.2.5 Tuning Sliding Mode Controller Parameters

The efficacy of the proposed Hybrid Fault-Tolerant Sliding Mode Control (HFTC) strategy is critically dependent on the precise selection of numerous controller gains and exponents. Due to the high number of variables 32 in total, each independently affecting the quadrotor's behaviour—and the need to guarantee optimal performance across diverse flight conditions, a stochastic optimization algorithm was adopted for parameter tuning.

A. Particle Swarm Optimization (PSO) Algorithm: Particle Swarm Optimization (PSO) is a class of metaheuristic algorithms, inspired by the collective intelligence resulting from the social behaviour of bird flocks and schooling fish [166]. This algorithm was selected for its proven ability to solve complex, multi-variable optimization problems efficiently. The movement and updating of a particle's position $x_i(t)$ and velocity $v_i(t)$ are governed by the following equations [167]:

$$\begin{cases} v_i(t+1) = \omega v_i(t) + c_1 r_1[P_{\text{best},i}(t) - x_i(t)] + c_2 r_2[G(t) - x_i(t)] \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \tag{4.66}$$

The update rules for the particle's personal best position, $P_{\text{best},i}(t)$, and the whole swarm's best position, $G(t)$, are defined as [168]:

$$P_{\text{best},i}(t+1) = \begin{cases} x_i(t+1) & \text{if } J(x_i(t+1)) < J(P_{\text{best},i}(t)) \\ P_{\text{best},i}(t) & \text{else} \end{cases} \tag{4.67}$$

$$G(t) = \min_{i=1,2,3,\ldots,N} J(P_{\text{best},i}(t)) \tag{4.68}$$

Where 'ω' is the inertia weight, 'c_1' and 'c_2' are the cognitive and social parameters, and 'r_1' and 'r_2' are random coefficients distributed uniformly in the range of.

The specific parameters utilized for the PSO algorithm are summarized in *Table 4.2*.

Table 4.2: *Parameters of the PSO algorithm*

Description	Parameter	Value
Population size	n	150
Number of iterations	M_{max}	400
Cognitive parameter	c_1	1.2
Social parameter	c_2	1.4
random coefficients	r_1 & r_2	0.75
Inertia weight	ω	0.4

B. Fitness Function and Optimization Strategy: The adopted fitness function, 'J', for evaluating the algorithm's performance and guiding the search for optimal SMC parameters, is an integral objective function that minimizes a combination of tracking error and control effort [169]:

$$J = \int_0^T \left(\lambda_1 \|e(t)\|^2 + \lambda_2 \|u(t)\|^2\right) dt \tag{4.69}$$

where $e(t)$ denotes the tracking error; $u(t)$ is the control signal; 'λ_1' and 'λ_2' are weighting factors; and 'T' is the time at which the response settles at a steady state. The term '$\lambda_2\|u(t)\|^2$' represents the penalized high control effort, encouraging the controller to use less aggressive actions, thereby reducing chattering and actuator wear.

C. Offline Pre-computation of Fault Accommodation Data: The proposed hybrid strategy for fault accommodation is based on the robust combination of offline optimization and online adaptation. This process is structured in three phases: the generation of the Fault Accommodation Library, the definition of nominal operation parameters, and the online reconfiguration mechanism [166].

The overall procedure for generating this Bank of Parameters is detailed in *Figure 4.12*.

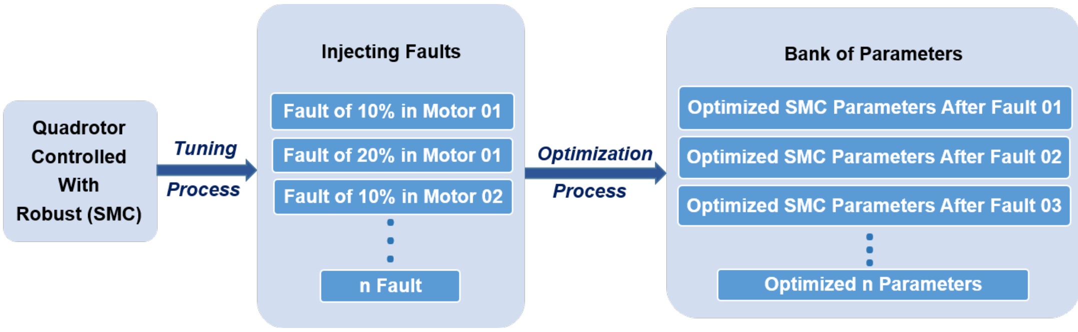

Figure 4.12: Steps of generating the bank of parameters [157]

A. Parameter Bank Generation (Offline): A comprehensive set of SMC parameters (including the gains and exponents for the ETPRL) is optimized offline for multiple fault magnitudes (e.g., 10%, 20%, 30% ...) across different actuators (Motor 01, Motor 02, etc.). The process, illustrated in *Figure 4.12*, involves systematically injecting a fault, running the PSO algorithm with the objective function, and determining the optimal parameter set that compensates for that specific fault. Each optimized parameter set is then memorized and indexed in a "Bank of Parameters." This pre-computation, while resource-intensive, ensures rapid and precisely tailored control reconfiguration during flight, thus circumventing the computational burden of a purely online adaptive fault-tolerant control system [157].

B. Nominal Operation Parameters: The baseline parameters for the HFTC-SMC, required when the quadrotor is operating under nominal (fault-free) conditions, were determined using the PSO algorithm and fitness function (4.5) configured to maximize performance in the absence of faults. These 32 baseline parameters are utilized until the Extended Kalman Filter (EKF) in the FDI module detects and identifies a fault [157]. *Table 4.3* presents the complete set of optimized parameters for the nominal flight scenario.

Table 4.3: *Optimized parameters of the HFTC-SMC with ETPRL for nominal (fault-free) operation*

Parameter	Value	Parameter	Value	Parameter	Value
$k_{1,z}$	5.582	$k_{3,z}$	3.417	α_z	1.563
$k_{1,\phi}$	29.515	$k_{3,\phi}$	30.251	α_ϕ	1.534
$k_{1,\theta}$	23.251	$k_{3,\theta}$	33.214	α_θ	1.601
$k_{1,\psi}$	16.152	$k_{3,\psi}$	19.558	α_ψ	1.582
$k_{2,z}$	9.252	$k_{4,z}$	0.514	β_z	0.432
$k_{2,\phi}$	16.151	$k_{4,\phi}$	13.548	β_ϕ	0.478
$k_{2,\theta}$	17.518	$k_{4,\theta}$	9.554	β_θ	0.498
$k_{2,\psi}$	8.515	$k_{4,\psi}$	14.597	β_ψ	0.462
γ_z	1.752	C_z	33.25		
γ_ϕ	1.764	C_ϕ	38.512		
γ_θ	1.695	C_θ	30.144		
γ_ψ	1.735	C_ψ	29.269		

C. Online Reconfiguration (During Flight): During flight, whenever an actuator fault occurs, the EKF-based Fault Detection and Identification (FDI) unit estimates the fault's location and amplitude. The system then selects the corresponding pre-optimized parameter set from the

"Bank of Parameters," ensuring a swift and precisely tailored control reconfiguration [170].

The final step of the hybrid FTC strategy occurs in real time. Whenever an actuator enters a Loss of Effectiveness fault situation, the EKF-based Fault Detection and Identification (FDI) unit estimates the fault's location and amplitude $\hat{f}$. The Reconfiguration Mechanism then swiftly and precisely selects the corresponding pre-optimized parameter set (Θ^*) from the "Bank of Parameters" (as outlined in the overall architecture in *Figure 4.11*). This event-driven selection instantly tailors the core SMC's behaviour to the quadrotor's compromised health status, ensuring continued stability and tracking performance [157].

4.4.3 Discussion on the Potential for AI Integration

The preceding sections have detailed the design of a highly effective, model-based Hybrid Fault-Tolerant Control (HFTC) system. The proposed architecture, leveraging an optimized SMC with a novel ETPRL, an EKF-based FDI module, and a pre-computed fault accommodation library, demonstrates significant robustness and high performance in simulation. However, the very structure of this conventional approach reveals certain inherent limitations which, in turn, open promising avenues for the integration of Artificial Intelligence, aligning with the central inquiry of this thesis.

The primary limitations of the implemented HFTC include:

- *Discrete Fault Accommodation:* The system relies on a pre-computed library of parameters for discrete fault severities. It may not exhibit optimal performance for fault magnitudes that fall between these pre-optimized points.

- *Dependence on Model Accuracy:* Both the SMC's equivalent control law and the EKF's estimation capabilities are fundamentally dependent on the fidelity of the quadrotor's dynamic model. Significant unmodeled dynamics or parametric drift could degrade performance.

- *Intensive Offline Optimization:* The generation of the fault accommodation library via PSO is a computationally expensive, non-trivial offline process that must be repeated if the UAV platform is modified.

AI-based techniques offer powerful solutions to address these specific challenges. This discussion will explore two primary pathways for this integration: first, augmenting the existing SMC framework with AI components for enhanced adaptivity, and second, conceptualizing a fully AI-driven FTC architecture as a direction for future work [171], [172].

4.4.3.1 Augmenting SMC with AI (Creating a Hybrid Intelligent System)

Rather than replacing the robust SMC framework, AI can be strategically integrated to enhance its capabilities, leading to a more adaptive and intelligent hybrid system.

- *Intelligent Parameter Interpolation:* A significant enhancement would be to replace simple look-up table interpolation with an AI-based function approximator. A small, well-trained Neural Network (NN) could learn the complex, non-linear mapping from the FDI's fault estimate, f , to the full set of optimal SMC parameters, . This would allow for a continuous and smooth adaptation of the controller gains for any fault severity, not just the discrete points optimized offline, thereby improving the control performance for the entire fault spectrum.

- *Adaptive Reaching Law using Fuzzy Logic:* The gains of the ETPRL (k1,...,k4) are currently fixed. A Fuzzy Logic System (FLS) could be designed to adapt these gains online. By defining fuzzy rules based on the magnitude of the sliding surface, S, and its derivative, an FLS could intelligently tune the reaching law. For instance, a rule could be: "IF S is large, THEN increase the gain k3." This would make the reaching phase more aggressive when far from the manifold and more gentle when close, further optimizing the trade-off between convergence speed and chattering [108].

- *NN-Based Uncertainty and Disturbance Observer:* To address the system's model dependency, a Radial Basis Function Neural Network (RBFNN) could be implemented as an online uncertainty observer. The RBFNN would be trained to estimate the lumped uncertainty, which includes unmodeled dynamics, parameter variations, and external disturbances. This estimate would then be fed into the SMC control law as a cancellation term, significantly improving the controller's robustness and its ability to adapt to changing flight conditions (e.g., payload variations, aerodynamic changes) without requiring a perfect a priori model [132].

4.4.3.2 Conceptual Framework for a Purely AI-Based FTC

A more radical and forward-looking approach involves replacing the entire model-based SMC architecture with a purely AI-driven control system. Deep Reinforcement Learning (DRL) stands out as the most promising candidate for such an endeavour.

A DRL-based FTC agent could learn a robust control policy directly from interaction with a high-fidelity simulation of the quadrotor. The conceptual framework would be as follows:

- *The Agent:* A deep neural network that functions as the end-to-end controller.

- *State Space:* The agent's input would be a vector of the quadrotor's sensor readings (e.g., position, velocity, attitude, angular rates) and potentially the fault estimates from the FDI module.

- *Action Space:* The agent's output would be the direct motor commands for the four rotors.

- *Reward Function:* The core of the design would be a meticulously crafted reward function that penalizes trajectory deviations, excessive control effort, and instability, while rewarding smooth and stable flight, even after a fault is introduced [131].

The DRL agent would be trained over millions of episodes, experiencing a vast and continuous range of actuator fault scenarios and external disturbances. The potential advantages of such an approach are immense: it could develop highly complex and non-intuitive control strategies that are inherently robust and adaptive without relying on an explicit mathematical model. However, significant challenges remain, primarily concerning safety verification, the sim-to-real transfer gap, and the extensive computational cost of training. Nonetheless, this paradigm represents a key research frontier and a logical next step in the pursuit of truly autonomous and resilient flight control.

4.5 Simulation and Analysis

In this section, the performance and effectiveness of the proposed Hybrid Fault-Tolerant Control (HFTC) strategy are rigorously evaluated. To validate the theoretical contributions presented in the previous sections, specifically the efficacy of the Enhanced Triple Power Reaching Law (ETPRL) within a Sliding Mode Control framework, extensive simulations were conducted using the MATLAB/Simulink platform.

The primary objective of this analysis is to demonstrate the quadrotor's capability to maintain stability and precise trajectory tracking despite the presence of severe actuator faults and external disturbances. The proposed control scheme, optimized via Particle Swarm Optimization (PSO), is subjected to distinct flight scenarios designed to challenge the system's manoeuvrability and fault-tolerance mechanisms.

To provide a comprehensive assessment, the simulation is divided into two main test cases involving different path shapes and fault sequences:

- *The First Test*: Focuses on a helical trajectory subject to three successive Loss of Effectiveness (LOE) faults.
- *The Second Test*: Focuses on a square trajectory subject to two successive LOE faults.

Furthermore, to highlight the superiority of the proposed approach, the results obtained using the ETPRL are compared against conventional reaching laws, including the *double power*, *standard power*, and *constant reaching laws*. This comparative analysis serves to verify the proposed method's improvements regarding convergence speed, chattering attenuation, and robustness under critical fault conditions.

4.5.1 The 1st Test (Helical path)

The first simulation scenario is designed to evaluate the tracking performance and stability of the quadrotor while executing a complex 3D manoeuvre. In this test, the UAV is commanded to follow a helical trajectory over a total simulation duration of 60 seconds. This trajectory was selected because it requires simultaneous coordination of the 'x', 'y', and 'z' axes, thereby effectively coupling the translational and rotational dynamics.

To rigorously assess the fault-tolerant capabilities of the proposed controller, a sequence of three successive Loss of Effectiveness (LOE) faults is injected into different motors. The fault scenario is designed to simulate a progressive degradation of the propulsion system, ranging from minor to critical severity. The timeline of these fault injections is defined as follows [157]:

- *Phase 1 (Minor Fault)*: At ($t = 15s$), a partial loss of effectiveness of 20% is introduced in Motor 1.
- *Phase 2 (Moderate Fault)*: At ($t = 30s$), a secondary fault occurs in Motor 2 with a magnitude of 35%, adding to the system's stress.
- *Phase 3 (Severe Fault)*: At ($t = 45s$), a critical fault is injected into Motor 3 with a significantly high amplitude of 55%.

This testing protocol, involving multi-actuator failures with increasing severity, serves to verify the ETPRL-based controller's ability to reconfigure control allocation and suppress chattering under rapidly changing dynamic constraints.

The figures below represents this test's results:

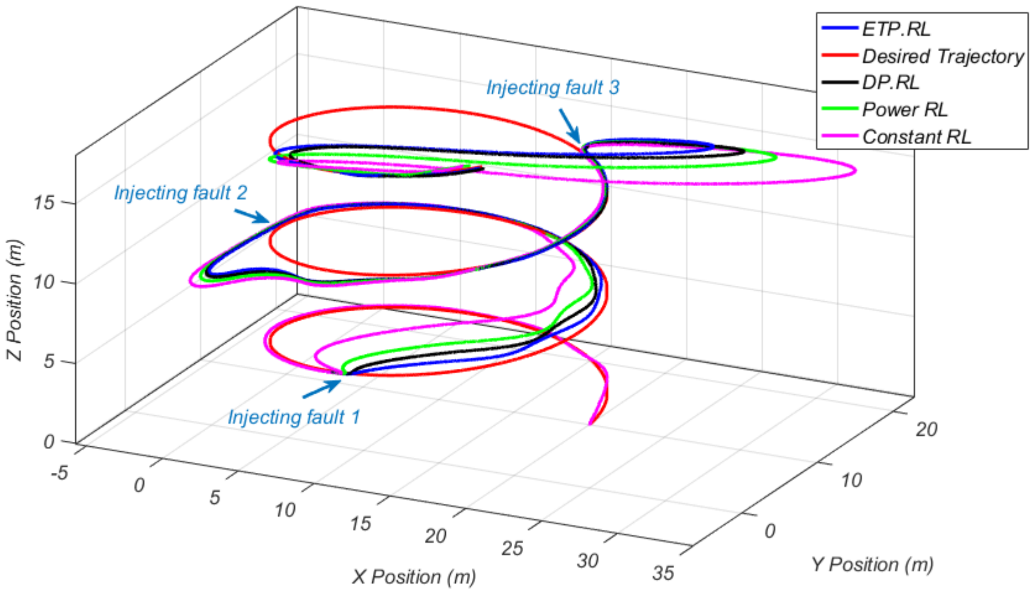

Figure 4.13: The followed helix trajectory during test 01

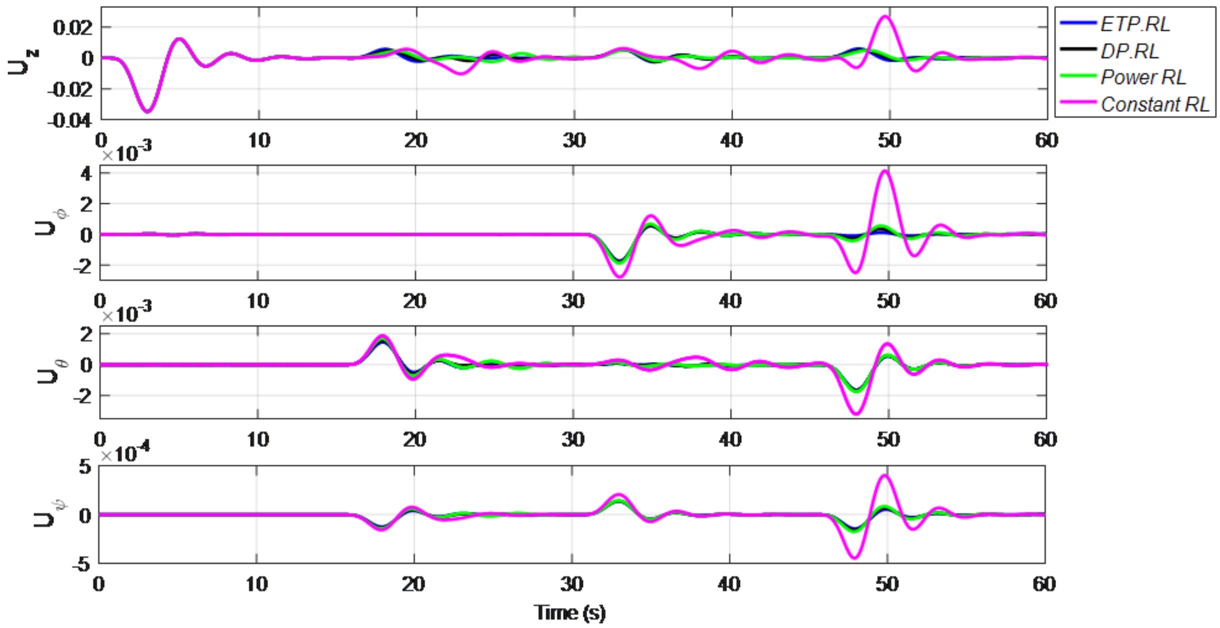

Figure 4.14: Variation of control signals during Test 01

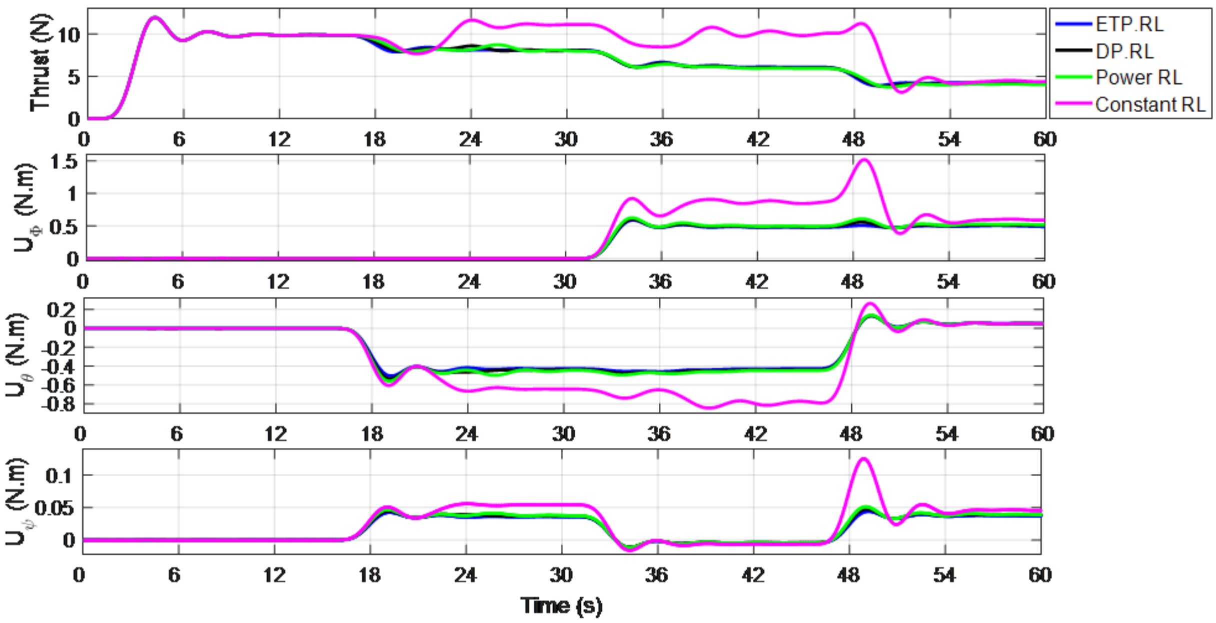

Figure 4.15: Applied torques during test 01

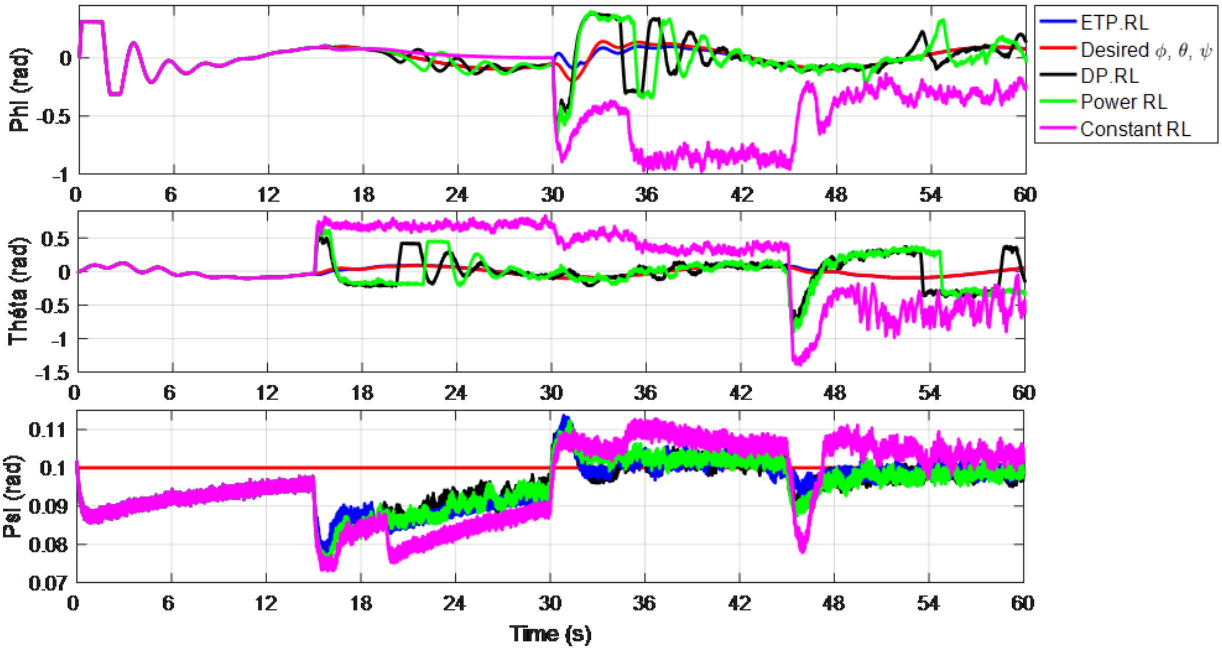

Figure 4.16: Evolution of roll, pitch and yaw angles during test 01

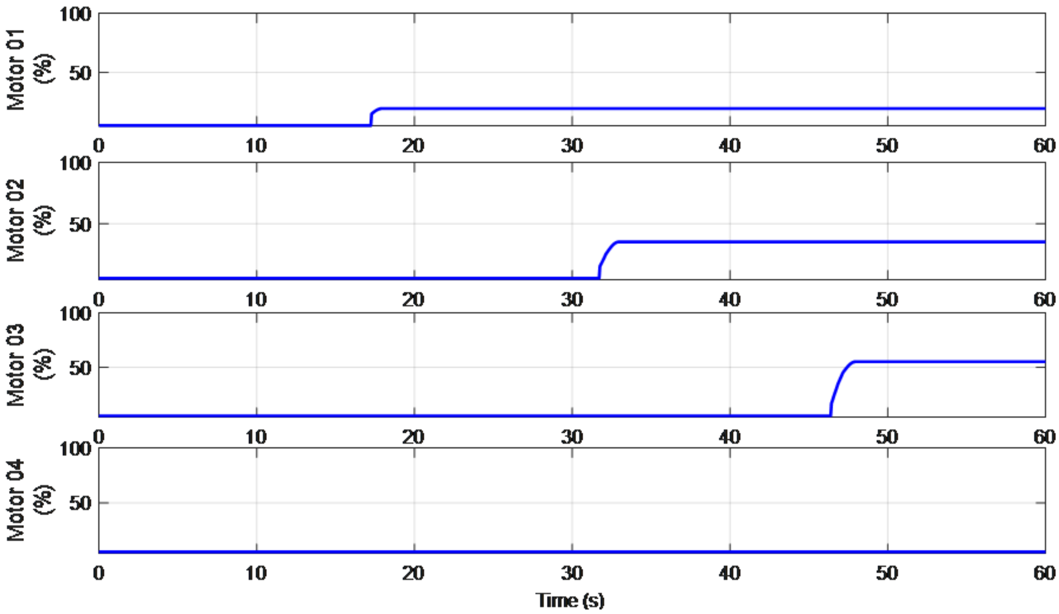

Figure 4.17: Faults estimated amplitude by the FDI unit

4.5.1.1 Performance Metrics and Discussion

The graphical results from Test 01, involving a helical trajectory under progressive and cumulative Loss of Effectiveness (LOE) faults, provide the empirical foundation for validating the superior performance of the proposed Hybrid Fault-Tolerant Sliding Mode Controller (HFTC) utilizing the Enhanced Triple Power Reaching Law (ETPRL). The following analysis integrates quantitative and qualitative performance metrics with the visual evidence presented in Figures 12 through 16, comparing the ETPRL (blue) against the DPRL (black), Power RL (green), and Constant RL (magenta) benchmarks.

A. Quantitative Performance Assessment:

The efficacy of the ETPRL is rigorously assessed using the estimated peak trajectory deviation *Table 4.4* and the Integrated Absolute Error (IAE) for both tracking accuracy and control effort *Table 4.5*.

Table 4.4: Estimated trajectory tracking performance metrics during test 01 (helix)

Controller	**Peak Deviation after fault 01 (m)**	**Peak Deviation after fault 02 (m)**	**Peak Deviation after fault 03 (m)**
ETPRL	0.851	4.013	8.573
DPRL	1.345	4.652	10.513
Power RL	2.093	5.265	12.843
Constant RL	3.788	6.628	18.093

Table 4.5: Comparison of Tracking and Control Effort using Integrated Absolute Error (IAE)

Fault	**Reaching Law**	$\int_0^{t_f} \lvert\phi_d - \phi\rvert dt$	$\int_0^{t_f} \lvert\theta_d - \theta\rvert dt$	$\int_0^{t_f} \lvert\psi_d - \psi\rvert dt$	$\int_0^{t_f} (\Omega_1 + \Omega_2 + \Omega_3 + \Omega_4) dt$
20%	ETPRL	0.0180	0.0150	0.113	1.0645e6
	DPRL	0.036	0.0213	0.237	1.0879e6
	Power RL	0.049	0.0342	0.291	1.0938e6
	Constant RL	0.092	0.0556	0.342	1.152e6
35%	ETPRL	0.0214	0.0191	0.2134	1.0698e6
	DPRL	0.0484	0.0285	0.3146	1.0924e6
	Power RL	0.0585	0.0432	0.3841	1.0985e6
	Constant RL	0.1013	0.0821	0.5493	1.1595e6
55%	ETPRL	0.0289	0.0264	0.2951	1.0788e6
	DPRL	0.0521	0.0365	0.3815	1.1452e6
	Power RL	0.0632	0.0491	0.4695	1.1952e6
	Constant RL	0.1568	0.1389	0.8815	1.3351e6

The quantitative data provides conclusive evidence of the ETPRL's superior performance:

- *Tracking Accuracy and Robustness (Table 4.4)*: The ETPRL consistently achieves the lowest *Peak Deviation* across all fault scenarios. After the most severe 55% LOE fault, the ETPRL's maximum deviation (8.573m) is significantly better than the DPRL (10.513m) and almost half that of the Constant RL (18.093m). This robust performance confirms its enhanced fault-tolerant capability under cumulative stress.

- *Precision and Efficiency (Table 4.5)*: For every tested fault level and axis, the ETPRL records the lowest *IAE* values, confirming the highest tracking precision. Furthermore, this superior tracking is achieved with the *lowest integrated total rotor speed* (IAE for control effort) across all fault conditions. This signifies a more efficient control solution that provides high performance while reducing the physical demand on the actuators.

B. Qualitative and Control Characteristic Assessment

The tables below provide a comparative summary, translating the observed graphical behaviour into formal control characteristics, vital for understanding the underlying mechanism of the ETPRL.

Table 4.6: Qualitative comparison of tracking performance during test 01

Feature	**ETPRL (Proposed)**	**DP.RL**	**Power RL**	**Constant RL**
Trajectory Tracking Accuracy	Highest (Minimal deviation)	Good (Moderate deviation)	Fair (Significant deviation)	Poor (Large deviation)
Attitude Tracking Accuracy	Highest (Minimal error)	Good (Moderate error)	Fair (Noticeable error)	Poor (Large error/oscillations)
Recovery Speed (Post-Fault)	Fastest	Fast	Moderate	Slowest
Stability Under Cumulative Faults	Excellent	Good	Moderate	Fair/Poor
Overall Tracking Robustness	Superior	High	Moderate	Low

Table 4.7: Qualitative comparison of control characteristics and chattering during test 01

Feature	**ETP.RL (Proposed)**	**DP.RL**	**Power RL**	**Constant RL**
Control Effort Smoothness (Torques)	High (Smooth profiles)	Good (Relatively smooth)	Moderate (Some peaks/oscillations)	Low (Sharp peaks/oscillations)
Control Signal Smoothness (Internal)	High (Minimal oscillations)	Good (Minor oscillations)	Moderate (Noticeable oscillations)	Low (Significant oscillations)
Chattering Level (Inferred)	Lowest	Low	Moderate	Highest
Convergence Speed (Sliding Variable)	Fastest	Fast	Moderate	Slowest
Control Demand (Post-Fault)	Efficient adaptation	Efficient adaptation	Higher demand/oscillations	Highest demand/oscillations

C. 3D Trajectory Tracking Performance (*Figure 4-13*):

Figure 4-13 illustrates the 3D position of the quadrotor alongside the desired helical trajectory (red line). The figure clearly delineates the three fault injection points at $t = 15s$, $t = 30s$, and $t = 45s$.

- *Transient Response to Faults*: Immediately following each fault injection, the Constant RL (magenta) and Power RL (green) exhibit the most pronounced and sustained transient deviations from the desired path (red). This indicates a slower recovery speed and a less robust ability to adapt to sudden changes in actuator effectiveness.

- *Performance Under Cumulative Stress*: The ETPRL controller consistently maintains the closest adherence to the desired trajectory. Even after the severe 55% LOE fault at $t = 45s$, the ETPRL trajectory remains highly stable and minimally deviated, a performance characteristic that strongly contrasts with the large, persistent spirals observed for the Constant RL.

- *Conclusion*: The visual evidence in *Figure 4-13* confirms the quantitative metrics (Tables 4.4 and 4.5 in the previous analysis): the ETPRL significantly minimizes tracking error and demonstrates superior resilience to cumulative, high-magnitude actuator faults, underscoring its enhanced fault-tolerant capability.

D. Evolution of Attitude Angles (*Figure 4-16*):

Figure 16 presents the evolution of the roll ('ϕ'), pitch ('θ'), and yaw ('ψ') Euler angles against their desired values (red lines). Attitude stability is critical for accurate trajectory tracking.

- *Stability and Oscillation*: In the fault-free phase, all controllers maintain the desired attitude. However, post-fault injection, the Constant RL (magenta) and Power RL (green) display significant, high-frequency oscillations and large sustained deviations, particularly in the roll and pitch axes. These oscillations are a clear indicator of poor damping and increased chattering due to slower convergence to the sliding manifold.

- *Fast Recovery and Chattering Suppression*: The ETPRL (blue) and DPRL (black) controllers maintain significantly better attitude tracking. The ETPRL, in particular, exhibits the smallest perturbations and the fastest return to the desired angle after each fault transient. The high-frequency noise is substantially attenuated, validating the core design objective of the Enhanced Triple Power Reaching Law to suppress chattering while accelerating convergence.

E. Control Effort and Chattering Analysis (Figures 4-14 & 4-15):

Figures 14 and 15 illustrate the controller's internal response (Control Signals U_ϕ, U_θ, U_ψ) and the external physical control effort (Applied Torques, Thrust, U_ϕ, U_θ, U_ψ).

- *Internal Control Signals (Figure 4-14)*: The internal signals for the Constant RL exhibit the largest, sharpest peaks and the most oscillatory behavior, especially after $t = 45s$. This aggressive, oscillatory signal profile is the primary source of the chattering phenomena. In contrast, the ETPRL and DPRL exhibit smooth, minimal fluctuations, suggesting rapid and effective convergence to the sliding surface, fulfilling the theoretical goal of chattering reduction.

- *Applied Torques (Figure 4-15)*: The applied torques reflect the total effort required from the actuators. Following the most severe fault at $t = 45s$, the magnitude of thrust and control torques increases for all controllers to compensate for the loss of effectiveness. Crucially, the profiles for the ETPRL and DPRL are remarkably smoother and more regulated. The Constant RL, however, displays sharp, high-magnitude spikes (e.g., in U_ϕ and U_ψ), which are indicative of high-frequency switching and significant chattering that can lead to actuator wear and potential control saturation.

- *Conclusion on Efficiency*: The ETPRL achieves the highest tracking accuracy (Figure 4-13) with the smoothest and most damped control signals (Figures 4-14 and 4-15). This juxtaposition demonstrates that the ETPRL not only enhances performance but does so with greater control efficiency and less aggressive actuation, which is a significant advantage for long-term UAV operation.

F. FDI Validation (Figure 4-17):

Figure 17, showing the "Faults estimated amplitude by the FDI unit," confirms the successful operation of the fault detection and identification (FDI) module, which is an integral part of the HFTC architecture. The Extended Kalman Filter (EKF)-based FDI accurately estimates the time of occurrence and the amplitude of the LOE faults for Motors 1, 2, and 3:

- *Motor 01*: 20% LOE initiated at $t \approx 15s$.
- *Motor 02*: 35% LOE initiated at $t \approx 30s$.
- *Motor 03*: 55% LOE initiated at $t \approx 45s$.

The precise and timely estimation of the fault information is validated, which is the foundational prerequisite for the subsequent control reconfiguration and parameter selection (based on the pre-computed PSO fault accommodation data).

G. Synthesis:

The evidence provided by the integrated quantitative and qualitative analysis of the figures and tables from Test 01 unequivocally supports the central claims of this thesis chapter. The ETPRL-based HFTC demonstrates superior robustness, the fastest post-fault recovery, and significantly attenuated chattering compared to conventional SMC reaching laws, confirming its effectiveness in maintaining stability and high-precision trajectory tracking for UAV quadrotors under severe, cumulative actuator faults.

4.5.2 The 2nd Test (Squared path)

To further evaluate the robustness and fault-tolerant capabilities of the proposed Hybrid Fault-Tolerant Control (HFTC) strategy under challenging manoeuvre conditions, Test 02 was conducted. This test involved commanding the quadrotor to follow a predefined square trajectory in 3D space over a total duration of 100 seconds, after an initial ascent to an altitude of 5 meters.

This scenario was specifically designed to challenge the controller's performance during sharp, instantaneous changes in direction at the corners of the square, simulating dynamic stress conditions. During this manoeuvre, the quadrotor was subjected to two successive Loss of Effectiveness (LOE) faults injected into different actuators :

- *First Fault*: A 20% 'LOE' fault was introduced in Motor 1 at $t = 30$ seconds.
- *Second Fault*: This was followed by a more severe 50% LOE fault in Motor 2 at $t = 70$ seconds.

The performance of the proposed hybrid fault-tolerant sliding mode controller based on the Enhanced Triple Power Reaching Law (ETPRL) was again rigorously compared against controllers using a double power reaching law, a standard power reaching law, and a basic constant reaching law. This comparative analysis under a different, geometrically challenging trajectory serves to generalize the findings from Test 01 and further highlight the ETPRL's advantages in achieving superior convergence speed, chattering attenuation, and post-fault recovery.

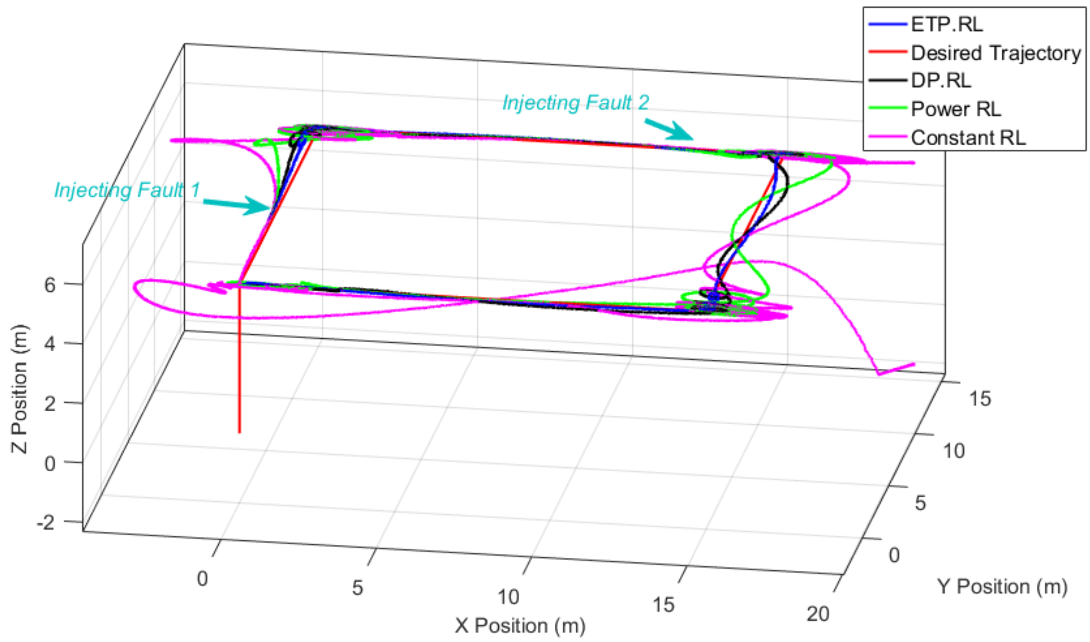

Figure 4.18: The followed square trajectory during test 02

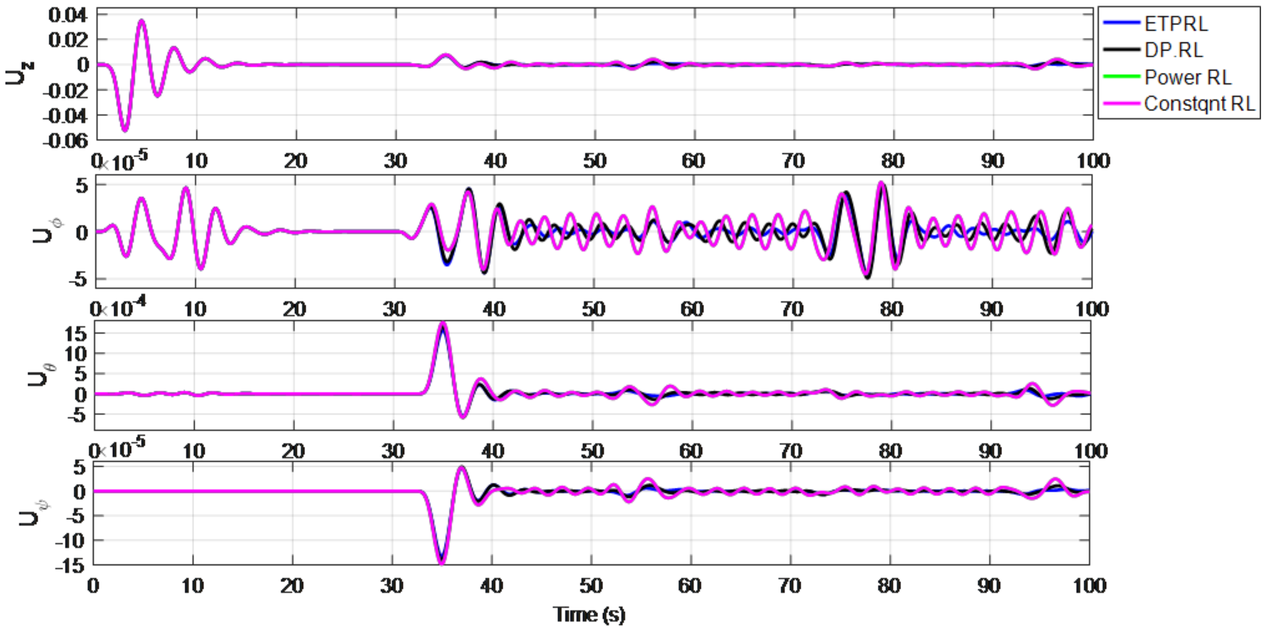

Figure 4.19: Variation of control signals during test 02

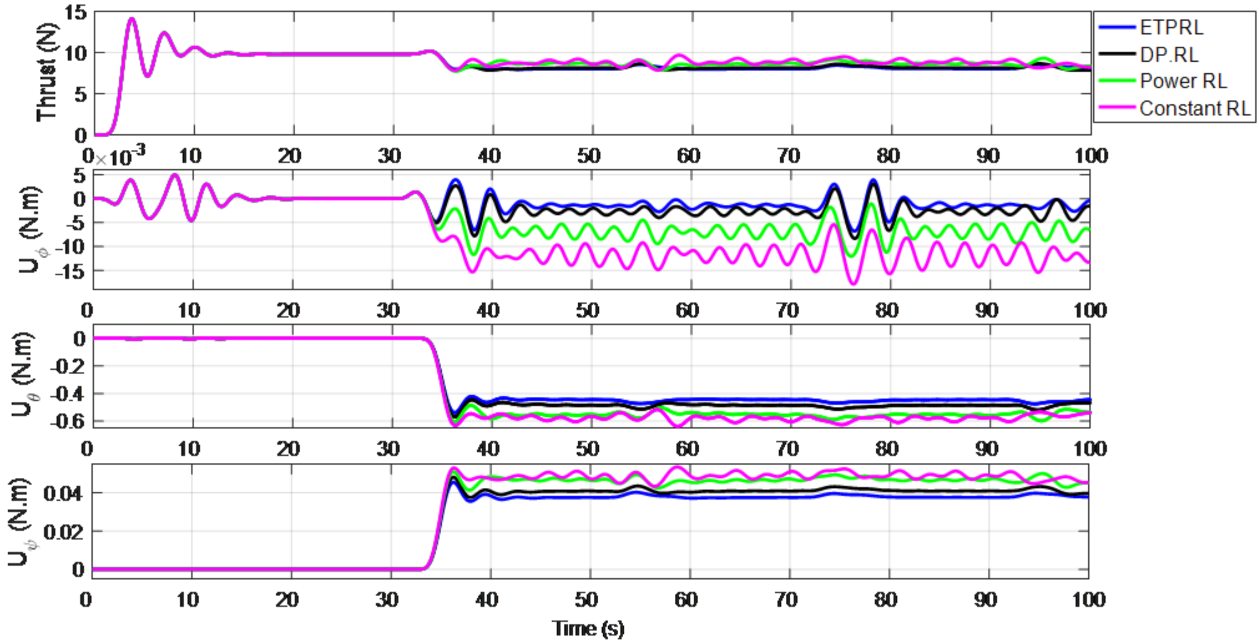

Figure 4.20: Applied torques during test 02

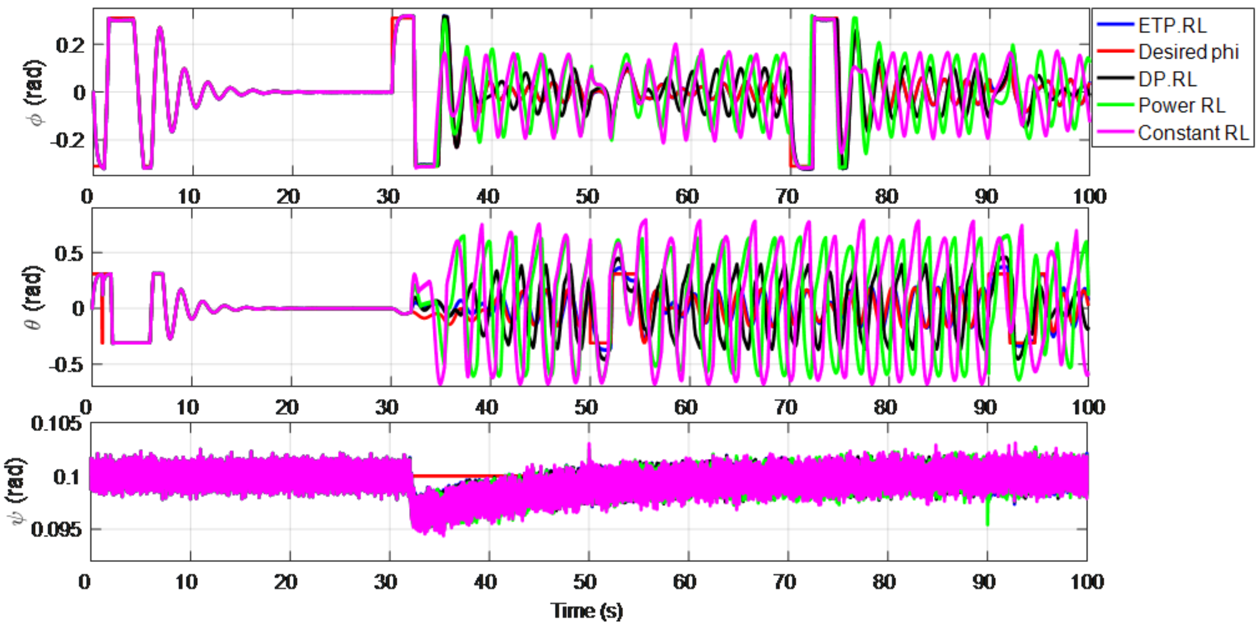

Figure 4.21: Evolution of roll, pitch yaw angles during test 02

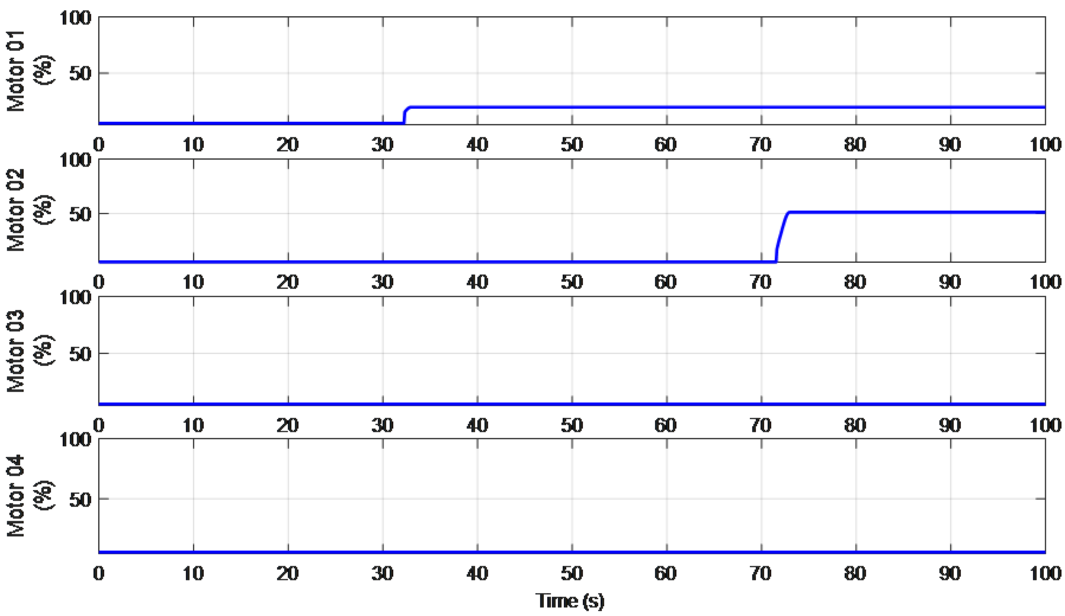

Figure 4.22: Faults estimated amplitude by the FDI unit

4.5.2.1 Performance Metrics and Discussion

The objective of Test *02*, involving a 3D square trajectory subject to two successive Loss of Effectiveness (LOE) faults (20% at $t = 30s$ and 50% at $t = 70s$), is to rigorously evaluate the capability of the Enhanced Triple Power Reaching Law (ETPRL) to handle sharp trajectory changes in the presence of faults. The results are compared against the DPRL, Power RL, and Constant RL benchmarks.

A. Quantitative Performance Assessment

Table 4.8 provides the estimated peak trajectory deviation for each controller following the two successive LOE fault injections, measured in meters (m).

Table 4.8: Estimated trajectory tracking performance metrics during test 02

Controller	Peak Deviation after Fault 01 (m)	Peak Deviation after Fault 02 (m)
ETPRL	0.2	0.4
DPRL	0.6	1.1
Power RL	1.8	2
Constant RL	4.9	4.9

The data presented in the table above demonstrates a clear hierarchy in fault tolerance. The ETPRL consistently achieves the *lowest Peak Deviation* after both the minor (20%) and severe (50%) faults. Following the severe second fault, the ETPRL deviation (0.4m) is notably better than the DPRL (1.1m) and significantly smaller than the Power RL (2.0m). The Constant RL exhibits a substantial and persistent deviation (4.9m), indicating a major struggle to compensate for the combined effect of the two faults.

Moreover, the minimal deviation of the ETPRL, particularly during the sharp corners of the square trajectory (as visually confirmed in *Figure 4.18*), highlights its superior ability to maintain a precise track under both actuator degradation and high dynamic manoeuvring demands, which proves its robustness under geometric stress.

B. Trajectory Tracking Performance (Figure 4.18)

Figure 4.18 illustrates the tracked 3D square trajectory. The fault injection points are indicated as "Injecting Fault 1" ($t = 30s$) and "Injecting Fault 2" ($t = 70s$).

- *Initial Tracking*: All controllers initially demonstrate satisfactory tracking. However, the introduction of the first fault (20% LOE) immediately stresses the system, and differences in robustness become apparent.

- *Post-Fault Adherence*: The Constant RL (magenta) displays the most significant tracking error and deviation, even before the second fault. Following the injection of the second, severe fault (50% LOE), the ETPRL (blue) shows the smallest transient deviation and maintains close adherence to the desired path (red) throughout the manoeuvre. The DPRL also shows reasonable fault tolerance, but with slightly larger deviations than the ETPRL, while the Power RL struggles more noticeably. This visually confirms the enhanced fault-tolerant capability provided by the ETPRL strategy, consistent with the minimal values in *Table 4.8*.

C. Evolution of Attitude Angles (Figure 4.21)

Figure 21 presents the evolution of the roll (ϕ), pitch (θ), and yaw (ψ) Euler angles.

- *Chattering in Fault-Free State*: Prior to the faults, all controllers track the reference angles, though the Constant RL exhibits higher frequency oscillations, indicative of an inherent chattering tendency.

- *Post-Fault Stability*: Following the first fault at $t = 30s$, the ETPRL demonstrates well-damped oscillations and rapid convergence back to the desired attitude references. The second fault at $t = 70s$ induces larger angular deviations for all controllers. The Constant RL shows pronounced oscillations, particularly in pitch, bordering on instability. In contrast, the proposed ETPRL effectively suppresses the disturbances caused by both faults, showcasing superior attitude stabilization and robustness with minimal oscillation. This smooth angular control is key to achieving the precise trajectory tracking seen in *Figure* 4.18.

D. Control Effort and Chattering Analysis (Figures 4.19 & 4.20):

Figures 19 and 20 illustrate the controller's internal response (Control Signals) and the external physical control effort (Applied Torques and Thrust), respectively.

- *Control Effort Compensation (Figure 4.19)*: Following successive actuator faults, the control scheme must adapt the applied thrust and torques. The Constant RL approach, while maintaining stability, requires significantly larger and more oscillatory torque transients, which indicates a higher control effort and greater potential for actuator wear. The proposed ETPRL-based controller effectively handles both minor and severe LOE faults, generating smooth, bounded, and rapidly converging torque signals that are comparable to, and in some aspects slightly better damped than, those produced by DPRL and Power RL.

- *Control Signal Response (Figure 4.20)*: The relative smoothness of the ETPRL control signals, particularly in U_ϕ and U_θ, compared to the more oscillating benchmarks, suggests rapid and effective convergence to the sliding manifold with inherent chattering reduction. This highlights the ETPRL's ability to ensure robust fault tolerance and stable trajectory tracking without demanding excessive or aggressive control actions, which is advantageous for actuator longevity and overall flight quality.

E. FDI Validation (Figure 4.22):

Figure 22, showing the "Faults estimated amplitude by the FDI unit," confirms the successful operation of the Fault Detection and Isolation (FDI) module. The plot validates that the EKF-based FDI unit successfully detects and accurately estimates the magnitude (20% and 50%) and timing ($t = 30s$ and $t = 70s$) of the injected LOE faults on Motors 1 and 2, respectively. This accurate online fault information is the essential input for the active fault-tolerant control component, enabling the system to adapt effectively via the pre-optimized compensation mechanism.

F. Synthesis:

Test *02* confirms the findings of Test *01* and generalizes the superior performance of the ETPRL-based HFTC to a more challenging manoeuvre (square trajectory). The ETPRL provides the best balance between tracking precision, chattering attenuation, and control efficiency.

The minimal tracking deviation (*Table 4.8*), smooth control signals, and stable attitude angles confirm the effectiveness of the proposed strategy in mitigating severe actuator faults while maintaining high stability and manoeuvrability.

4.6 Conclusion

This chapter served as the capstone of the thesis, successfully applying the theoretical and methodological frameworks developed in the preceding chapters to the practical problem of Fault-Tolerant Control for the highly nonlinear UAV Quadrotor system.

The chapter began by establishing the Nonlinear Dynamic Model of the quadrotor, which formed the White-Box foundation for the control design. This was followed by the definition of critical Actuator Fault Scenarios, with the primary focus being the Loss of Effectiveness (LOE) fault.

The core contribution was realized in the Fault-Tolerant Control System Design and Implementation (Section 04), which presented the novel Hybrid Fault-Tolerant Sliding Mode Control (HFTC) architecture. This architecture successfully merged two key advanced features:

1. *Online Diagnosis*: A real-time Extended Kalman Filter (EKF) was utilized for swift Fault Detection and Identification (FDI).

2. *Hybrid Accommodation*: A unique strategy was implemented wherein optimal SMC parameters were pre-computed offline using Particle Swarm Optimization (PSO) and rapidly deployed via the Reconfiguration Mechanism based on the EKF's fault estimate. The controller itself was further enhanced by proposing the Enhanced Triple Power Reaching Law (ETPRL), specifically formulated to accelerate convergence while mitigating the inherent chattering problem of SMC.

The efficacy of the proposed HFTC-ETPRL strategy was rigorously validated in Simulation and Analysis (Section 05) through two distinct and challenging manoeuvres:

- *Test 01 (Helical Path)*: This test validated the controller's robustness against cumulative and increasing severity LOE faults in three different motors. The quantitative results demonstrated that the ETPRL achieved the lowest Peak Deviation (e.g., nearly half that of the DPRL after the severe 55% fault) and the lowest IAE for tracking error and control effort, confirming superior tracking precision, stability, and control efficiency under severe, compounded stress.

- *Test 02 (Square Path)*: This test confirmed the controller's effectiveness during high-dynamic, non-smooth manoeuvres (sharp corners) under two successive LOE faults. The ETPRL again demonstrated its superiority by maintaining the lowest Peak Deviation (e.g., 0.4m after the 50% fault) and producing the smoothest control effort signals, thereby validating its enhanced ability to suppress chattering and ensure stable attitude control in challenging conditions.

In conclusion, the results of the case study unequivocally demonstrated the superior robustness and adaptive performance of the proposed Hybrid Fault-Tolerant Sliding Mode Controller based on the Enhanced Triple Power Reaching Law (ETPRL). The strategy proved highly effective in maintaining the stability and high-precision trajectory tracking of the quadrotor in the face of sudden, severe actuator failures. This successful application validates the overarching thesis methodology, confirming that the strategic combination of advanced analytical control design (SMC-ETPRL), model-based diagnosis (EKF), and AI-based optimization (PSO) provides a powerful, robust, and deployable solution for fault management in complex, nonlinear systems.

General Conclusion

This thesis has successfully addressed the critical challenge of ensuring the safety and reliability of nonlinear systems through the development and validation of a novel Hybrid Fault-Tolerant Control (HFTC) framework. The primary contribution of this research is the effective resolution of the fundamental scientific trade-off between the verifiable robustness of conventional control and the adaptive performance of Artificial Intelligence. By integrating a rigorous Sliding Mode Control (SMC) architecture with a metaheuristic-optimized offline parameter bank, this work has established a control strategy that is both mathematically sound and highly adaptable to severe system failures.

The empirical results obtained from the high-fidelity UAV quadrotor case study provide unequivocal evidence of the system's superiority. The proposed Enhanced Triple Power Reaching Law (ETPRL) was quantitatively proven to outperform standard reaching laws, achieving a reduction in peak trajectory deviation of up to 50 % under cumulative actuator faults (55% Loss of Effectiveness). Furthermore, the analysis demonstrated that this performance was achieved with significantly reduced control effort and attenuated chattering, thereby directly contributing to the longevity of electromechanical actuators and the energy efficiency of the system. The successful synchronization of the offline Particle Swarm Optimization (PSO) with online Extended Kalman Filter (EKF) diagnosis validates the hybrid architecture as a practical, high-performance solution for safety-critical applications.

To arrive at this innovative solution, the research followed a systematic theoretical exploration. *Chapters 01* and *02* critically evaluated the state-of-the-art in diagnosis and fault-tolerant control, identifying the limitations of purely model-based methods (brittleness) and purely AI-based methods (lack of interpretability) that necessitated a hybrid approach. *Chapter 03* established the modeling foundations, highlighting the necessity of balancing physical insight with computational tractability. This theoretical groundwork culminated in the design and implementation detailed in *Chapter 04*, where the theoretical hypothesis was transformed into a validated engineering solution.

In synthesis, this dissertation makes the following primary contributions to the field of control engineering:

1. **A comprehensive and critical synthesis of the state-of-the-art**: It has provided a structured and comparative review of conventional and AI-based methods for the diagnosis, control, and modeling of nonlinear systems, clearly articulating the research gaps

and motivating the direction of the proposed work.

2. **The design and validation of a novel control law**: It has introduced the Enhanced Triple Power Reaching Law (ETPRL) for Sliding Mode Control, demonstrating through rigorous simulation that this law provides faster system convergence and superior chattering attenuation compared to conventional reaching laws.

3. **The successful implementation of a high-performance HFTC system**: It has successfully integrated the novel ETPRL within a complete hybrid fault-tolerant control architecture, proving its practical efficacy in a challenging, safety-critical application. The results confirm that this architecture offers a robust and high-performance solution for maintaining stability and mission objectives in the presence of severe system faults.

Perspectives for Future Work

While this thesis has successfully achieved its objectives, it also opens several promising avenues for future research.

- **Experimental Validation**: The immediate and most crucial next step is to transition the proposed HFTC-SMC algorithm from the simulation environment to a real-world hardware platform. Experimental validation on a physical quadrotor testbed would provide the ultimate proof of the controller's effectiveness, accounting for real-world complexities such as communication delays, sensor noise, and aerodynamic uncertainties.

- **Integration of Advanced AI for Diagnosis**: The current FDI module is based on an EKF. A compelling future direction would be to replace or augment this module with an AI-based diagnostic system, such as a deep neural network, which could potentially offer faster and more robust fault detection and isolation, especially for more subtle or complex fault types.

- **Extension to Sensor and System-Wide Faults**: The case study focused on actuator faults. Future work could extend the HFTC framework to handle a broader class of anomalies, including sensor faults (bias, drift) and system-level failures, to create a more holistic and resilient control architecture.

- **Formal Verification and Safety Guarantees**: While the proposed ETPRL demonstrated superior performance, future research could focus on the formal mathematical verification of its stability and robustness properties, potentially using Lyapunov-based methods or Sum-of-Squares optimization, to provide certifiable safety guarantees for its deployment in mission-critical applications.

In conclusion, this research has advanced the state-of-the-art in fault-tolerant control by proposing and validating a novel and highly effective solution. The findings herein not only contribute to the body of academic knowledge but also offer a practical and powerful tool for engineers and researchers working to enhance the safety and autonomy of the next generation of complex nonlinear systems.

Bibliography

[1] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Chapman and Hall/CRC, 2024.

[2] Wenyin Gong, Zuowen Liao, Xianyan Mi, Ling Wang, and Yuanyuan Guo. Nonlinear equations solving with intelligent optimization algorithms: A survey. *Complex System Modeling and Simulation*, 1(1):15–32, 2021.

[3] Zongyu Zuo, Cunjia Liu, Qing-Long Han, and Jiawei Song. Unmanned aerial vehicles: Control methods and future challenges. *IEEE/CAA Journal of Automatica Sinica*, 9(4):601–614, 2022.

[4] Xiaoliang Chen, Che-Yu Liu, Roberto Proietti, Zhaohui Li, and SJ Ben Yoo. Automating optical network fault management with machine learning. *IEEE Communications Magazine*, 60(12):88–94, 2022.

[5] MohammadSaleh Hedayati, Ailin Barzegar, and Afshin Rahimi. Fault diagnosis and prognosis of satellites and unmanned aerial vehicles: A review. *Applied Sciences*, 14(20):9487, 2024.

[6] Andrzej Milecki and Patryk Nowak. Review of fault-tolerant control systems used in robotic manipulators. *Applied Sciences*, 13(4):2675, 2023.

[7] Steven X Ding. *Advanced methods for fault diagnosis and fault-tolerant control*, volume 184. Springer, 2021.

[8] Amara Katabarwa, Katerina Gratsea, Athena Caesura, and Peter D Johnson. Early fault-tolerant quantum computing. *PRX quantum*, 5(2):020101, 2024.

[9] Achu Govind KR. Enhancing fault tolerance and robustness in chemical process control using learning-based pid design. *Chemical Product and Process Modeling*, (0), 2026.

[10] Janki Kaushal. *Evaluation and Comparison of Linear Quadratic Control Techniques*. PhD thesis, Master's thesis, Delft University of Technology, 2024.

[11] Hao Cai, Limin Wang, Yun Liu, Jingxian Yu, and Tao Zou. H-infinity fault-tolerant control for fully actuated systems with its application. In *2025 CAA Symposium on Fault*

Detection, Supervision, and Safety for Technical Processes (SAFEPROCESS), pages 1–5. IEEE, 2025.

[12] Samir Zeghlache, Hilal Rahali, Ali Djerioui, Loutfi Benyettou, and Mohamed Fouad Benkhoris. Robust adaptive backstepping neural networks fault tolerant control for mobile manipulator uav with multiple uncertainties. *Mathematics and computers in simulation*, 218:556–585, 2024.

[13] Yiming Xu, Xiaohua Ge, Ruohan Guo, and Weixiang Shen. Recent advances in model-based fault diagnosis for lithium-ion batteries: A comprehensive review. *Renewable and Sustainable Energy Reviews*, 207:114922, 2025.

[14] Cristian López, Ángel Naranjo, Siliang Lu, and Keegan J Moore. Hidden markov model based stochastic resonance and its application to bearing fault diagnosis. *Journal of Sound and Vibration*, 528:116890, 2022.

[15] Hamed Badihi, Youmin Zhang, Bin Jiang, Pragasen Pillay, and Subhash Rakheja. A comprehensive review on signal-based and model-based condition monitoring of wind turbines: Fault diagnosis and lifetime prognosis. *Proceedings of the IEEE*, 110(6):754–806, 2022.

[16] Yazhong Zhou, Yigang He, Zhikai Xing, Lei Wang, Kaixuan Shao, Leixiao Lei, and Zihao Li. Vibration signal-based fusion residual attention model for power transformer fault diagnosis. *IEEE Sensors Journal*, 24(10):17231–17242, 2024.

[17] Saba Shafi and Anil V Parwani. Artificial intelligence in diagnostic pathology. *Diagnostic pathology*, 18(1):109, 2023.

[18] Midhun T Augustine. A survey on universal approximation theorems. *arXiv preprint arXiv:2407.12895*, 2024.

[19] Dennis Rochau, Robin Chan, and Hanno Gottschalk. New advances in universal approximation with neural networks of minimal width. *arXiv preprint arXiv:2411.08735*, 2024.

[20] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

[21] Remus C Avram. Fault diagnosis and fault-tolerant control of quadrotor uavs. 2016.

[22] Ziquan Yu, Youmin Zhang, Bin Jiang, and Chun-Yi Su. *Fault-Tolerant Cooperative Control of Unmanned Aerial Vehicles*. Springer, 2024.

[23] Ahmed Khattab. *Fault tolerant control of multi-rotor unmanned aerial vehicles using sliding mode based schemes.* University of Exeter (United Kingdom), 2020.

[24] Christoph Elias Aoun. Fault tolerant deep reinforcement learning for aerospace applications. 2021.

[25] Menglin He. *Commande tolérante aux fautes des systèmes non linéaires basée sur l'approche multi-modèles.* PhD thesis, Toulouse 3, 2021.

[26] Philip J Weiser. Toward a next generation regulatory strategy. *Loy. U. Chi. LJ*, 35:41, 2003.

[27] Tobias Overgaard, Maria-Ona Bertran, Bo Friis Nielsen, Pierantonio Facco, and Massimiliano Barolo. Beyond black box models: A systematic approach to measure physics-based vs. data-driven contributions in hybrid process models. *Data-Driven Contributions in Hybrid Process Models.*

[28] Abdelghani Djeddi, Abdelaziz Aouiche, Chaima Aouiche, and Yazeed Alkhrijah. State estimation based state augmentation and fractional order proportional integral unknown input observers. *Mathematics*, 13(11):1786, 2025.

[29] Mohanad Nawfal Mustafa. Classification of maintenance techniques and diagnosing failures methods. In *Journal of Physics: Conference Series*, volume 2060, page 012014. IOP Publishing, 2021.

[30] Heiko Webert, Tamara Döß, Lukas Kaupp, and Stephan Simons. Fault handling in industry 4.0: definition, process and applications. *Sensors*, 22(6):2205, 2022.

[31] Zhiwei Gao and Xiaoxu Liu. An overview on fault diagnosis, prognosis and resilient control for wind turbine systems. *Processes*, 9(2):300, 2021.

[32] Y. Bouali, B. Mohammed, and K. Ahmed. "review of renewable energy sources". *Random journal*, 1(2):01–02, 2023.

[33] Wenhao Yan, Jing Wang, Shan Lu, Meng Zhou, and Xin Peng. A review of real-time fault diagnosis methods for industrial smart manufacturing. *Processes*, 11(2):369, 2023.

[34] Roberto M Souza, Erick GS Nascimento, Ubatan A Miranda, Wenisten JD Silva, and Herman A Lepikson. Deep learning for diagnosis and classification of faults in industrial rotating machinery. *Computers & Industrial Engineering*, 153:107060, 2021.

[35] Yuanfang Chi, Yanjie Dong, Z Jane Wang, F Richard Yu, and Victor CM Leung. Knowledge-based fault diagnosis in industrial internet of things: a survey. *IEEE Internet of Things Journal*, 9(15):12886–12900, 2022.

[36] Chengze Liu, Andrzej Cichon, Grzegorz Królczyk, and Zhixiong Li. Technology development and commercial applications of industrial fault diagnosis system: a review. *The International Journal of Advanced Manufacturing Technology*, 118(11):3497–3529, 2022.

[37] Russul H Hadi, Haider N Hady, Ahmed M Hasan, Ammar Al-Jodah, and Amjad J Humaidi. Improved fault classification for predictive maintenance in industrial iot based on automl: A case study of ball-bearing faults. *Processes*, 11(5):1507, 2023.

[38] Merim Dzaferagic, Nicola Marchetti, and Irene Macaluso. Fault detection and classification in industrial iot in case of missing sensor data. *IEEE Internet of Things Journal*, 9(11):8892–8900, 2021.

[39] Abla Bouguerne, Aziz Boukadoum, and Mohamed Salah Djebbar. Artificial neural networks for automatic classification of induction machine faults. *AIJR Abstracts*, pages 59–61, 2024.

[40] Wenhao Yan, Jing Wang, Shan Lu, Meng Zhou, and Xin Peng. A review of real-time fault diagnosis methods for industrial smart manufacturing. *Processes*, 11(2):369, 2023.

[41] José Cação, José Santos, Mário Antunes, and António Completo. Machine learning-driven fault identification and classification: a two-step approach for industrial applications. *Procedia Computer Science*, 253:1073–1082, 2025.

[42] Jianbo Yu and Yue Zhang. Challenges and opportunities of deep learning-based process fault detection and diagnosis: a review. *Neural Computing and Applications*, 35(1):211–252, 2023.

[43] Ying Zuo, Chunyan Lai, and K. Lakshmi Varaha Iyer. A review of sliding mode observer based sensorless control methods for pmsm drive. *IEEE Transactions on Power Electronics*, 38(9):11352–11367, 2023.

[44] Yaofei Han, Shaofeng Chen, Chao Gong, Xing Zhao, Fanggang Zhang, and Yunwei Li. Accurate sm disturbance observer-based demagnetization fault diagnosis with parameter mismatch impacts eliminated for ipm motors. *IEEE Transactions on Power Electronics*, 38(5):5706–5710, 2023.

[45] Abdelghani Djeddi, Ahmad Taher Azar, Abdelhamid Djari, Djalel Dib, Samah Alshathri, and Walid El-Shafai. Robust fault diagnosis of fractional order takagi–sugeno systems with uncertainties in premise variables. *Scientific Reports*, 15(1):32163, 2025.

[46] Jin-Xi Zhang, Qing-Guo Wang, and Wei Ding. Global output-feedback prescribed performance control of nonlinear systems with unknown virtual control coefficients. *IEEE Transactions on Automatic Control*, 67(12):6904–6911, 2022.

[47] Z. Q. Zhu, Dawei Liang, and Kan Liu. Online parameter estimation for permanent magnet synchronous machines: An overview. *IEEE Access*, 9:59059–59084, 2021.

[48] Luis Enciso, Matti Noack, Johann Reger, and Gustavo Pérez-Zuñiga. Modulating function based fault diagnosis using the parity space method. *IFAC-PapersOnLine*, 54(7):268–273, 2021. 19th IFAC Symposium on System Identification SYSID 2021.

[49] Jialu Liu, Jinkui Feng, and Dapeng Zhang. Application of parity relation in elevator fault diagnosis. In *2023 35th Chinese Control and Decision Conference (CCDC)*, pages 688–692, 2023.

[50] Siwen Hao, Maiying Zhong, Ningfan Zhong, Qin Wang, and Yanzheng Zhu. Parity space-based sensor fault detection for traction rectifier in high-speed train. In *2024 China Automation Congress (CAC)*, pages 3511–3516, 2024.

[51] MohammadSaleh Hedayati, Ailin Barzegar, and Afshin Rahimi. Fault diagnosis and prognosis of satellites and unmanned aerial vehicles: A review. *Applied Sciences*, 14(20), 2024.

[52] A new extended state observer for uncertain nonlinear systems. *Automatica*, 131:109772, 2021.

[53] Guoyuan Qi, Xia Li, and Zengqiang Chen. Problems of extended state observer and proposal of compensation function observer for unknown model and application in uav. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(5):2899–2910, 2022.

[54] Shoresh Shokoohi and Jamal Moshtagh. Beyond signal processing: A model-based luenberger observer approach for accurate bearing fault diagnosis. *AUT Journal of Electrical Engineering*, 57(1):163–184, 2025.

[55] Laatra Yousfi, Sakina Aoun, and Moussa Sedraoui. Speed sensorless vector control of doubly fed induction machine using fuzzy logic control equipped with luenberger observer. *International Journal of Dynamics and Control*, 10(6):1876–1888, 2022.

[56] Yousef Al-Mutayeb and Moayed Almobaied. Luenberger observer-based speed sensor fault detection of bldc motors. In *2021 International Conference on Electric Power Engineering – Palestine (ICEPE- P)*, pages 1–7, 2021.

[57] Antoine Hugo. *Design of Interval Observers for Fault Diagnosis and Robust Control with Application to Quadcopter UAVs.* Theses, Normandie Université, November 2024.

[58] Songchun Zou, Wanzhong Zhao, Weihe Liang, Chunyan Wang, and Feng Chen. Fault diagnosis and fault-tolerant compensation strategy for wheel angle sensor of steer-by-wire vehicle via extended kalman filter. *IEEE Sensors Journal*, 22(2):1756–1766, 2022.

[59] Aziz Boukadoum, Abla Bouguerne, Tahar Bahi, Mohamed Salah Djebbar, and Abderrahmane Khechekhouche. Four-leg inverter based on three-dimensional space vector modulation method for desired output frequency. *Tehnički vjesnik*, 31(3):688–695, 2024.

[60] Xiaojia Han, Yiren Hu, Anhuan Xie, Xufei Yan, Xiaobo Wang, Chao Pei, and Dan Zhang. Quadratic-kalman-filter-based sensor fault detection approach for unmanned aerial vehicles. *IEEE Sensors Journal*, 22(19):18669–18683, 2022.

[61] Oscar Javier Montañez, Marco Javier Suarez, and Eduardo Avendano Fernandez. Application of data sensor fusion using extended kalman filter algorithm for identification and tracking of moving targets from lidar–radar data. *Remote Sensing*, 15(13), 2023.

[62] Vicente Borja-Jaimes, Antonio Coronel-Escamilla, Ricardo Fabricio Escobar-Jiménez, Manuel Adam-Medina, Gerardo Vicente Guerrero-Ramírez, Eduardo Mael Sánchez-Coronado, and Jarniel García-Morales. Fractional-order sliding mode observer for actuator fault estimation in a quadrotor uav. *Mathematics*, 12(8), 2024.

[63] Mohamed Salah Djebbar, Aziz Boukadoum, and Abla Bouguerne. Performances of a wind power system based on the doubly fed induction generator controlled by a multi-level inverter. *International Journal of Power Electronics and Drive Systems*, 14(1):100–110, 2023.

[64] Romeo Falcón, Héctor Ríos, and Alejandro Dzul. A robust fault diagnosis for quadrotors: A sliding-mode observer approach. *IEEE/ASME Transactions on Mechatronics*, 27(6):4487–4496, 2022.

[65] Tingting Yin, Zhou Gu, and Xiangpeng Xie. Observer-based event-triggered sliding mode control for secure formation tracking of multi-uav systems. *IEEE Transactions on Network Science and Engineering*, 10(2):887–898, 2023.

[66] Jie Jiang, Peter M. Atkinson, Chunsheng Chen, Qiang Cao, Yongchao Tian, Yan Zhu, Xiaojun Liu, and Weixing Cao. Combining uav and sentinel-2 satellite multi-spectral images to diagnose crop growth and n status in winter wheat at the county scale. *Field Crops Research*, 294:108860, 2023.

[67] Hamed Badihi, Youmin Zhang, Bin Jiang, Pragasen Pillay, and Subhash Rakheja. A comprehensive review on signal-based and model-based condition monitoring of wind turbines: Fault diagnosis and lifetime prognosis. *Proceedings of the IEEE*, 110(6):754–806, 2022.

[68] Sihem Ghoudelbourk, Djalel Dib, Amar Omeiri, and Ahmad Taher Azar. Mppt control in wind energy conversion systems and the application of fractional control (piα) in pitch wind turbine. *International Journal of Modelling, Identification and Control*, 26(2):140–151, 2016.

[69] Bayu Adhi Tama, Malinda Vania, Seungchul Lee, and Sunghoon Lim. Recent advances in the application of deep learning for fault diagnosis of rotating machinery using vibration signals. *Artificial Intelligence Review*, 56(5):4667–4709, 2023.

[70] Michael Greenacre, Patrick JF Groenen, Trevor Hastie, Alfonso Iodice d'Enza, Angelos Markos, and Elena Tuzhilina. Principal component analysis. *Nature Reviews Methods Primers*, 2(1):100, 2022.

[71] Jian Hu, Fan Su, Xia Ren, Lei Cao, Yumei Zhou, Yuhan Fu, Grace Tatenda, Mingfei Jiang, Huan Wu, and Yufeng Wen. Prediction of carotid plaque by blood biochemical indices and related factors based on fisher discriminant analysis. *BMC Cardiovascular Disorders*, 22(1):371, 2022.

[72] Mohamad Hazwan Mohd Ghazali and Wan Rahiman. A novel fault detection approach in uav with adaptation of fuzzy logic and sensor fusion. *IEEE/ASME Transactions on Mechatronics*, 30(1):381–391, 2025.

[73] Nizar Benayad, Abdelaziz Aouiche, and Abdelghani Djeddi. Fuzzy logic speed controller for dual stator induction motor based on indirect vector control. In *IAM*, pages 126–135, 2023.

[74] Tao Lei, Yanbo Wang, Xianqiu Jin, Zhihao Min, Xingyu Zhang, and Xiaobin Zhang. An optimal fuzzy logic-based energy management strategy for a fuel cell/battery hybrid power unmanned aerial vehicle. *Aerospace*, 9(2), 2022.

[75] Abdelaziz Aouiche, Asma Djellid, and Farid Bouttout. Fuzzy neuroconformal analysis of multilayer elliptical cylindrical and asymmetrical coplanar striplines. *AEU-International Journal of Electronics and Communications*, 69(9):1151–1166, 2015.

[76] Luttfi A. Al-Haddad and Alaa Abdulhady Jaber. An intelligent fault diagnosis approach for multirotor uavs based on deep neural network of multi-resolution transform features. *Drones*, 7(2), 2023.

[77] Sihem Ghoudelbourk, Ahmad Taher Azar, and Djalel Dib. Three-level (npc) shunt active power filter based on fuzzy logic and fractional-order pi controller. *International Journal of Automation and Control*, 15(2):149–169, 2021.

[78] Yumeng Ma, Faizal Mustapha, Mohamad Ridzwan Ishak, Sharafiz Abdul Rahim, and Mazli Mustapha. Structural fault diagnosis of uav based on convolutional neural network and data processing technology. *Nondestructive Testing and Evaluation*, 39(2):426–445, 2024.

[79] Iqbal Misbah, Carman KM Lee, and Kin Lok Keung. Fault diagnosis in rotating machines based on transfer learning: Literature review. *Knowledge-Based Systems*, 283:111158, 2024.

[80] Andrzej Milecki and Patryk Nowak. Review of fault-tolerant control systems used in robotic manipulators. *Applied Sciences*, 13(4), 2023.

[81] Fang Nan, Sihao Sun, Philipp Foehn, and Davide Scaramuzza. Nonlinear mpc for quadrotor fault-tolerant control. *IEEE Robotics and Automation Letters*, 7(2):5047–5054, 2022.

[82] Fuqiang Liu, Zuxing Ma, Bingxian Mu, Chaoqun Duan, Rui Chen, Yi Qin, Huayan Pu, and Jun Luo. Review on fault-tolerant control of unmanned underwater vehicles. *Ocean Engineering*, 285:115471, 2023.

[83] Xiaoyu Guo, Chenliang Wang, and Lu Liu. Adaptive fault-tolerant control for a class of nonlinear multi-agent systems with multiple unknown time-varying control directions. *Automatica*, 167:111802, 2024.

[84] Salman Ijaz, Hamdoon Ijaz, Mirza Tariq Hamayun, and Umair Javaid. Active and passive fault tolerant control allocation strategy for nonlinear systems. *Journal of Vibration and Control*, 29(15-16):3492–3515, 2023.

[85] Manuel A. Estrada, Leonid Fridman, and Jaime A. Moreno. Passive fault-tolerant control via sliding-mode-based lyapunov redesign. *IEEE Transactions on Automatic Control*, 69(10):6777–6788, 2024.

[86] Yashar Shabbouei Hagh, Reza Mohammadi Asl, Afef Fekih, Huapeng Wu, and Heikki Handroos. Active fault-tolerant control design for actuator fault mitigation in robotic manipulators. *IEEE Access*, 9:47912–47929, 2021.

[87] Robab Ebrahimi Bavili, Ardashir Mohammadzadeh, Jafar Tavoosi, Saleh Mobayen, Wudhichai Assawinchaichote, Jihad H. Asad, and Amir H. Mosavi. A new active fault tolerant control system: Predictive online fault estimation. *IEEE Access*, 9:118461–118471, 2021.

[88] Saddam Hocine Derrouaoui, Yasser Bouzid, and Mohamed Guiatni. Nonlinear robust control of a new reconfigurable unmanned aerial vehicle. *Robotics*, 10(2), 2021.

[89] Abdelaziz Aouiche. Physique des semi-conducteurs. 2022.

[90] Abdelaziz Hamza, Abdallah H Mohamed, and Ayman El-Badawy. Robust h-infinity control for a quadrotor uav. In *AIAA SCITECH 2022 Forum*, page 2033, 2022.

[91] Nanmu Hui, Yunqian Guo, Xiaowei Han, and Baoju Wu. Robust h-infinity dual cascade mpc-based attitude control study of a quadcopter uav. In *Actuators*, volume 13, 2024.

[92] Toufik Amieur, Djamel Taibi, Sami Kahla, Mohcene Bechouat, and Moussa Sedraoui. Tilt-fractional order proportional integral derivative control for dc motor using particle swarm optimization. , (2):14–19, 2023.

[93] Yodsadej Kanokmedhakul, Sujin Bureerat, Natee Panagant, Thana Radpukdee, Nantiwat Pholdee, and Ali Riza Yildiz. Metaheuristic-assisted complex h-infinity flight control tuning for the hawkeye unmanned aerial vehicle: A comparative study. *Expert Systems with Applications*, 248:123428, 2024.

[94] Dehua Zhang, Lei Meng, Yao Hao, and Ruixue Xia. Adaptive sliding mode fault-tolerant control of uav systems based on radial basis function neural networks. *Scientific Reports*, 15(1):27504, 2025.

[95] Moshu Qian, Zhen Zhang, Zhong Zheng, and Cuimei Bo. Sliding mode control-based distributed fault tolerant tracking control for multiple unmanned aerial vehicles with input constraints and actuator faults. *International Journal of Robust and Nonlinear Control*, 33(15):9150–9173, 2023.

[96] Nizar Benayad, Abdelaziz Aouiche, and Abdelghani Djeddi. Fuzzy adaptive sliding mode control for robust fault tolerant operation of a uav quadrotor. *Journal of Engineering Science & Technology Review*, 18(5), 2025.

[97] Abla Bouguerne, Aziz Boukadoum, Tahar Bahi, Laatra Yousfi, and Nadhir Mesbahi. Performance analysis of a matrix converter under unbalanced load. In *International Conference on Electrical Engineering and Control Applications*, pages 118–130. Springer, 2024.

[98] Spasena Dakova, Julia L. Wagner, Andreas Gienger, Cristina Tarín, Michael Böhm, and Oliver Sawodny. Reconfiguration strategy for fault-tolerant control of high-rise adaptive structures. *IEEE Robotics and Automation Letters*, 6(4):6813–6819, 2021.

[99] Yuanyuan Tu, Dayi Wang, Steven X. Ding, Fangzhou Fu, and Wenbo Li. A reconfiguration-based fault-tolerant control method for nonlinear uncertain systems. *IEEE Transactions on Automatic Control*, 67(11):6060–6067, 2022.

[100] Rouzbeh Moradi and Sajjad Zalaghi. Managing reconfiguration time in optimal spacecraft active fault-tolerant attitude stabilization. *Sādhanā*, 50(2):70, 2025.

[101] Jun Zhang, Yi Zuo, and Shaocheng Tong. Observer-based fuzzy adaptive formation ftc for nonlinear mass against communication link and sensor faults. *International Journal of Fuzzy Systems*, pages 1–15, 2024.

[102] Zhengchao Xie, Wei You, Pak Kin Wong, Wenfeng Li, Xinbo Ma, and Jing Zhao. Robust fuzzy fault tolerant control for nonlinear active suspension systems via adaptive

hybrid triggered scheme. *International Journal of Adaptive Control and Signal Processing*, 37(7):1608–1627, 2023.

[103] Arslan Ahmed Amin, Muhammad Irfan, Turki Alsuwian, Saifur Rahman, Ansa Mubarak, and Saba Waseem. Applications of fault-tolerant control system in the design of wind turbine generation systems: a comprehensive review and future prospects. *Results in Engineering*, page 106205, 2025.

[104] Muhammad Hamza Shahbaz and Arslan Ahmed Amin. Design of hybrid fault-tolerant control system for air-fuel ratio control of internal combustion engines using artificial neural network and sliding mode control against sensor faults. *Advances in Mechanical Engineering*, 15(3):16878132231160729, 2023.

[105] Turki Alsuwian, Muhammad Tayyeb, Arslan Ahmed Amin, Muhammad Bilal Qadir, Saleh Almasabi, and Mohammed Jalalah. Design of a hybrid fault-tolerant control system for air–fuel ratio control of internal combustion engines using genetic algorithm and higher-order sliding mode control. *Energies*, 15(15), 2022.

[106] Yves Sohège, Marcos Quiñones-Grueiro, and Gregory Provan. A novel hybrid approach for fault-tolerant control of uavs based on robust reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10719–10725, 2021.

[107] Gulay Unal. Integrated design of fault-tolerant control for flight control systems using observer and fuzzy logic. *Aircraft Engineering and Aerospace Technology*, 93(4):723–732, 2021.

[108] Toufik AMIEUR, Djamel TAIBI, Mohcene BECHOUAT, Sami KAHLA, and Moussa SEDRAOUI. Fuzzy sliding mode with adaptive gain control for nonlinear mimo systems. *Romanian Journal of Information Technology and Automatic Control*, 32(3):95–108, 2022.

[109] Himanshukumar R. Patel and Vipul A. Shah. Fuzzy logic based metaheuristic algorithm for optimization of type-1 fuzzy controller: Fault-tolerant control for nonlinear system with actuator fault∗∗the author(s) received funding for the acods-2022 registration fees from dharmsinh desai university, nadiad-387001, gujarat, india. *IFAC-PapersOnLine*, 55(1):715–721, 2022. 7th International Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2022.

[110] Mohamed Salah Djebbar and H Benalla. The high performances of a seven levels active power filter with a fuzzy logic controller under a high voltage. *Journal of Engineering Science and Technology*, 13(5):1315–1329, 2018.

[111] Yousfi Laatra, Benayad Nizar, Aouiche Abdelaziz, and Djeddi Abdelghani. Advanced speed control of induction machine based on vector control. *AIJR Abstracts*, pages 110–111, 2024.

[112] Sohrab Mokhtari, Alireza Abbaspour, Kang K. Yen, and Arman Sargolzaei. Neural network-based active fault-tolerant control design for unmanned helicopter with additive faults. *Remote Sensing*, 13(12), 2021.

[113] AOUICHE Abdelaziz, SOUDANI Mouhamed Salah, and AOUICHE El Moundher. Efficient neural and fuzzy models for the identification and control of nonlinear systems. *AIJR Abstracts*, pages 10–12, 2024.

[114] PA Kolganov, AI Kondratiev, A Yu Tiumentsev, and Yu V Tiumentsev. Neural network nonlinear adaptive fault tolerant motion control for unmanned aerial vehicles. *Optical Memory and Neural Networks*, 31(1):1–15, 2022.

[115] Spencer M Richards, Navid Azizan, Jean-Jacques Slotine, and Marco Pavone. Adaptive-control-oriented meta-learning for nonlinear systems. *arXiv preprint arXiv:2103.04490*, 2021.

[116] Yan Ji, Zhen Kang, Xiao Zhang, and Ling Xu. Model recovery for multi-input signal-output nonlinear systems based on the compressed sensing recovery theory. *Journal of the Franklin institute*, 359(5):2317–2339, 2022.

[117] Anatoly Kulik. The principle of control by diagnosis as a result of the systematic application of fundamental control principles. *International Scientific Technical Journal" Problems of Control and Informatics"*, 68(1):7–22, 2023.

[118] Laatra YOUSFI. *Les algorithmes génétiques pour la caractérisation du tissu osseux sur radiographies osseuses*. PhD thesis, 2019.

[119] Davide Castagna. Combined grey-and black-box models with problem-tailored authority distribution. 2021.

[120] Shanlin Liu, Ben Niu, Guangdeng Zong, Xudong Zhao, and Ning Xu. Data-driven-based event-triggered optimal control of unknown nonlinear systems with input constraints. *Nonlinear Dynamics*, 109(2):891–909, 2022.

[121] Hanaa Elsherbiny, Laszlo Szamel, Mohamed Kamal Ahmed, and Mahmoud A. Elwany. High accuracy modeling of permanent magnet synchronous motors using finite element analysis. *Mathematics*, 10(20), 2022.

[122] Youcef Soufi, Sami Kahla, and Mohcene Bechouat. Feedback linearization control based particle swarm optimization for maximum power point tracking of wind turbine equipped by pmsg connected to the grid. *International journal of hydrogen energy*, 41(45):20950–20955, 2016.

[123] Youcef Soufi. Disturbance rejection control based sensorless fuzzy takagi-sugeno model for pmsm. *J. Electrical Systems*, 20(2):225–239, 2024.

[124] Yunfei Zhang, Can Zhao, Bin Dai, and Zhiheng Li. Dynamic simulation of permanent magnet synchronous motor (pmsm) electric vehicle based on simulink. *Energies*, 15(3), 2022.

[125] S Ghoudelbourk, D Dib, and A Omeiri. Decoupled control of active and reactive power of a wind turbine based on dfig and matrix converter. *Energy Systems*, 7(3):483–497, 2016.

[126] Saurav Gupta, Ajit Kumar Sahoo, and Upendra Kumar Sahoo. Volterra and wiener model based temporally and spatio-temporally coupled nonlinear system identification: A synthesized review. *IETE Technical Review*, 38(3):303–327, 2021.

[127] Luis Gustavo Giacon Villani, Samuel da Silva, and Americo Cunha Jr. An optimization-less stochastic volterra series approach for nonlinear model identification. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 44(6):260, 2022.

[128] Joseph Kelley and Martin T. Hagan. Comparison of neural network narx and narmax models for multi-step prediction using simulated and experimental data. *Expert Systems with Applications*, 237:121437, 2024.

[129] Rajintha Gunawardena, Zi-Qiang Lang, and Fei He. Nonsysid: Nonlinear system identification with improved model term selection for narmax models. *Journal of Open Source Software*, 10(114):8028, 2025.

[130] Fabio Di Nunno, Marco Race, and Francesco Granata. A nonlinear autoregressive exogenous (narx) model to predict nitrate concentration in rivers. *Environmental Science and Pollution Research*, 29(27):40623–40642, 2022.

[131] Weibin Gu, Kimon P Valavanis, Matthew J Rutherford, and Alessandro Rizzo. Uav model-based flight control with artificial neural networks: A survey. *Journal of Intelligent & Robotic Systems*, 100(3):1469–1491, 2020.

[132] Hossein Mirinejad and Tamer Inanc. An rbf collocation method for solving optimal control problems. *Robotics and Autonomous Systems*, 87:219–225, 2017.

[133] Jing-Jing Xiong, Xiang-Yu Wang, and Chen Li. Recurrent neural network based sliding mode control for an uncertain tilting quadrotor uav. *International Journal of Robust and Nonlinear Control*, 2025.

[134] XiuZe Xia and Long Cheng. Adaptive takagi-sugeno fuzzy model and model predictive control of pneumatic artificial muscles. *Science China Technological Sciences*, 64(10):2272–2280, 2021.

[135] Abdelghani Djeddi, Djalel Dib, Ahmad Taher Azar, and Salem Abdelmalek. Fractional order unknown inputs fuzzy observer for takagi–sugeno systems with unmeasurable premise variables. *Mathematics*, 7(10):984, 2019.

[136] José R. Rivera-Ruiz, José R. García-Martínez, Trinidad Martínez-Sánchez, Edson E. Cruz-Miguel, Luis D. Ramírez-González, Omar A. Barra-Vázquez, and Ákos Odry. Takagi–sugeno–kang fuzzy inference tracking controller for uav bicopter. *Symmetry*, 17(5), 2025.

[137] Yutoku Takahashi, Motoyasu Tanaka, and Kazuo Tanaka. Coordinated flight path generation and fuzzy model-based control of multiple unmanned aerial vehicles in windy environments. *International Journal of Fuzzy Systems*, 25(1):1–14, 2023.

[138] Huijia Liao, Guoliang Zhao, and Changsheng Xu. Quadrotor uav's takagi-sugeno fuzzy h-infinity filter design based on tensor fuzzy model transformation. In *2025 IEEE 8th International Conference on Signal Processing and Machine Learning (SPML)*, pages 1–8. IEEE, 2025.

[139] Ayad Al-Mahturi, Fendy Santoso, Matthew A. Garratt, and Sreenatha G. Anavatti. Chapter 2 - modeling and control of a quadrotor unmanned aerial vehicle using type-2 fuzzy systems. In Anis Koubaa and Ahmad Taher Azar, editors, *Unmanned Aerial Systems*, Advances in Nonlinear Dynamics and Chaos (ANDC), pages 25–46. Academic Press, 2021.

[140] Yunfeng Kang and Lina Yao. Fault diagnosis and fault-tolerant control for ts fuzzy time-varying delay stochastic distribution systems with actuator and sensor faults. *International Journal of Systems Science*, 52(11):2217–2227, 2021.

[141] Yue Wu, Yang Wang, Tieshan Li, and Zhixin Yang. A new fault tolerant control scheme for non-linear systems by t-s fuzzy model approach. *IET Control Theory & Applications*, 15(15):1915–1930, 2021.

[142] Youcef Soufi, Mohcene Bechouat, and Sami Kahla. Fuzzy-pso controller design for maximum power point tracking in photovoltaic system. *International Journal of hydrogen energy*, 42(13):8680–8688, 2017.

[143] T Bahi. Fuzzy logic control for vector control six phase induction machines. may. 2013.

[144] Tahar Bahi, Mohamed Fezari, George Barakat, and Nasr Eddine Debbache. Localization of faulty transistor in a three-phase inverter. *Asian journal of information technology*, 4(11):1068–1073, 2005.

[145] Syed Agha Hassnain Mohsan, Nawaf Qasem Hamood Othman, Yanlong Li, Mohammed H Alsharif, and Muhammad Asghar Khan. Unmanned aerial vehicles (uavs): Practical aspects, applications, open challenges, security issues, and future trends. *Intelligent service robotics*, 16(1):109–137, 2023.

[146] Mingyang Lyu, Yibo Zhao, Chao Huang, and Hailong Huang. Unmanned aerial vehicles for search and rescue: A survey. *Remote Sensing*, 15(13), 2023.

[147] Güray SONUGÜR. A review of quadrotor uav: Control and slam methodologies ranging from conventional to innovative approaches. *Robotics and Autonomous Systems*, 161:104342, 2023.

[148] Asif Ali Laghari, Awais Khan Jumani, Rashid Ali Laghari, and Haque Nawaz. Unmanned aerial vehicles: A review. *Cognitive Robotics*, 3:8–22, 2023.

[149] Moad Idrissi, Mohammad Salami, and Fawaz Annaz. A review of quadrotor unmanned aerial vehicles: applications, architectural design and control algorithms. *Journal of Intelligent & Robotic Systems*, 104(2):22, 2022.

[150] Syed Agha Hassnain Mohsan, Muhammad Asghar Khan, Fazal Noor, Insaf Ullah, and Mohammed H. Alsharif. Towards the unmanned aerial vehicles (uavs): A comprehensive review. *Drones*, 6(6), 2022.

[151] Faiyaz Ahmed, JC Mohanta, Anupam Keshari, and Pankaj Singh Yadav. Recent advances in unmanned aerial vehicles: a review. *Arabian Journal for Science and Engineering*, 47(7):7963–7984, 2022.

[152] Jinjin Guo, Juntong Qi, and Chong Wu. Robust fault diagnosis and fault-tolerant control for nonlinear quadrotor unmanned aerial vehicle system with unknown actuator faults. *International Journal of Advanced Robotic Systems*, 18(2):17298814211002734, 2021.

[153] George K. Fourlas and George C. Karras. A survey on fault diagnosis and fault-tolerant control methods for unmanned aerial vehicles. *Machines*, 9(9), 2021.

[154] Radosław Puchalski and Wojciech Giernacki. Uav fault detection methods, state-of-the-art. *Drones*, 6(11), 2022.

[155] Zhongyu Yang, Mengna Li, Ziquan Yu, Yuehua Cheng, Guili Xu, and Youmin Zhang. Fault detection and fault-tolerant cooperative control of multi-uavs under actuator faults, sensor faults, and wind disturbances. *Drones*, 7(8), 2023.

[156] T Bahi, S Lachtar, Y Soufi, S Lekhchine, and H Merabet. Speed control by sliding mode of synchronous motor. In *International Aegean Conference on Electrical Machines and Power Electronics and Electromotion, Joint Conference*, pages 386–391. IEEE, 2011.

[157] Nizar Benayad, Abdelaziz Aouiche, and Abdelghani Djeddi. Hybrid fault-tolerant sliding mode controller for unmanned aerial vehicle quadrotor based on optimized novel triple power reaching law. *Journal of Control, Automation and Electrical Systems*, 36(5):825–848, 2025.

[158] Nader Al-lQubaydhi, Abdulrahman Alenezi, Turki Alanazi, Abdulrahman Senyor, Naif Alanezi, Bandar Alotaibi, Munif Alotaibi, Abdul Razaque, and Salim Hariri. Deep

learning for unmanned aerial vehicles detection: A review. *Computer Science Review*, 51:100614, 2024.

[159] Promise Elechi and Kingsley Eyiogwu Onu. Unmanned aerial vehicle cellular communication operating in non-terrestrial networks. In *Unmanned Aerial Vehicle Cellular Communications*, pages 225–251. Springer, 2022.

[160] Dasom Lee, David J. Hess, and Michiel A. Heldeweg. Safety and privacy regulations for unmanned aerial vehicles: A multiple comparative analysis. *Technology in Society*, 71:102079, 2022.

[161] Linker Criollo, Carlos Mena-Arciniega, and Shen Xing. Classification, military applications, and opportunities of unmanned aerial vehicles. *Aviation*, 28(2):115–127, 2024.

[162] Shutong Huang, Ju Jiang, and Ouxun Li. Sliding mode backstepping control for the ascent phase of near-space hypersonic vehicle based on a novel triple power reaching law. *Aerospace*, 9(12):755, 2022.

[163] Lihua Wang, Siyuan Liu, Shaojie Jiang, Zengnan Zhao, Ning Wang, Xiaoyu Liang, and Minghui Zhang. A sliding mode control method based on improved reaching law for superbuck converter in photovoltaic system. *Energy Reports*, 8:574–585, 2022. The 2021 7th International Conference on Advances in Energy Resources and Environment Engineering.

[164] Zhenjie Gong, Xin Ba, Chengning Zhang, and Youguang Guo. Robust sliding mode control of the permanent magnet synchronous motor with an improved power reaching law. *Energies*, 15(5), 2022.

[165] Rajesh Kumar. A lyapunov-stability-based context-layered recurrent pi-sigma neural network for the identification of nonlinear systems. *Applied Soft Computing*, 122:108836, 2022.

[166] Nizar Benayad, Abdelaziz Aouiche, and Abdelghani Djeddi. *Artificial Neural Network Speed Controller for Squirrel Cage Induction Motor Based on Direct Torque Control*, pages 199–203. Springer Nature Switzerland, Cham, 2025.

[167] Zhenhua Yu, Zhijie Si, Xiaobo Li, Dan Wang, and Houbing Song. A novel hybrid particle swarm optimization algorithm for path planning of uavs. *IEEE Internet of Things Journal*, 9(22):22547–22558, 2022.

[168] Weizheng Zhang and Wei Zhang. An efficient uav localization technique based on particle swarm optimization. *IEEE Transactions on Vehicular Technology*, 71(9):9544–9557, 2022.

[169] Yu Tang, Kaicheng Huang, Zhiping Tan, Mingwei Fang, and Huasheng Huang. Multi-subswarm cooperative particle swarm optimization algorithm and its application. *Information Sciences*, 677:120887, 2024.

[170] Himanshu Gupta and Om Prakash Verma. A novel hybrid coyote–particle swarm optimization algorithm for three-dimensional constrained trajectory planning of unmanned aerial vehicle. *Applied Soft Computing*, 147:110776, 2023.

[171] Nizar Benayad, Taki Eddine Tria, and Laatra Encadre par Yousfi. *Application des techniques avancées pour la commande d'une machine asynchrone*. PhD thesis, 2020.

[172] Benayad Nizar, Aouiche Abdelaziz, and Djeddi Abdelghani. Advanced speed control of asynchronous ac motor based on direct torque control. In *2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC)*, pages 1–6, 2024.

Appendix A

Scientific Production

A.1 International Publications

1. Benayad, Nizar, Abdelaziz Aouiche, and Abdelghani Djeddi. "Hybrid Fault-Tolerant Sliding Mode Controller for Unmanned Aerial Vehicle Quadrotor Based on Optimized Novel Triple Power Reaching Law." Journal of Control, Automation and Electrical Systems 36, no. 5 (2025): 825-848.

2. Benayad, Nizar, Abdelaziz Aouiche, and Abdelghani Djeddi. "Fuzzy Adaptive Sliding Mode Control for Robust Fault Tolerant Operation of a UAV Quadrotor." Journal of Engineering Science & Technology Review 18, no. 5 (2025).

A.2 National Publications

1. Benayad, Nizar, Abdelaziz Aouiche, and Abdelghani Djeddi. "Fuzzy Logic Speed Controller for Permanent Magnet Synchronous Motor Based on Direct Torque Control." International Journal of Electronics and Electrical Engineering Systems IJEEES, Volume 7, Issue 1S (2024).

A.3 Indexed International Conferences

1. Laatra, Yousfi, Benayad Nizar, Aouiche Abdelaziz, and Djeddi Abdelghani. "Advanced speed control of induction machine based on vector control." AIJR Abstracts (2024): 110-111.

2. Benayad, Nizar, Abdelaziz Aouiche, and Abdelghani Djeddi. "Fuzzy Logic Speed Controller for Dual Stator Induction Motor Based On Indirect Vector Control." In IAM, pp. 126-135. 2023.

3. Nizar, Benayad, Aouiche Abdelaziz, and Djeddi Abdelghani. "Advanced Speed Control of Asynchronous AC Motor Based On Direct Torque Control." In 2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC), pp. 1-6. IEEE, 2024.

4. Benayad, Nizar, Abdelaziz Aouiche, and Abdelghani Djeddi. "Artificial Neural Network Speed Controller for Squirrel Cage Induction Motor Based on Direct Torque Control." In Technological and Innovative Progress in Renewable Energy Systems: Proceedings of the 2024 International Renewable Energy Days (IREN Days' 2024)., pp. 199-203. Cham: Springer Nature Switzerland, 2025.

A.4 National Conferences

1. Nizar Benayad, Aouiche Abdelaziz, and Djeddi Abdelghani. "Sliding Mode Control of Permanent Magnet Synchronous Machine." The 1st National Conference on maintenance and safety in the service of the environment (CNMSSE 2023), University of Tebessa, Algeria.

2. Nizar Benayad, Aouiche Abdelaziz, and Djeddi Abdelghani. "A Review on Applications of Artificial Intelligence in Medical Science." The Second National Conference on the Digital Revolution in Health: Applications and Impacts of Artificial Intelligence, University of Tebessa, Algeria.

3. Nizar Benayad, Aouiche Abdelaziz, and Djeddi Abdelghani. "Electric Vehicle Performance Improvement." The First National Conference on Innovation in Electronics and Telecommunication Technologies (NCIETT'2025), University of Tebessa, Algeria.