DEEP-LEARNING BASED MULTIPLE-MODEL BAYESIAN ARCHITECTURE FOR SPACECRAFT FAULT ESTIMATION

By

Rocío Jado Puente

A Thesis Submitted to the Faculty of Embry-Riddle Aeronautical University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Aerospace Engineering

August 2023

Embry-Riddle Aeronautical University

Daytona Beach, Florida

DEEP-LEARNING BASED MULTIPLE-MODEL BAYESIAN ARCHITECTURE

FOR SPACECRAFT FAULT ESTIMATION

By

Rocío Jado Puente

This Thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Hever Moncayo, Department of Aerospace Engineering, and has been approved by the members of the Thesis Committee. It was submitted to the Office of the Senior Vice President for Academic Affairs and Provost, and was accepted in the partial fulfillment of the requirements for the Degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE

Chair, Dr. Hever Moncayo	Member, Dr. Richard Prazenica
Member, Dr. K. Merve Dogan	
Graduate Program Coordinator, Dr. Daewon Kim	Date
Dean of the College of Engineering, Dr. James W. Gregory	Date
Associate Provost of Academic Support, Dr. Christopher Grant	Date

ABSTRACT

This thesis presents recent findings regarding the performance of an intelligent architecture designed for spacecraft fault estimation. The approach incorporates a collection of systematically organized autoencoders within a Bayesian framework, enabling early detection and classification of various spacecraft faults such as reaction-wheel damage, sensor faults, and power system degradation.

To assess the effectiveness of this architecture, a range of performance metrics is employed. Through extensive numerical simulations and in-lab experimental testing utilizing a dedicated spacecraft testbed, the capabilities and accuracy of the proposed intelligent architecture are analyzed. These evaluations provide valuable insights into the architecture's ability to detect and classify different types of faults in a spacecraft system.

The study has successfully implemented an intelligent architecture for detecting and classifying faults in spacecraft. The architecture was analyzed through numerical simulations and experimental tests, demonstrating enhanced early detection capabilities. The incorporation of autoencoders and Bayesian methods proved to be a powerful combination, allowing the architecture to effectively capture and learn from complex spacecraft system dynamics and detect various types of faults.

This research presents an advanced and reliable approach to early fault detection and classification in spacecraft systems, highlighting the potential of the intelligent architecture and paving the way for future developments in the field.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table **Page**

NOMENCLATURE

$6DOF$	System of Six Degrees of Freedom
$\hat{x}$	Reconstructed data
$A - MMAE$	Autoencoder-based Multiple-Model Adaptive Estimation
ANN	Artificial Neural Network
$EASY$	Extreme Access System
$FDIR$	Fault Detection, Isolation, and Recovery
K	Number of Models analyzed
KF	Kalman Filter
$MMAE$	Multiple-Model Adaptive Estimation
MSE	Mean Squared Error
$Prob_i$	Bayesian probability
$ReLU$	Rectified Linear Unit function
Res_i	Residuals (estimation error)
RW	Reaction Wheel
S_i	Covariance matrix
UAV	Unmanned Aerial Vehicles
UF	Unknown Failure
x	Input data

1 Introduction

Spacecraft Fault Estimation is a critical aspect of space exploration that requires robust and effective fault detection and isolation mechanisms. Navigating through the challenging conditions of space, characterized by high levels of uncertainty and the potential for catastrophic failures, demands reliable systems that can detect and mitigate faults in real-time. Traditional model-based approaches, such as Multiple-Model Adaptive Estimation (MMAE), have been widely utilized for fault detection and isolation in aerospace systems. These approaches rely on accurate mathematical models of the system to predict its behavior and detect anomalies. However, acquiring precise models can be challenging due to the complex dynamics and uncertainties inherent in space systems. Furthermore, model-based techniques may struggle to adapt to evolving failure modes and may not fully capture the intricate relationships within the system. Therefore, there is a growing need to explore data-driven techniques that can learn the system's normal behavior and detect anomalies without relying on detailed mathematical models.

1.1 Problem Statement

The problem addressed in this thesis is the development of a data-driven health management system for spacecraft fault estimation. While model-based approaches have been effective in certain cases, they heavily rely on accurate mathematical models, which may not always be available or reliable for complex aerospace systems. This limitation poses a challenge in accurately predicting and estimating failures within dynamic aerospace systems, where uncertainties and variations can significantly impact system behavior. To overcome this limitation, a novel architecture is proposed that utilizes autoencoders, a type of artificial neural network, to capture the system's normal behavior and identify anomalies. The objective is to improve the accuracy and efficiency of spacecraft fault estimation by leveraging the capabilities of deep learning and Bayesian methods, thereby reducing reliance on explicit mathematical models.

In this context, the proposed intelligent architecture is centered around the detection and

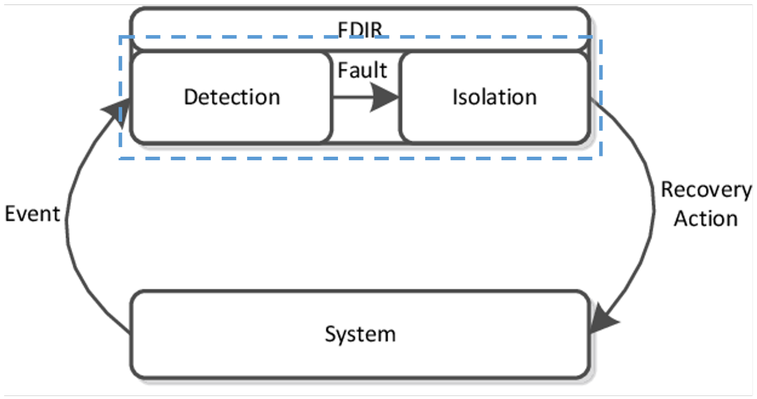

Figure 1.1 Interaction between system and FDIR from Müller [1].

isolation of failures in spacecraft systems, as depicted in Figure 1.1. By employing autoencoders and Bayesian methods, the architecture aims to accurately detect and classify various types of faults, enabling early detection and prompt response to potential system failures. The use of autoencoders allows for the learning and representation of complex relationships and patterns in the data, while Bayesian methods provide a probabilistic framework for robust fault estimation. Through extensive numerical simulations and in-lab experimental testing, the architecture's capabilities and accuracy are thoroughly evaluated, showcasing its potential for spacecraft fault estimation. The results obtained from this research contribute to the advancement of failure detection and isolation techniques in the aerospace domain, paving the way for enhanced spacecraft health management and improved mission success rates.

1.2 Thesis Objectives

The primary objective of this thesis is to develop a Deep-Learning Based Multiple-Model Bayesian Architecture for Spacecraft Fault Estimation. The specific objectives include:

1. Investigating fault detection and isolation techniques, both model-based and data-driven, to identify their advantages and disadvantages within the context of spacecraft fault estimation. This exploration will provide insights into the limitations of traditional model-based approaches and highlight the potential of data-driven techniques.

2. Designing an architectural framework that integrates autoencoders as estimators for fault estimation in spacecraft, replacing the traditional reliance on Kalman Filters. This framework will leverage the capabilities of deep learning to capture the system's normal behavior and detect anomalies, thereby enabling more accurate and timely fault estimation.

3. Conducting comprehensive simulations and empirical experimentation to validate the performance and effectiveness of the proposed architecture using real-world data. These experiments will involve training the autoencoders using time-history spacecraft data and evaluating their ability to accurately detect and isolate faults in different scenarios.

4. Analyzing and comparing the results obtained from the simulations and experiments to evaluate the accuracy, efficiency, and robustness of the proposed data-driven approach. This analysis will provide quantitative assessments of the proposed architecture's performance and its potential advantages over traditional model-based techniques.

1.3 Thesis Outline

This thesis is organized into the following chapters:

Chapter 2 provides a comprehensive review of Fault Detection, Isolation, and Recovery (FDIR) techniques used in aerospace systems. It explores both model-based and data-driven approaches, analyzing their strengths, limitations, and applications. The chapter highlights the need for improved fault estimation techniques and introduces the concept of integrating deep learning methodologies into FDIR systems. **Chapter 3** focuses on deep learning techniques applied to Fault Detection and Isolation (FDI) in aerospace systems. It explores the capabilities and advantages of deep learning, with a specific emphasis on autoencoders. The chapter discusses how autoencoders can be utilized to capture the normal behavior of the system, detect anomalies, and enhance fault estimation. In **Chapter 4**, the methodology for developing the Deep-Learning Based Multiple-Model Bayesian Architecture for Spacecraft Fault Estimation is presented. The chapter outlines the architectural framework that integrates autoencoders as estimators for fault estimation, replacing traditional model-based approaches. The design considerations, implementation details, and decision-making processes behind the proposed architecture are discussed. **Chapter 5** presents the results of numerical simulations conducted to evaluate the performance of the proposed architecture. The chapter analyzes the accuracy, efficiency, and robustness of the data-driven approach in spacecraft fault estimation. It provides a detailed assessment of the architecture's effectiveness in detecting and isolating faults under various scenarios, showcasing the benefits of utilizing deep learning techniques. **Chapter 6** describes the experimental verification of the proposed architecture using real flight data. It presents the data collection process, experimental setup, and the analysis of the results obtained. The chapter validates the performance of the data-driven approach using actual spacecraft data, highlighting its effectiveness in real-world scenarios and showcasing its potential for practical implementation. The final chapter, **Chapter 7** summarizes the key findings, conclusions, and contributions of the research. It discusses the significance of the proposed architecture in improving the accuracy and efficiency of fault detection and isolation. The chapter also suggests future research directions, potential enhancements, and opportunities for further development in the field of data-driven FDIR techniques.

By following this structured approach, the thesis provides a comprehensive exploration of FDIR techniques, the application of deep learning in fault estimation, the development of the proposed architecture, and the evaluation of its performance through numerical simulations and experimental verification. The research outcomes contribute to the advancement of fault estimation methodologies in aerospace systems, with implications for enhanced reliability, safety, and efficiency in space exploration missions.

2 Fault Detection, Isolation and Recovery (FDIR) Techniques

In the realm of space systems, the efficient operation and reliability of complex spacecraft are of paramount importance. To ensure the smooth functioning and mitigate potential risks, Fault Detection, Isolation, and Recovery (FDIR) techniques play a pivotal role. These techniques are designed to detect, locate, and mitigate faults that may arise during the operation of space systems.

The objective of FDIR techniques is to monitor the system's behavior, identify deviations from normal operation, and take appropriate actions to rectify or mitigate the impact of faults. These techniques are crucial in ensuring mission success, prolonging the operational lifespan of spacecraft, and safeguarding the safety of crew and mission-critical assets.

Various FDIR approaches have been proposed and employed in the aerospace industry. These techniques encompass a wide array of methods ranging from model-based approaches to data-driven approaches, leveraging the power of advanced technologies such as neural networks, machine learning, and Bayesian estimation.

One of the fundamental challenges in FDIR is to detect faults accurately while minimizing false alarms. This requires a comprehensive understanding of system behavior, fault signatures, and potential failure modes. Researchers have extensively studied FDIR techniques, resulting in a substantial body of literature encompassing different methodologies and approaches.

In recent years, there have been significant advancements in fault detection, isolation, and recovery (FDIR) techniques across various domains, including aerospace, automotive, and industrial systems. These techniques play a crucial role in ensuring the reliable and safe operation of complex systems. In the next subsections, we will elaborate on FDIR techniques, providing examples of recent studies in model-based, data-driven, and hybrid approaches.

2.1 Model-based FDIR techniques

Model-based fault diagnosis techniques have been extensively utilized in the design of fault diagnosis systems across various domains. These techniques leverage mathematical models of the system, incorporating knowledge of the system inputs and sensor measurements, to effectively detect, isolate, and diagnose faults. By formulating a set of equations that capture the relationships between inputs, outputs, and states of the system, model-based fault diagnosis systems compare measured outputs with predicted outputs to identify deviations indicative of faults. This approach offers robustness to measurement noise and the ability to verify and validate the models using established principles of system dynamics.

The study of fault detection has witnessed the incorporation of diverse methodologies, one of which being the utilization of Kalman Filters (KF). In the realm of fault detection, the utilization of KF pivots around their inherent capability to estimate the states of a dynamic system in the presence of noise. Despite noise corruption, the inherent optimization property of KF allows for effective fault detection by observing deviations in the estimated states from the expected values. While the presence of Gaussian noise is a common assumption in KF-based detection methods, these filters have shown considerable efficacy in detecting abnormalities even under non-Gaussian and non-linear circumstances, further cementing their versatility in fault detection [6, 7].

Complementing the role of Kalman Filters in fault detection, observers have demonstrated their pivotal importance in modern fault diagnosis systems. Serving as a continuous, real-time monitoring mechanism, the state observer methodology estimates the system states based on the system model and input-output measurements. An integral part of this technique lies in tracking discrepancies between the observed and measured system states, indicative of potential faults. The effective application of observers necessitates precise modeling of the system dynamics and meticulous handling of system noise. These prerequisites often lead to robustness issues, particularly in the face of modeling inaccuracies or external disturbances, thus requiring advanced observer designs to counteract these challenges [8, 9].

The implementation of the H-infinity method in fault detection exemplifies a different approach, deeply rooted in robust control theory. It aims to optimize the worst-case scenario by minimizing the maximum possible error, thereby ensuring a higher level of robustness against uncertainties and disturbances. H-infinity based fault detection formulates the fault detection problem as an optimization problem where the aim is to minimize the impact of the worst disturbances. While the application of this method requires complex computations and in-depth system understanding, the derived results often provide valuable insights into the robustness of the fault detection system in the presence of uncertainties [10, 11].

The application of model-based techniques is particularly advantageous due to their ability to handle complex system dynamics and ensure reliable fault detection and isolation. However, one of the key challenges associated with model-based FDIR is the requirement for prior knowledge about the system. Acquiring this knowledge can be time-consuming and costly, especially in the context of complex systems like space systems that exhibit intricate and non-linear behavior. Developing accurate models that capture the dynamics of such systems poses a significant challenge.

In recent years, researchers have made noteworthy advancements in addressing the challenges of model-based FDI for complex systems like multi-agent architectures, such as connected and autonomous vehicles (CAVs) [12, 13]. The utilization of advanced modeling techniques, such as nonlinear and adaptive models, has enabled more accurate representation of system dynamics. These models incorporate sophisticated algorithms that can adapt to changes in the system's behavior and handle non-linearities. Furthermore, the integration of machine learning algorithms, such as neural networks and deep learning models, with model-based approaches has enhanced fault diagnosis capabilities by leveraging the power of data-driven techniques to complement the established models.

Recent studies have demonstrated the effectiveness of model-based FDI in various domains. For example, in the field of robotics, model-based techniques have been applied to fault diagnosis in autonomous robots, enabling them to detect and isolate faults in their actuators or sensors. In the aerospace industry, model-based FDI plays a crucial role in ensuring the safety and reliability of aircraft systems by detecting and isolating faults in components such as engines, avionics, and flight control systems. Furthermore, in industrial applications, model-based FDI techniques are employed to monitor and diagnose faults in complex manufacturing processes, enhancing productivity and minimizing downtime.

In conclusion, model-based fault diagnosis techniques offer a powerful approach for detecting, isolating, and diagnosing faults in complex systems. While they require prior knowledge of the system dynamics, advancements in modeling techniques and the integration of data-driven methods have improved the accuracy and applicability of model-based FDI. These techniques have found widespread applications in autonomous systems, aerospace, robotics, and industrial domains, contributing to the overall safety, reliability, and performance of complex systems.

Recent studies have also explored the application of advanced techniques in FDIR. For example, Bayesian networks offer a powerful probabilistic modeling framework for capturing complex dependencies and uncertainties in spacecraft systems. The utilization of probabilistic graphical models, such as Bayesian networks, improves the accuracy and reliability of fault detection and isolation in space systems [14, 15].

Among these techniques, the Multiple-Model Adaptive Estimation (MMAE) method gained significant attention due to its high detection accuracy and robustness against system noise and disturbances. MMAE utilizes a set of candidate models and a statistical algorithm to continuously estimate the system's operating condition and detect faults in real-time [2, 16, 17].

The MMAE framework consists of N identification models running in parallel, each equipped with an optimized Kalman Filter to estimate the system state. These models capture different possible fault scenarios and the associated system behaviors. By calculating residuals, which measure the differences between predicted and measured system outputs, the MMAE algorithm probabilistically selects the model that best represents the current operating condition of the actual system. 2.1 illustrates a traditional MMAE structure.

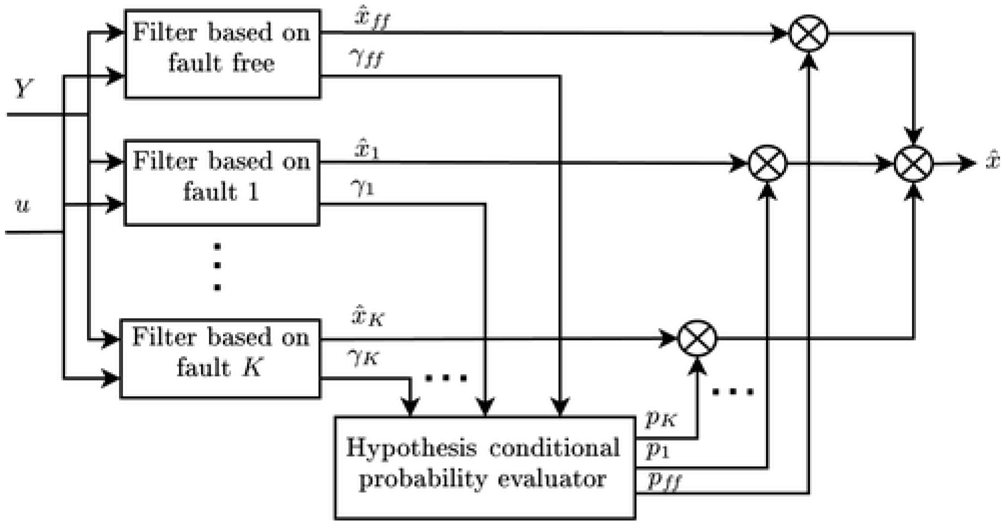

Figure 2.1 Block diagram for the MMAE approach from Lu et al. [2]

In the reference by Carbone et al. [18], the authors explore the application of the MMAE method for fault diagnosis in deep space power systems. The study focuses on identifying and isolating faults in power system components, such as solar arrays and batteries, which are critical for maintaining power generation and storage in space missions. The authors propose a multiple-model-based approach that combines the MMAE framework with a comprehensive set of power system models to enhance fault diagnosis capabilities.

This study highlights the effectiveness of the MMAE method in deep space power system fault diagnosis. By employing a diverse set of models that capture different fault scenarios, the MMAE approach successfully detects and isolates faults in the power system components. The authors emphasize the importance of accurate fault diagnosis in deep space missions to ensure the efficient operation and longevity of the power system, which is crucial for mission success.

This reference contributes to the existing body of research on MMAE and its application in fault diagnosis for deep space power systems. It demonstrates the practicality and effectiveness of the MMAE method in real-world scenarios, where the detection and isolation of faults are essential for maintaining the functionality and reliability of critical systems, underscoring the significance of advanced fault diagnosis techniques in aerospace applications, providing valuable insights for researchers and practitioners working in the field.

2.2 Data-driven and hybrid FDIR techniques

Data-driven fault diagnosis strategies involve analyzing the data generated by the system to detect and diagnose faults. This approach is useful when the system dynamics and failure modes are not fully understood or when the system behavior is highly nonlinear or complex. These methods use various data sources, such as sensor measurements or system logs, to build models using machine learning algorithms that can detect faults or anomalies. Data-driven fault diagnosis methods can be divided into two categories: supervised and unsupervised. In supervised methods, labeled data with known fault instances is used to train a model that can classify and identify faults in the system. These methods have the advantage of explicit knowledge about the fault instances, allowing for accurate fault detection and diagnosis. However, they rely on the availability of labeled data, which may not always be feasible or cost-effective to obtain, especially for rare or complex fault scenarios.

Support Vector Machine (SVM) is an example of a supervised learning method that has gained recognition in FDIR for its superior performance in high-dimensional spaces and its ability to generate complex decision boundaries. SVM operates by finding the hyperplane that maximally separates different classes of data in the feature space. In the context of FDIR, SVM can effectively distinguish between normal and faulty system conditions by training on labeled datasets, thus making it a supervised approach. A detailed examination of SVM can be found in the work by Cortes and Vapnik [19].

On the other hand, unsupervised methods operate without the need for labeled data and instead focus on identifying patterns and anomalies in the system's data. These methods leverage statistical techniques, clustering algorithms, or anomaly detection algorithms to detect deviations from normal system behavior, which may indicate the presence of faults. Unsupervised methods have the advantage of not requiring labeled data, making them applicable to a wider range of systems and fault scenarios. However, they may face challenges in accurately distinguishing between normal variations and actual faults, leading to potential false alarms or missed detections.

Principal Component Analysis (PCA), a machine learning technique, is frequently employed in FDIR owing to its dimensionality reduction capabilities and its ability to reveal relationships in multivariate data. PCA is particularly adept at modeling linear correlations within the data. In FDIR applications, the PCA technique works by projecting high-dimensional spacial data onto a lower-dimensional subspace that captures the most significant variations in the data, essentially extracting the principal components. Consequently, any deviation from these principal components can be indicative of an abnormal system behavior or fault. A notable work related to the use of PCA in FDIR is by Jolliffe and Cadima [20].

Independent Component Analysis (ICA) is another unsupervised, machine learning technique utilized for FDIR. Unlike PCA, which focuses on data uncorrelatedness, ICA emphasizes statistical independence. ICA works by separating a multivariate signal into additive independent components, thereby making it useful for isolating fault signals from normal operating signals in systems with high dimensionality. One of the key assumptions in ICA is that the independent components are non-Gaussian, which can pose a challenge in certain applications. A seminal reference on ICA is the work by Oja and Hyvarinen [21].

Blind Source Separation (BSS) is an unsupervised method often utilized to separate a set of source signals from a set of mixed signals without the aid of information (or with very little information) about the source signals or the mixing process. BSS is frequently employed in FDIR applications to distinguish fault conditions from normal operations in systems where little is known about the fault mechanism. The book by Comon and Jutten [22] provides a comprehensive look at BSS methodologies. This machine learning approach can be integrated synergistically with deep learning methodologies in order to augment its operational performance and predictive precision. This fusion of methods creates a more robust and adaptive framework capable of processing complex multivariate data streams

([23])

Both supervised and unsupervised data-driven fault diagnosis methods offer valuable insights into system health and can contribute to improved system reliability. The choice between the two approaches depends on the availability of labeled data, the complexity of the fault scenarios, and the desired trade-off between accuracy and computational complexity.

Hybrid approaches that combine both supervised and unsupervised techniques can also be explored to leverage the strengths of both methods and enhance fault diagnosis performance. In the study by Wang et al. [24], a model-based hybrid Fault Detection and Isolation (FDI) method for quadrotor unmanned aerial vehicles (UAVs) is proposed. The method employs a nonlinear model of the UAV dynamics and utilizes a sliding mode observer to estimate the UAV's state. By incorporating a statistical approach that analyzes the residuals of the observer, faults in the system can be detected and isolated effectively. The proposed method is evaluated through simulations using a quadrotor UAV model, demonstrating accurate and efficient fault detection and isolation capabilities.

In another study by Khoukhi and Khalid [25], a hybrid FDI technique is introduced, combining a mathematical model of the system with historical data. This hybrid approach integrates a model-based component that provides a theoretical understanding of the system's behavior with a data-driven component that captures the system's complex and nonlinear characteristics. By synergistically leveraging the strengths of both approaches, the hybrid system framework enhances the performance of FDI. It is worth noting that this approach shows promising results across various domains; however, it requires a comprehensive understanding of the system's behavior and may necessitate a sufficient amount of data to accurately estimate the model parameters.

3 Deep Learning Applied to FDIR

In recent years, the field of Fault Detection, Isolation, and Recovery (FDIR) has witnessed significant advancements through the integration of deep learning techniques. Deep learning, a subfield of artificial intelligence, has revolutionized various domains by enabling computers to learn from data and make intelligent decisions. Neural networks, the cornerstone of deep learning, mimic the structure and functioning of the human brain, consisting of interconnected artificial neurons that process and analyze information. These networks have the remarkable ability to learn from examples and improve their performance over time without explicit programming.

3.1 Neural Networks

An Artificial Neural Network (ANN) is a computational model inspired by the structure and functioning of biological neural networks, such as the human brain. It is composed of interconnected nodes, or "neurons", as the one illustrated in Figure 3.2, organized into layers. Each neuron receives input signals, performs a computation, and passes the result to other neurons. Figure 3.1 shows a typical structure of an Artificial Neural Network.

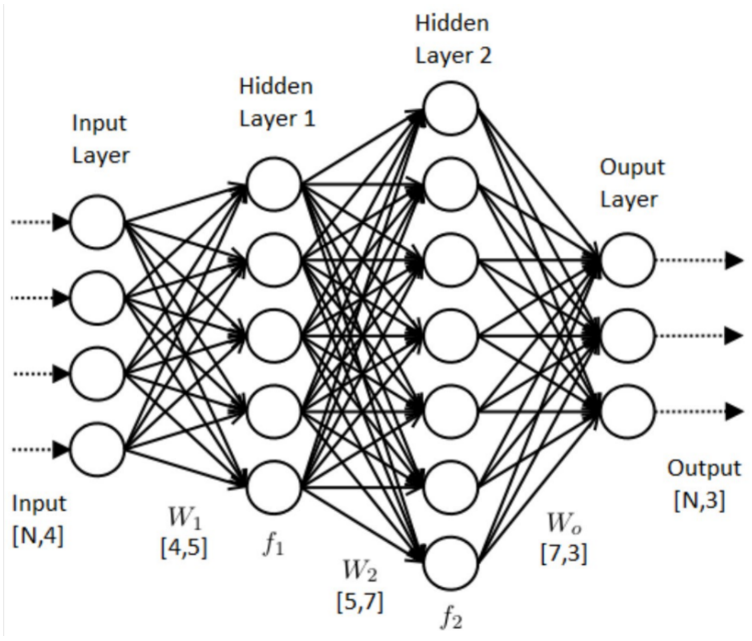

Figure 3.1 Example of an Artificial Neural Network (ANN) from web [3].

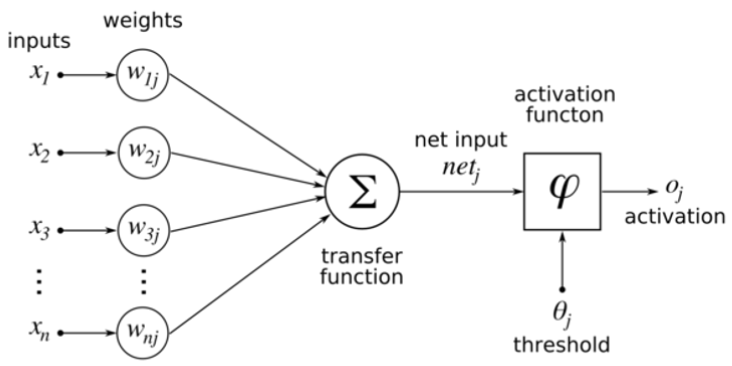

Figure 3.2 Example of an Artificial Neuron from Vochozka et al. [4]

At a high level, the functioning of a neural network involves two main processes: forward propagation and backpropagation. In forward propagation, input data is fed into the network, and computations are performed layer by layer to generate an output. The input data is multiplied by weights associated with the connections between neurons, and then passed through an activation function to introduce non-linear transformations. The output of one layer becomes the input for the next layer, and this process continues until the final output is obtained.

The activation function, typically applied to each neuron, introduces non-linearities and allows the neural network to model complex relationships between input and output. Commonly used activation functions include the sigmoid function, tanh function, and rectified linear unit (ReLU) function.

During forward propagation, the network's output is compared to the desired output using a loss function that quantifies the difference between the predicted and target values. The goal is to minimize this loss function, and that is where backpropagation comes into play. Backpropagation involves the calculation of gradients, which represent the sensitivity of the loss function to changes in network parameters (weights and biases). These gradients are used to update the weights and biases in the network, adjusting their values in a way that minimizes the loss function.

This iterative process of forward propagation followed by backpropagation is repeated multiple times, known as epochs, to improve the network's performance and reduce the loss.

The network learns by adjusting its weights and biases based on the gradients computed during backpropagation. This learning process allows the network to discover patterns, relationships, and representations in the data.

It is important to note that the architecture and design choices of a neural network, such as the number of layers, the number of neurons per layer, the activation functions, and the loss function, are crucial factors that impact the network's ability to learn and generalize from the data.

Neural networks have demonstrated remarkable success in various domains, including image and speech recognition, natural language processing, and autonomous systems. Their ability to learn complex patterns and relationships from data, combined with advancements in hardware and algorithms, has propelled the field of deep learning and paved the way for innovative applications in artificial intelligence.

3.1.1 Types of Neural Networks

Neural networks encompass a wide range of architectures, each designed to tackle specific types of problems and data. Here are some of the commonly used types of ANN:

- Feedforward Neural Networks (FNN): Feedforward neural networks are the most basic type of neural network, where information flows in one direction, from input nodes to output nodes. These networks are typically used for tasks such as classification and regression. An example of FNN is the Multilayer Perceptron (MLP) network. MLPs have been applied in various domains, including image classification, speech recognition, and financial forecasting.

- Convolutional Neural Networks (CNN): CNNs are widely used for image and video processing tasks. They consist of convolutional layers that automatically extract relevant features from the input data, enabling effective pattern recognition. CNNs have achieved remarkable success in image classification tasks, such as the famous application of ImageNet challenge winner, AlexNet, introduced by Krizhevsky et al. [26].

- Recurrent Neural Networks (RNN): RNNs are designed to process sequential data, such as time series or natural language. They have the ability to retain information from previous steps, making them suitable for tasks that involve temporal dependencies. One popular variant of RNNs is the Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber [27]), which addresses the vanishing gradient problem and has been applied to natural language processing tasks like language modeling and machine translation.

- Generative Adversarial Networks (GAN): GANs consist of a generator network and a discriminator network that compete against each other in a game-theoretic framework. GANs are used for generating new data that resembles the training data distribution. They have been successfully applied in image synthesis tasks, such as generating realistic images from random noise. Notable examples include the DCGAN (Deep Convolutional GAN) introduced by Radford et al. [28].

- Autoencoders: Autoencoders are unsupervised learning models that aim to learn efficient representations of input data. They consist of an encoder network that compresses the input data into a lower-dimensional latent space and a decoder network that reconstructs the original input from the compressed representation. Autoencoders have been used for various applications, including data reconstruction, anomaly detection, image denoising, and dimensionality reduction. Noteworthy examples include the Variational Autoencoder (VAE) by Kingma and Welling [29] and the Denoising Autoencoder (DAE) by Vincent et al. [30].

3.1.2 Autoencoders Used for Data Reconstruction

Autoencoders are a class of ANN widely used for data reconstruction and feature extraction tasks. They possess a distinctive architecture consisting of two primary components: an encoder and a decoder, as illustrated in Figure 3.3. The objective of an autoencoder is to learn a compact and meaningful representation of the input data by compressing it into a lower-dimensional latent space and subsequently reconstructing the original data from this compressed representation.

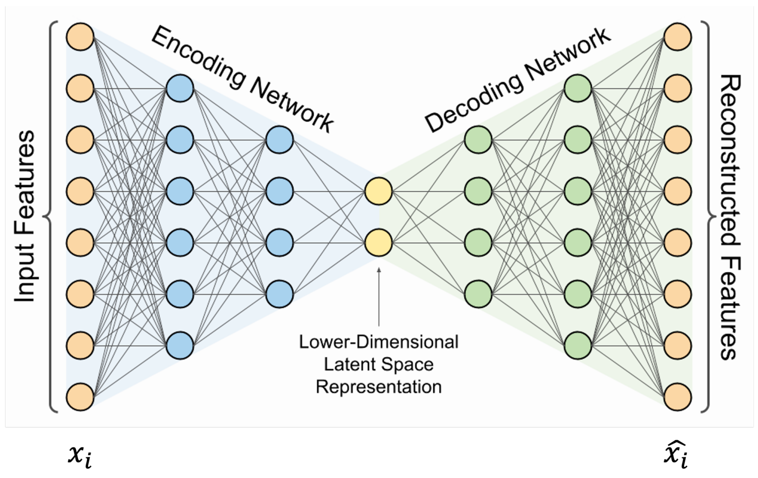

Figure 3.3 Structure of a Fully Connected Autoencoder from web [5].

The encoder component of an autoencoder is responsible for transforming the input data into a compressed representation. It achieves this by applying a series of non-linear transformations to reduce the dimensionality of the data. The encoder transforms the input data $\mathbf{x}$ into a lower-dimensional latent representation $\mathbf{z}$. Mathematically, the encoder can be represented as:

$$\mathbf{z} = f(\mathbf{W}_e \cdot \mathbf{x} + \mathbf{b}_e) \tag{3.1}$$

$\mathbf{W}_e$ represents the weight matrix of the encoder, $\mathbf{b}_e$ is the bias vector, and f is the activation function.

The decoder component plays a vital role in the autoencoder architecture by reconstructing the original input data from the compressed representation obtained from the encoder. The reconstructed data is represented as $\hat{\mathbf{x}}$. The decoder takes the compressed representation as input and applies a series of transformations to upsample and expand the data back to its original dimensionality. Mathematically, the decoder can be represented as:

$$\hat{\mathbf{x}} = g(\mathbf{W}_d \cdot \mathbf{z} + \mathbf{b}_d) \tag{3.2}$$

$\mathbf{W}_d$ represents the weight matrix of the decoder, $\mathbf{b}_d$ is the bias vector, and g is the activation function.

In both the encoder and decoder, the activation functions introduce non-linearities that enable the neural network to learn complex mappings between the input data and the latent representation, as well as between the latent representation and the reconstructed data. Common activation functions used in autoencoders include the sigmoid function, hyperbolic tangent (tanh) function, and ReLU function.

During training, an autoencoder minimizes the reconstruction error between the input data and the reconstructed output. This is achieved by comparing the reconstructed data with the original input data using a suitable loss function, such as mean squared error (MSE). The network's parameters, including the weights and biases of the encoder and decoder, are adjusted through optimization algorithms, such as gradient descent, to minimize the loss and enhance the reconstruction performance.

4 Autoencoder-based MMAE (A-MMAE)

In alignment with the conventional structure of Multiple Model Adaptive Estimation (MMAE), the objective of this thesis is to introduce a novel approach that replaces the conventional model-based estimators, namely the Kalman Filters (KFs) (Figure 2.1), with a collection of autoencoders. These autoencoders are meticulously trained to capture the distinct characteristics and patterns associated with a diverse range of flight conditions, encompassing both nominal and failure data. Within the ensemble of autoencoders, each model is equipped with a trained policy to serve as the state estimator, enabling the extraction of residuals by subtracting the system's states from the predicted states. By adopting this parallel configuration of autoencoder-based estimators, the proposed architecture operates in a fully data-driven manner, thereby alleviating the necessity for comprehensive knowledge of the system dynamics.

4.1 Architecture of Autoencoders

As commented in Chapter 3, Auto-encoders are composed of densely connected neural layers, which implies the use of specific mathematical functions to model the behavior of neurons and the operations performed on each layer. The following describes some of the key mathematical functions that govern the behavior and performance of dense auto-encoders in data reconstruction:

- Activation function: The activation functions play a crucial role in the autoencoder architecture as they are applied to the outputs of the neurons in each layer. In this study, two commonly employed activation functions, namely the Rectified Linear Unit (ReLU) function and thehyperbolic tangent function (tanh), were utilized to introduce nonlinearity into the model. The choice of activation function depended on the specific requirements of the autoencoder. The ReLU function, given by:

$$Relu(z) = max(0, z) \tag{4.1}$$

was predominantly employed for attitude-related features, while the tanh function, given by:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{4.2}$$

was predominantly used for the remaining features. This distinction in activation functions was made based on the characteristics of the data and the desired performance of the model.

- Loss function: The loss function quantifies the difference between the output generated by the auto-encoder and the original input data. The choice of loss function depends on the type of data and the specific goal of the model. For example, for continuous data reconstruction problems, mean square error (MSE) is the most used:

$$L_{AE} = \sum_{i=1}^{n} (x_i - \hat{x_i})^2 \tag{4.3}$$

- Optimization function: The optimization function defines how the auto-encoder weights and parameters are adjusted during the training process. Some common optimization functions are stochastic gradient descent (SGD), Adam optimizer, and RMSprop optimizer. These functions allow finding the optimal values of the model parameters to minimize the loss function. This study found using Adam optimization minimized their loss function in little time.

The decision to employ the tanh function for high feature volume datasets with negative values was motivated by its ability to handle and represent such data effectively. Furthermore, the use of tanh activation in these cases contributed to improved model accuracy and reconstruction performance, resulting in lower residuals. Conversely, the ReLU activation function was chosen for attitude-related features due to its ability to capture and preserve the non-linear relationships inherent in such data. This strategic selection of activation functions not only optimized the performance of the autoencoders but also reduced the computational time required for feature reconstruction.

Moreover, by splitting the set of autoencoders based on common features, such as angular rates, attitude angles, and torques or actuations, the overall time complexity of the computation was significantly reduced. This partitioning facilitated the parallel processing of these common features, leading to enhanced efficiency in reconstructing the respective data. Additionally, this approach augmented the accuracy of the model and the fidelity of the reconstruction process, thereby reducing the residuals and improving the overall performance of the data-driven architecture.

4.2 Model-based Estimators Replacement

Figure 4.1 underlines in blue the replacement of Kalman Filter (KF) estimators with autoencoders within the Multiple-Model Adaptive Estimation (MMAE) architecture proposed. It is a seamless process facilitated by the fundamental equivalence between the outputs of these two models. While KF estimators are traditionally employed to obtain estimated states for each defined model, autoencoders are trained to reconstruct data, effectively estimating the original input. As such, the outputs of the autoencoders can be considered equivalent to the estimated states obtained from KF estimators.

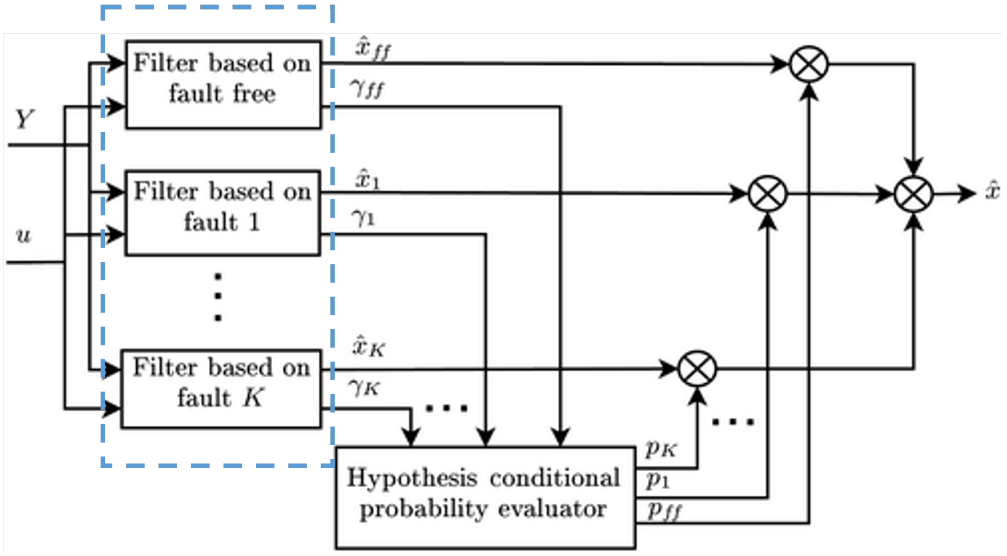

Figure 4.1 Proposed modification of the block diagram for the MMAE approach from Lu et al. [2]

To integrate this set of autoencoders into the MMAE architecture, only the residuals, also known as the estimation errors, need to be extracted by subtracting the original states from the reconstructed states:

$$Res_i = \mathbf{x_i} - \hat{\mathbf{x}}_\mathbf{i} \qquad i = 1, 2, ..., K \tag{4.4}$$

where K is the number of models analyzed. These residuals serve as the key input for computing the Bayesian probabilities of the system, which are essential for the fault detection and isolation process. Notably, the covariance matrix plays a critical role in this computation, as highlighted in the equations from Chapter 2 (4.5, 4.6). The covariance matrix serves as a free parameter that can be adjusted to optimize the accuracy and performance of the system.

The replacement of the Kalman Filters with autoencoders occurs once the residuals from the autoencoders have been obtained. Despite the initial perception of complexity, the transition is remarkably straightforward and yields significant benefits in terms of computational efficiency. By leveraging the capabilities of autoencoders for data reconstruction and the subsequent extraction of residuals, the overall computational time is reduced while preserving the accuracy and efficacy of the fault estimation process. This replacement showcases the seamless integration of deep learning techniques into the established MMAE framework, paving the way for enhanced fault detection and isolation in complex systems.

4.3 Proposed Architecture

The proposed architecture subsequent to the replacement of the previous Kalman Filter estimators with autoencoders is now elucidated. Figure 4.2 establishes the structure that will be later examined. To this end, four distinct models are defined, each corresponding to a unique system behavior characterized by a specific dataset and associated autoencoder.

4.3.1 Description of Models

Model 1 represents the system's non-fault behavior, effectively capturing the nominal operational state. Given the project's objective of detecting abnormal behavior, the con-

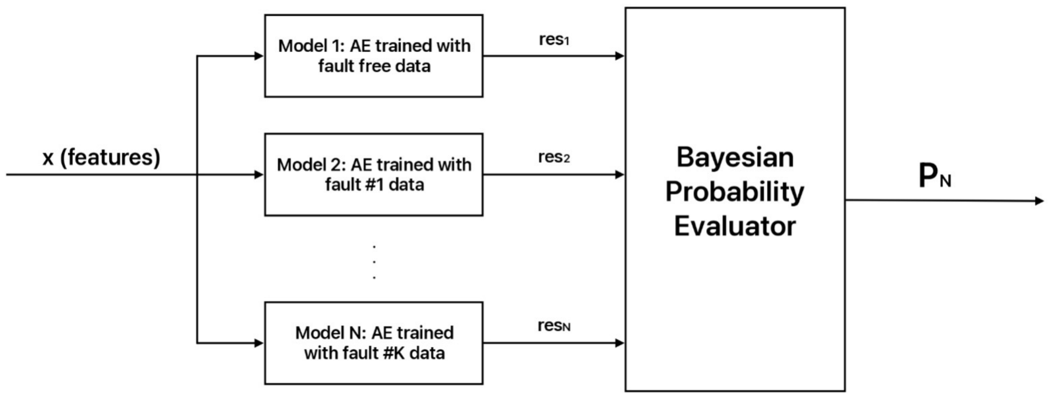
x (features)
Model 1: AE trained with fault free data
Model 2: AE trained with fault #1 data
Model N: AE trained with fault #K data
res1
res2
resN
Bayesian Probability Evaluator
PN

Figure 4.2 Block diagram for the proposed A-MMAE approach.

struction of a well-trained nominal model assumes paramount significance. The residuals obtained from the reconstruction of new nominal data should ideally approach zero, thereby attesting to the accuracy and reliability of the autoencoder in capturing the system's expected behavior.

Subsequently, a set of three known failures is introduced and their associated data is collected for the purpose of constructing the remaining models. The rationale behind this selection lies in the necessity for the Autoencoder-based Multiple-Model Adaptive Estimation (A-MMAE) framework to successfully extrapolate its predictive capabilities to unknown failures that share similar characteristics with the known ones.

Upon defining the models and assembling the bank of estimators, the system's probabilities of exhibiting specific behaviors are calculated for each of the four models simultaneously. These probabilities are expected to equal 1 when the system adheres to the behavior represented by the respective model. In cases where the probabilities fall below 1 but exceed 0, it indicates the presence of an unknown behavior, potentially resembling the failure patterns observed, and favoring the model with the highest probability as the most plausible approximation.

The subsequent two chapters of this thesis will delve into detailed examples illustrating the implementation and evaluation of the proposed architecture. These chapters encompass both simulation-based experiments as well as real-world flight data analysis, offering comprehensive insights into the practical application and performance of the A-MMAE framework.

4.3.2 Computation of Bayesian Probabilities

In order to ascertain whether the system is manifesting a fault behavior, *i.e.* if it aligns with one of the last three defined models, the employed approach in the context of the A-MMAE framework involves the computation of the probability that the system's behavior matches a given defined model. This quantifiable measure enables the assessment of potential fault conditions within the system's operation.

For the purpose of obtaining said Bayesian probabilities, it is imperative that the covariance matrix for the system is first ascertained. The covariance matrix, S often denoted by P, characterizes the uncertainty or error in the estimation of the system's state. In other words, it provides a quantification of the estimated accuracy of the state estimate.

$$S_i = CP_{ss}C^T + R \qquad B_i = \frac{1}{(2\pi)^{\frac{l}{2}} S_i^{\frac{1}{2}}} \tag{4.5}$$

where l is the number of measurements for each system, P_{ss} is the steady-state covariance of the estimation, and R is the covariance measurement.

Additionally, an approximation of the covariance matrix can be performed. In the context of this thesis, the covariance matrix will be treated as a free variable.

Once the covariance matrix of the system has been determined, the subsequent step involves the calculation of Bayesian probabilities. These probabilities represent the likelihood of a model matching a given model based on a constraint sourced from input and output measurements.

$$Prob_i(k) = \frac{B_i e^{-0.5[Res_i^T(k) S_i^{-1} Res_i(k)]} Prob_i(k-1)}{\sum_{j=1}^{K} B_j e^{-0.5[Res_j^T(k) S_j^{-1} Res_j(k)]} Prob_j(k-1)} \tag{4.6}$$

5 Numerical Simulation

In order to comprehensively evaluate the accuracy and efficacy of the proposed architecture, a detailed analysis of numerical simulations will be conducted. This investigation will provide valuable insights into the performance of the Autoencoder-based Multiple-Model Adaptive Estimation (A-MMAE) framework under controlled conditions. By subjecting the architecture to various simulated scenarios encompassing both nominal and fault behaviors, we aim to assess its ability to accurately detect and classify system anomalies.

Through the utilization of state-of-the-art simulation techniques, a wide range of operating conditions and failure scenarios will be emulated, enabling a comprehensive exploration of the A-MMAE framework's capabilities. By carefully designing and executing these simulations, we can effectively assess the accuracy of the architecture in detecting and isolating failures, as well as its overall robustness and reliability.

This numerical simulation-based analysis will provide valuable quantitative data and insights into the performance characteristics of the A-MMAE framework. The results obtained from these simulations will serve as a crucial foundation for further refining and optimizing the architecture, ensuring its efficacy in real-world applications. Furthermore, these findings will contribute to the overall understanding and advancement of fault detection and isolation techniques in complex dynamic systems, particularly in the context of aerospace engineering.

In this chapter, a detailed overview of the simulation methodology will be presented, including the simulated scenarios, data generation process, and performance evaluation metrics. Through an analysis of these numerical simulations, the proposed architecture is validated and refined, paving the way for its successful implementation in practical applications.

5.1 Case of Study: On-Orbit Inspection of Space Assets with a CubeSat Fleet. Mission Description. Simulation environment.

The proposed mission involves deploying a fleet of eight CubeSats for on-orbit inspection of a target space asset in low Earth orbit (LEO), as shown in Figure 5.1. These CubeSats will be launched from standard dispensers and placed in passive relative orbits (PROs) around the target spacecraft. Equipped with sensors for inspection, each CubeSat's sensor pointing will be precisely controlled by the Attitude Determination and Control System (ADCS), allowing for exceptional accuracy of 0.003 degrees per axis.

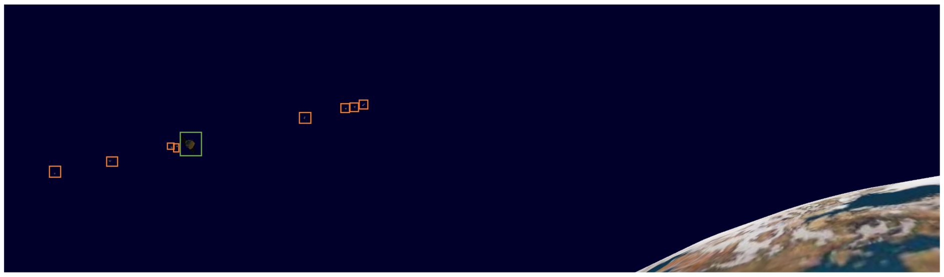

Figure 5.1 Disposition of CubeSats fleet around chief.

To optimize fuel consumption, the CubeSats will utilize stable PROs throughout the inspection, only transferring to different PROs when the information gain justifies the fuel cost of the transfer. Additionally, the CubeSats will be equipped with a cold/warm propulsion unit that enables reconfiguration to different PROs while minimizing fuel expenditure.

Close coordination is essential for the collective inspection task performed by the CubeSat fleet. Constant communication among the CubeSats ensures information sharing regarding points of interest (POIs) on the target spacecraft. This shared data is used to make global decisions on which POIs each CubeSat should focus on in the next planning horizon. It is crucial to minimize delays and potential misinformation in communication, as any degradation or intentional attacks on POI information could adversely impact the overall performance of the coordinated inspection.

The proposed mission serves as a blueprint for future on-orbit inspection missions. With further development, this mission concept can extend the lifetime of space assets, enhance space safety, and enable new space exploration capabilities.

The simulation environment utilized in this research is designed with modularity and flexibility in mind. It enables the simulation of multi-spacecraft missions, processing of flight data under normal and abnormal conditions, testing of various failure and disturbance scenarios, and conducting risk assessments associated with such missions. It is based on a comprehensive 6DOF system that incorporates the interaction between the spacecraft's power plant system and the reaction wheel model. The inputs to the power plant system are comprised of external forces, external moments, and the torques produced by the reaction wheels.

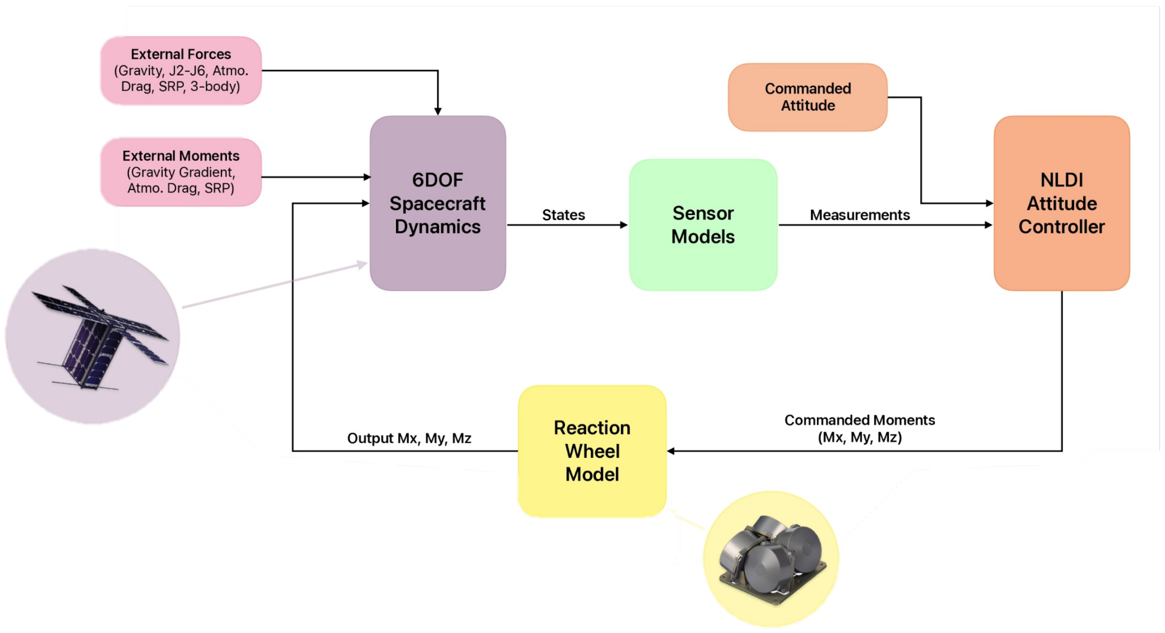
External Forces (Gravity, J2-J6, Atmo. Drag, SRP, 3-body)
External Moments (Gravity Gradient, Atmo. Drag, SRP)
6DOF Spacecraft Dynamics
States
Sensor Models
Measurements
Commanded Attitude
NLDI Attitude Controller
Commanded Moments (Mx, My, Mz)
Reaction Wheel Model
Output Mx, My, Mz

Figure 5.2 Overview of the Simulation Environment employed.

Within this simulation environment, the primary focus lies on the attitude sensors, which provide measurements of the spacecraft's orientation. These measurements are then compared to the desired attitude, represented by the commanded quaternions. This assessment enables a thorough evaluation of the spacecraft's actual orientation in relation to the desired orientation.

To achieve the desired attitude, a quaternion-based controller is employed. This controller generates torque commands that are applied to the reaction wheel model. These torques, combined with the external forces and moments acting on the spacecraft, drive the power plant system and contribute to the adjustment and stabilization of the spacecraft's attitude.

A graphical overview of the primary elements of this simulation environment, and their interconnected relationships, can be viewed in Figure 5.2. The simulation utilizes a time step of dt = 0.1s for its operations.

5.1.1 Reaction Wheel Model

The focus is now directed towards the Reaction Wheel model. In the realm of Attitude Control Systems (ACS), momentum exchange systems like Reaction Wheels are often favored due to their non-reliance on fuel and the precision they offer through a distinct blend of torque and momentum storage, vital for achieving pointing accuracy. Despite the presence of disturbance torques or failures, the ultimate objective of ACS is to maintain system stability and guide it towards the desired attitude. Consequently, the provision of a high fidelity mathematical model of the Reaction Wheel configuration aids in deriving realistic responses for evaluating distinct failures within the ACS subsystem.

This Reaction Wheel model adheres to a redundant structure, encompassing a quartet of Reaction Wheels that are arrayed in a tetrahedral configuration, as shown in Figure 5.3.

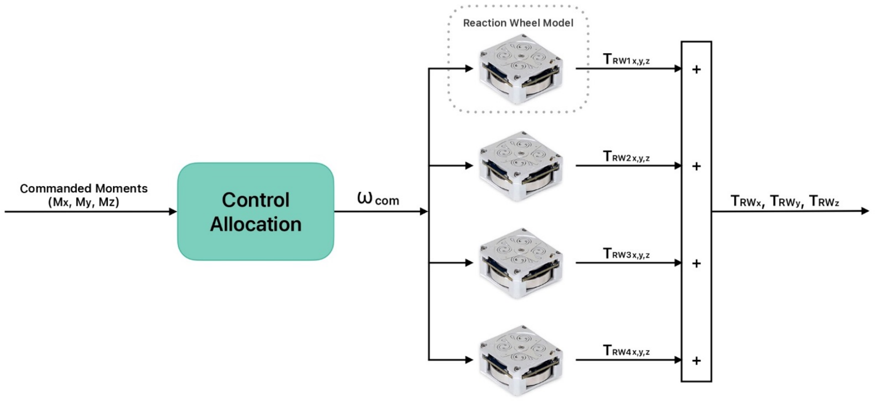

Figure 5.3 Reaction Wheel Model.

It portrays the control scheme of the four Reaction Wheels, in which the input - a commanded voltage - governs the motor current, subsequently influencing the angular velocity of the wheel derived from the desired moment. Figure 5.4 presents the components of the Reaction Wheel Model, including the DC Motor, Friction, Rotational Imbalances, and Thermals. Notably, Friction plays a role in the generation of torque loss within the system, which is then deducted from the overall torque.

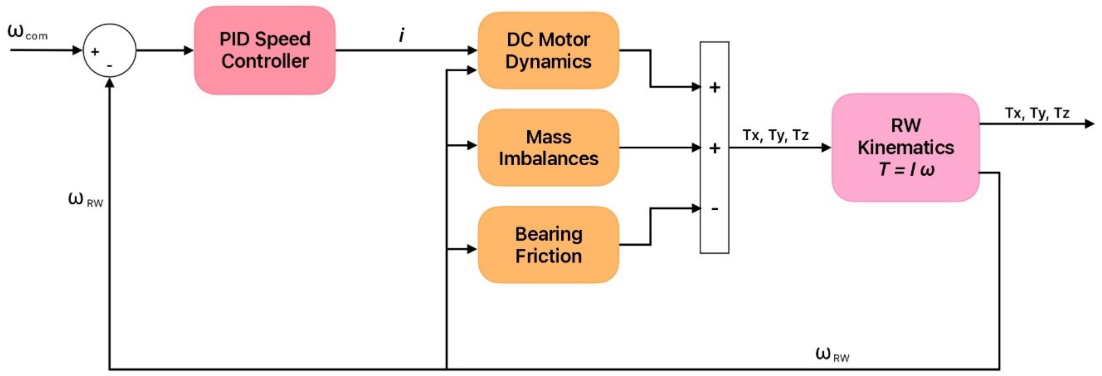

Figure 5.4 Components of the Reaction Wheel Model.

5.2 Failure Scenarios

For the purpose of this thesis, the author has chosen to concentrate on a single spacecraft's detection of possible failures within its attitude control system, comprising four reaction wheels. The spacecraft will be subjected to various scenarios during its commanded orbit, with three failures intentionally injected into its attitude control system for analysis.

In the context of reaction wheels, various types of failures can occur, which can significantly impact the performance and stability of a spacecraft's attitude control system. In this research, three specific failures have been intentionally injected into the system for analysis and evaluation.

- **Failure Case I** involves a current dropout in reaction wheel 1, which occurs at time t=0s. This failure disrupts the power supply RW#1, resulting in a loss of its rotational capabilities.
- **Failure Case II**, also occurring at time t=0s, involves reaction wheel saturation. Saturation refers to a situation where the torque demand exceeds the maximum capacity of the reaction wheel, 0.004 N-m. This failure leads to compromised control authority and can severely affect the spacecraft's ability to maintain the desired attitude.
- **Failure Case III**, again at time t=0s, is characterized by a current dropout in two

reaction wheels. This failure affects the power supply to both reaction wheels simultaneously, resulting in a loss of control torque from these wheels.

Testing failures in more than one reaction wheel is of paramount importance for several reasons. Firstly, a spacecraft's attitude control system relies on redundancy, which means it can operate even if one or more reaction wheels fail. Understanding the behavior and implications of multiple failures allows for the development of robust fault detection and isolation algorithms to handle complex scenarios.

Furthermore, injecting failures in the reaction wheels provides valuable insights into the system's fault tolerance and resilience. By studying the effects of various failures, it becomes possible to design and implement appropriate mitigation strategies, such as utilizing backup reaction wheels or redistributing control authority among the functioning wheels.

Testing these specific failures in a spacecraft's attitude system is particularly significant because they represent common failure modes that can occur in reality. Current dropouts are often encountered due to electrical issues or malfunctions, while saturation can result from unexpected disturbances or high demand maneuvers. Assessing the system's response to these failures helps in developing strategies to detect, isolate, and recover from such situations, ultimately enhancing the reliability and performance of spacecraft attitude control systems in real-world scenarios.

5.3 Results

The features shown in Table 5.1 were selected as primary features. Sensor measurements, Euler angles, quaternions, and reaction wheel torques are analyzed to improve the accuracy of the Health Management System (HMS) performance analysis and to identify potential failures earlier. These features were selected to capture different types of failures.

For example, the sensors measurements can be used to detect sensor failures, as well as failures in the system's control algorithms. The Euler angles and quaternions can be used to detect attitude failures, while the reaction wheel torques can be used to detect actuator failures.

Table 5.1 Selected features and states from Spacecraft.

	Feature	Description
1	p	Roll Rate, deg/s
2	q	Pitch Rate, deg/s
3	r	Yaw Rate, deg/s
4	ϕ	Roll angle, deg
5	θ	Pitch angle, deg
6	ψ	Yaw angle, deg
7	$q0_{actual}$	Quaternion actual q0
8	$q1_{actual}$	Quaternion actual q1
9	$q2_{actual}$	Quaternion actual q2
10	$q3_{actual}$	Quaternion actual q3
11	$T_{x_{RW1}}$	Torque in x axis in $_{RW1}$
12	$T_{y_{RW1}}$	Torque in y axis in $_{RW1}$
13	$T_{z_{RW1}}$	Torque in z axis in $_{RW1}$
14	$T_{x_{RW2}}$	Torque in x axis in $_{RW2}$
15	$T_{y_{RW2}}$	Torque in y axis in $_{RW2}$
16	$T_{z_{RW2}}$	Torque in z axis in $_{RW2}$
17	$T_{x_{RW3}}$	Torque in x axis in $_{RW3}$
18	$T_{y_{RW3}}$	Torque in y axis in $_{RW3}$
19	$T_{z_{RW3}}$	Torque in z axis in $_{RW3}$
20	$T_{x_{RW4}}$	Torque in x axis in $_{RW4}$
21	$T_{y_{RW4}}$	Torque in y axis in $_{RW4}$
22	$T_{z_{RW4}}$	Torque in z axis in $_{RW4}$

To emulate the Kalman filters used in the Multiple Model Adaptive Estimation (MMAE) approach, several models were created with tailored autoencoders. These models were designed to capture and reconstruct data from different scenarios. Specifically, Model 1 was constructed using Nominal data, Model 2 utilized data from Failure Case I, Model 3 incorporated data from Failure Case II, and Model 4 was trained on data from Failure Case III.

5.3.1 Residual Calculation and Analysis

To train and validate these models, data sets were acquired by running simulations for each specific case. The spacecraft executed a commanded orbit for a duration of 1050 seconds. For each model, the first 1000 seconds were allocated for training and validation of each autoencoder, while the last 50 seconds were reserved for testing.

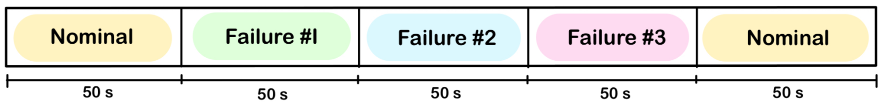

Figure 5.5 Disposition of the data set used in the numerical simulation to detect failures.

To create a comprehensive data set that encompasses various scenarios, the final 50 seconds of each model's simulation output were combined. As shown in Figure 5.5, this resulted in a merged data set consisting of 50 seconds of Nominal data, followed by 50 seconds of Failure Case I data, 50 seconds of Failure Case II data, 50 seconds of Failure Case III data, and concluding with another 50 seconds of Nominal data. Notably, all 250 seconds of data originated from the spacecraft performing the same commanded quaternions, ensuring consistency across the merged set.

By organizing the data sets in this manner, it becomes possible to evaluate the performance and effectiveness of the autoencoder-based models in distinguishing between nominal and failure scenarios. The merged data set offers a comprehensive representation of the spacecraft's behavior under different conditions, enabling thorough testing and assessment of the MMAE approach for fault detection and estimation in the attitude control system.

The nature of failure 3, which involves the malfunction of two reaction wheels, is characterized by its aggressive impact, resulting in a significant change in the spacecraft's attitude, as illustrated in Figure 5.6. At t=150s, there is an abrupt change of attitude. This distinctiveness is reflected in the plots illustrating the residuals for all models.

In Figures 5.7, 5.8, 5.9, and 5.10, the time histories of residuals are illustrated when the validation data, which remained unused during training, was processed through Model 1, Model 2, Model 3, and Model 4.

A priori, Models 1, 2, and 3 seem capable of approximating nominal conditions, Failure 1, and Failure 2. For enhanced clarity, Figures 5.11, 5.12, and 5.13 spotlight the disparities among residuals in Model 1, Model 2, and Model 3. From each residual, the values pertaining to the 50 seconds of the data used for autoencoder training in each scenario have been

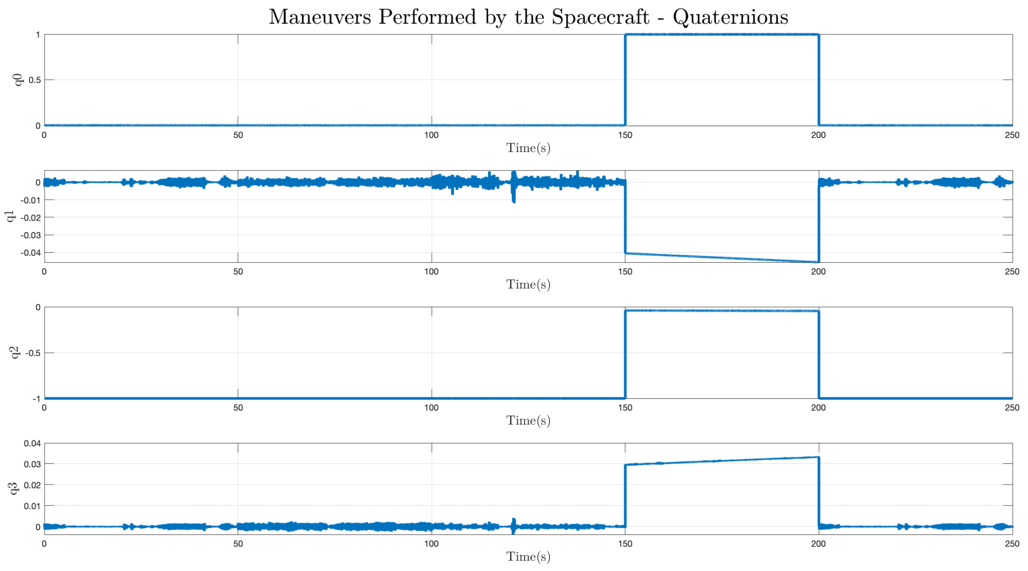

Figure 5.6 Maneuvers performed by the Spacecraft following 50 seconds of the same orbit under different scenarios.

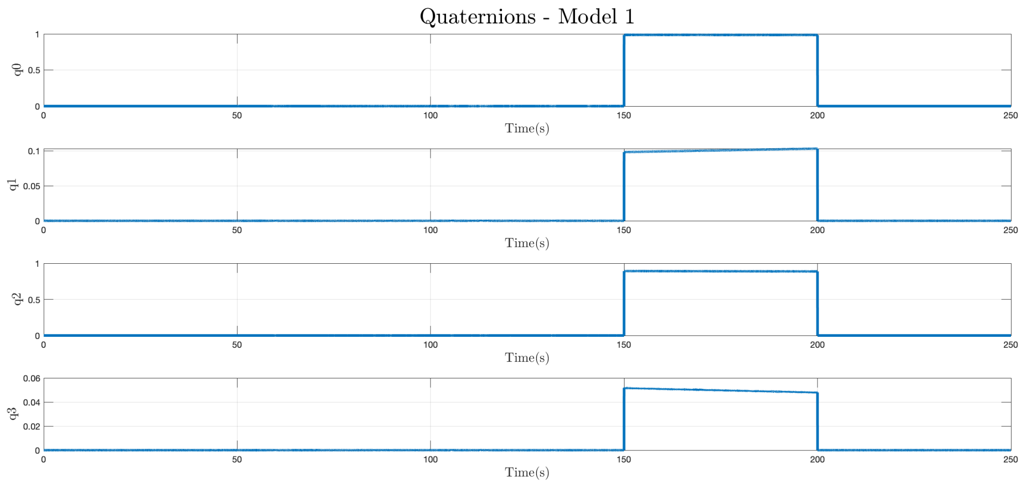

Figure 5.7 Residuals in Quaternions obtained when passing new data through Model 1.

deducted. Taking Figure 5.11 as an example, residuals from the nominal behavior have been removed, leading to an observed difference of zero units in the residuals during the initial 50 seconds. This approach accentuates the differences observed in each case.

The notably elevated residuals in Model 4, when juxtaposed with other models, can be attributed to the intense nature of Failure 3 and its marked impact on spacecraft dynamics.

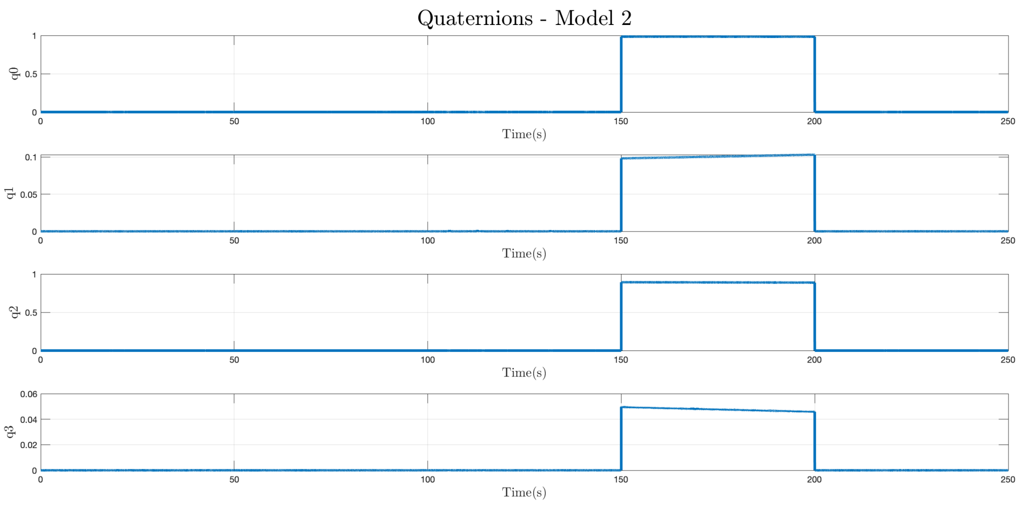

Figure 5.8 Residuals in Quaternions obtained when passing new data through Model 2.

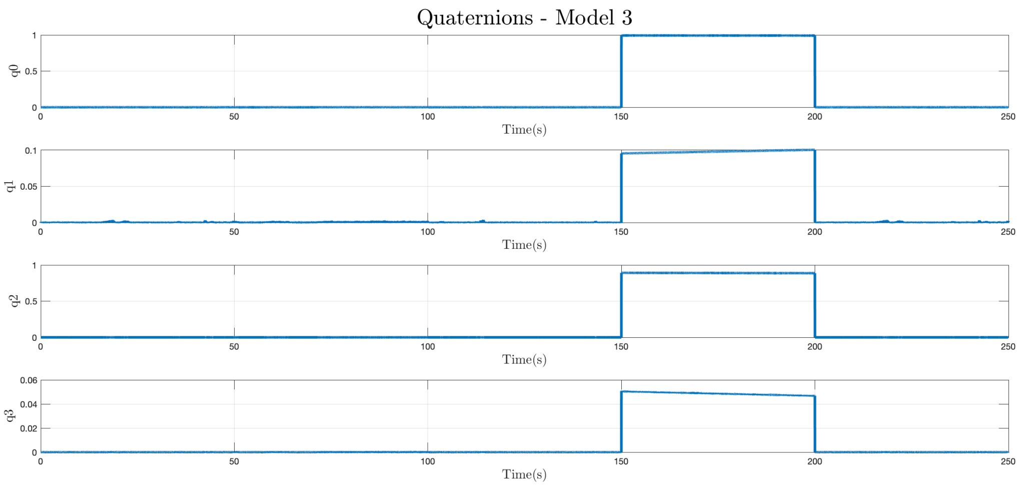

Figure 5.9 Residuals in Quaternions obtained when passing new data through Model 3.

These residual patterns underscore the challenges introduced by this pronounced failure, emphasizing the paramount importance of its precise detection and intervention to guarantee spacecraft stability and successful mission execution.

5.3.2 Probabilities with the A-MMAE Architecture

As discussed in Chapter 4, this proposed architecture introduces a modified approach to calculate probabilities within the framework of the Multiple Model Adaptive Estimation (MMAE) technique. Departing from the traditional model-based MMAE, this novel

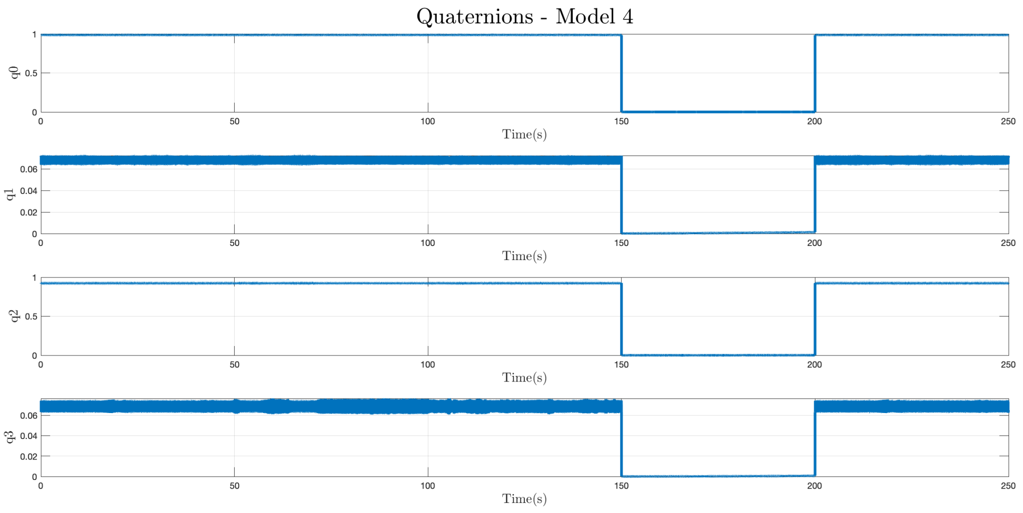

Figure 5.10 Residuals in Quaternions obtained when passing new data through Model 4.

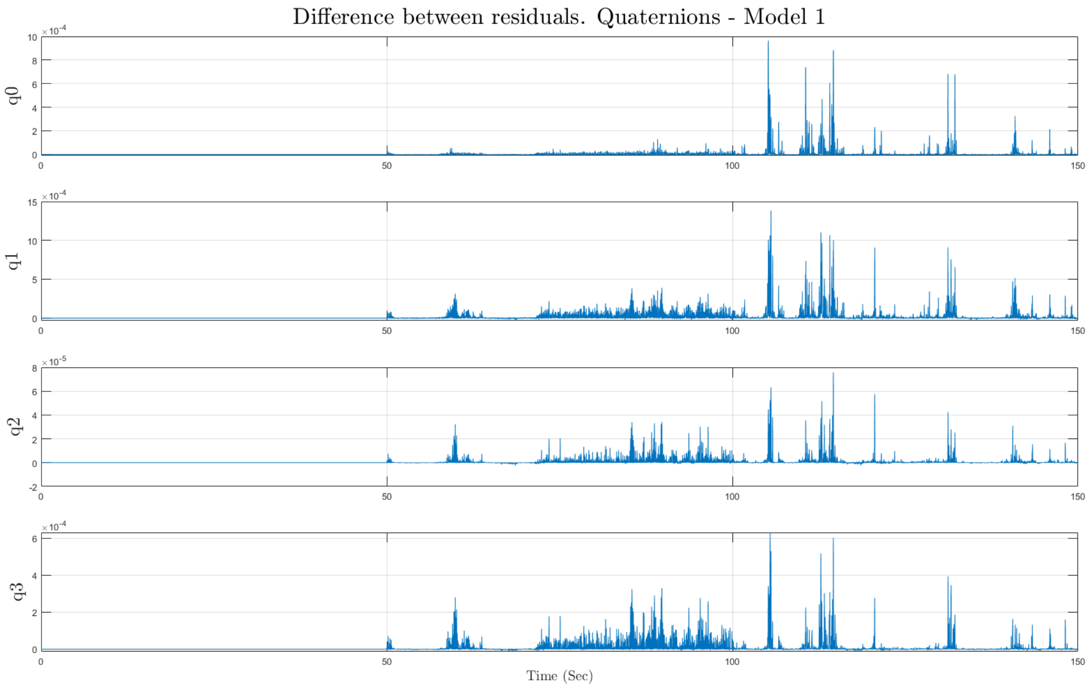

Figure 5.11 Difference between residuals obtained in Model 1.

approach emphasizes a data-driven methodology. In this revised architecture, the original estimator parameters, previously associated with the Kalman filter, have been replaced with selected parameters, including the Covariance matrix. The utilization of trained autoencoders for data reconstruction facilitates the simplified calculation of residuals, leading to a

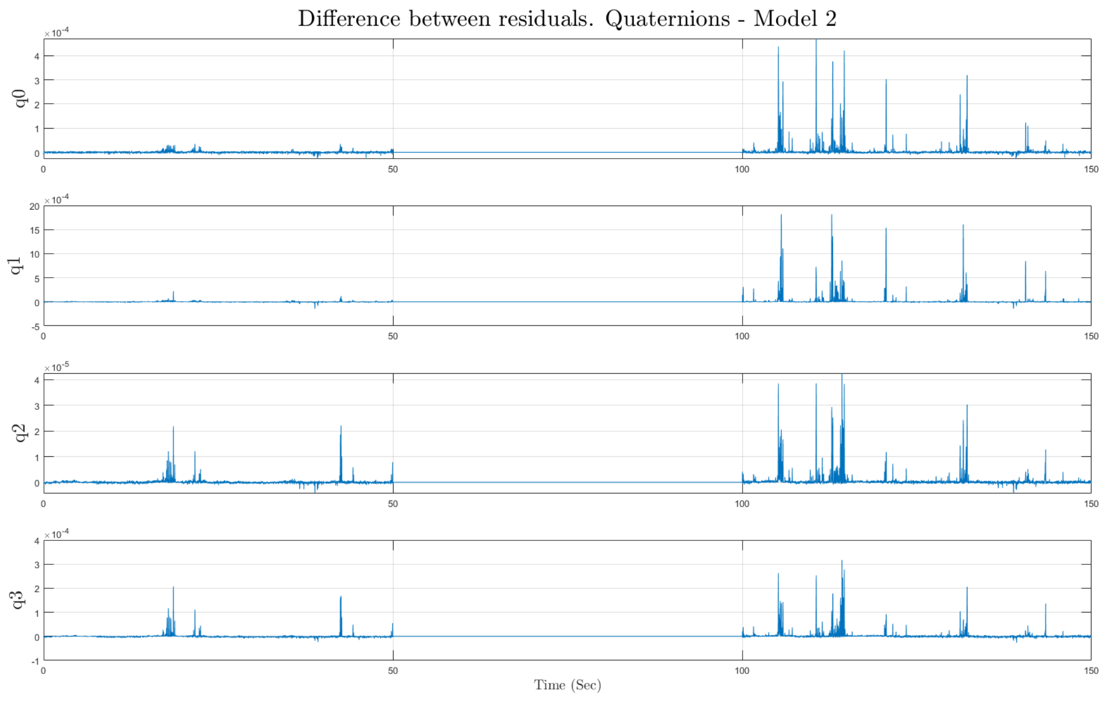

Figure 5.12 Difference between residuals obtained in Model 2.

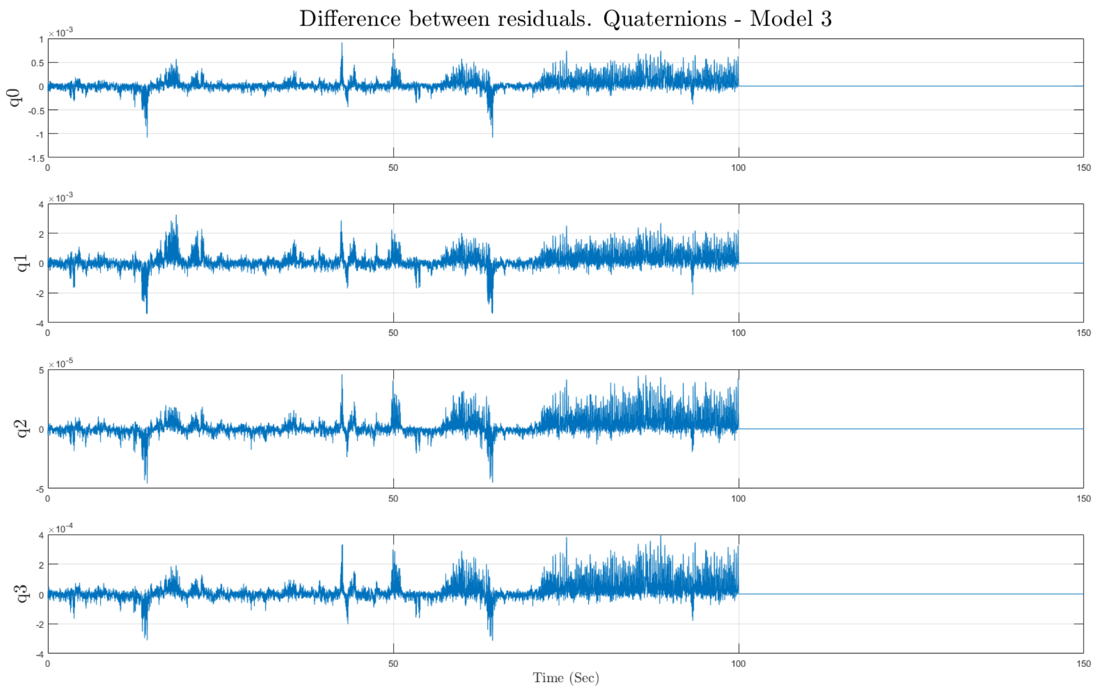

Figure 5.13 Difference between residuals obtained in Model 3.

streamlined probability calculation process without the need for additional computations.

By transitioning to a data-driven approach, the proposed architecture leverages the power of autoencoders in capturing essential patterns and features within the spacecraft's dynamics. The trained autoencoders serve as the foundation for obtaining residuals, enabling a direct assessment of data reconstruction quality. Consequently, the calculation of probabilities becomes more straightforward and eliminates the necessity for complex calculations that were required in the traditional model-based MMAE approach.

This modification in the calculation of probabilities enhances the efficiency and efficacy of the proposed architecture. By embracing a data-driven perspective, it leverages the advantages of autoencoders to simplify the estimation process while maintaining a high level of accuracy and reliability. The utilization of trained autoencoders allows for a more intuitive and streamlined approach to probability calculation, showcasing the adaptability and versatility of this novel architecture for fault detection and estimation in aerospace systems.

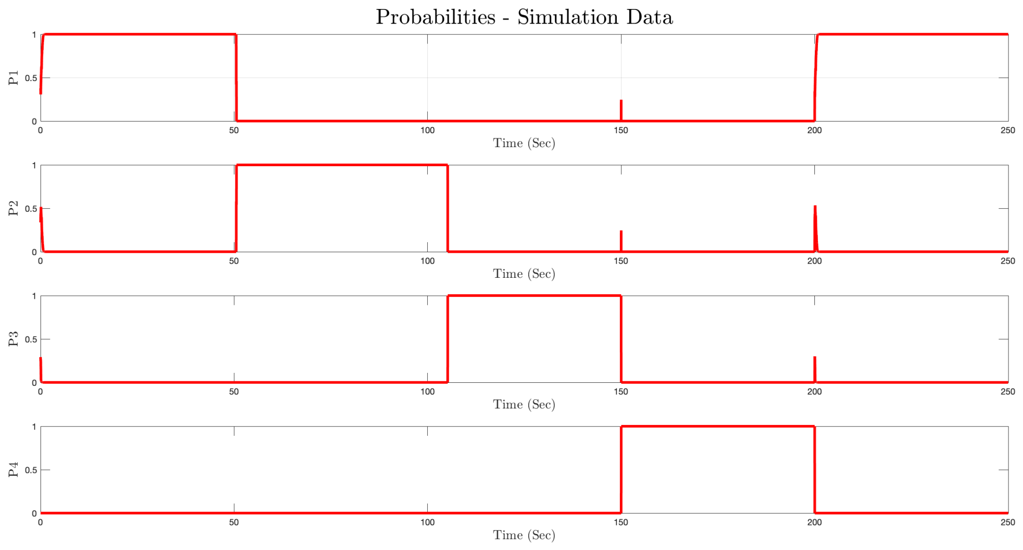

Figure 5.14 Probabilities obtained with the proposed architecture. Simulation Data.

Figure 5.14 presents the probability analysis indicating the likelihood of different scenarios occurring within the proposed architecture. Specifically, the first subplot, denoted as P1, represents the probability of Model 1, corresponding to the nominal condition, being the active model. Observing the plot, it becomes evident that from t=0s to t=50s and again from t=200s to t=250s, the probability value for Model 1 remains close to 1, indicating a high likelihood of the spacecraft operating in its nominal state during these time intervals.

However, for the subsequent 55-second interval, the probability shifts towards Model 2, which corresponds to Failure 1. The second subplot shows this transition, where Failure 1 becomes more likely to occur during this time segment. Similarly, in the following 45-second period, the probability shifts towards Model 3, associated with Failure 2, indicating an increased likelihood of encountering Failure 2 during that timeframe. The observed five-second deviation can be attributed to the occurrence of both faults on the same reaction wheel (RW#1). As a result, considering that the commanded orbit remains consistent, the states estimated by the autoencoder exhibit a certain degree of similarity. This similarity contributes to a lag of approximately 10% in the estimation process, resulting in the observed deviation.

Finally, in the last subplot, the probability analysis suggests that Failure 3 becomes the most probable scenario from t=150s to t=200s. This observation aligns with the behavior depicted in Figure 5.5, providing validation and consistency between the calculated probabilities and the actual occurrence of failures within the spacecraft system.

The probability analysis showcased in Figure 5.14 provides valuable insights into the occurrence of different failure scenarios over time. This information is instrumental in identifying and addressing potential issues, enabling proactive measures to mitigate the impact of failures and ensure the overall robustness and reliability of the spacecraft system.

6 Experimental Verification

To complement the numerical simulations, an experimental verification of the proposed architecture was conducted using real flight data. This experimental validation provided valuable insights into the performance and practical applicability of the Autoencoder-based Multiple-Model Adaptive Estimation (A-MMAE) framework in real-world scenarios. By analyzing actual flight data collected from aerospace systems, we can assess the architecture's ability to detect and isolate faults under realistic operating conditions.

The experimental verification aims to bridge the gap between theoretical simulations and practical implementation. By utilizing real flight data, we can evaluate the architecture's performance in real-world environments, where factors such as sensor noise, uncertainties, and operational constraints may influence the system's behavior. This validation process will enable us to assess the architecture's robustness, accuracy, and reliability, ultimately ensuring its effectiveness in enhancing fault detection and isolation capabilities in aerospace systems.

In this chapter, a comprehensive overview of the experimental setup, data acquisition process, and analysis techniques employed for the verification are provided. By rigorously analyzing the experimental results and comparing them with the expected outcomes, the proposed architecture can be validated and valuable insights can be gained into its performance under real operational conditions. The findings from this experimental verification will further enhance the understanding and practical implementation of the A-MMAE framework, contributing to advancements in fault detection and isolation techniques for aerospace systems.

6.1 Spacecraft Testbed Experimental Setup

A high fidelity spacecraft tested, the Extreme Access System (EASY) Spacecraft, was used to test the auto-encoder based MMAE (A-MME) algorithm for operational system verification and validation. The EASY Spacecraft Testbed is a concept spacecraft designed at the Advanced Dynamics and Control Laboratory (ADCL) of Embry-Riddle Aeronautical University, FL, with the primary purpose of supporting the development of innovative autonomous space exploration spacecraft for *in situ* resource utilization in environments such as asteroids, where gravitational force is minimal. Figure 6.1 provides a picture of this testbed configuration. The EASY Spacecraft is mounted on a three-degree-of-freedom gimbaled platform, enabling free motion in roll, pitch, and yaw axes. This setup simulates full attitude control and angular rate regulation in microgravity environments, facilitating the testing of trajectory tracking and recovery from tumbles or other abnormal conditions that may arise in space.

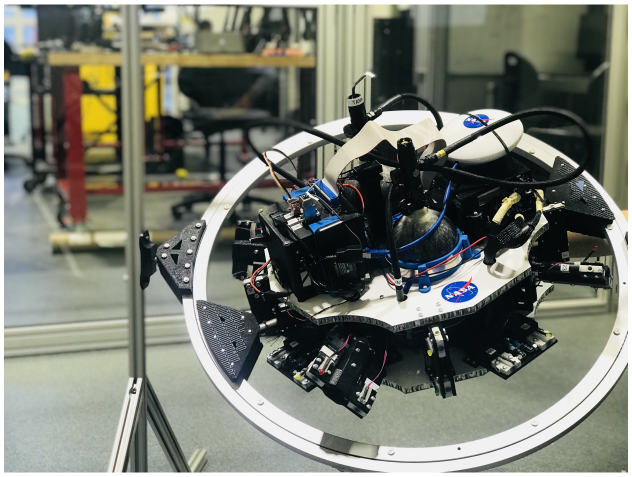

Figure 6.1 Extreme Access System (EASY) Spacecraft Test Bed.

The EASY Spacecraft features 24 solenoid-valves acting as thrusters, with sixteen fixed thrusters arranged in pairs and eight Thrust Vectoring Control (TVC) thrusters placed in pairs around the z-axis. The fixed thrusters are organized in horizontal and vertical configurations, enabling yaw and pitch rotations, respectively. The TVC thrusters are connected to a servo motor mechanism, allowing each pair to rotate around its position for additional yaw rotation control.Figure 6.2 shows the location of each thruster around the x and y axes of EASY.

The propulsion system of the spacecraft relies on compressed air stored in two high-pressure reservoirs, which can hold up to 4500 psi of pressure. Four pressure regulators reduce the pressure from the reservoirs to the desired operating pressure of 130 psi for

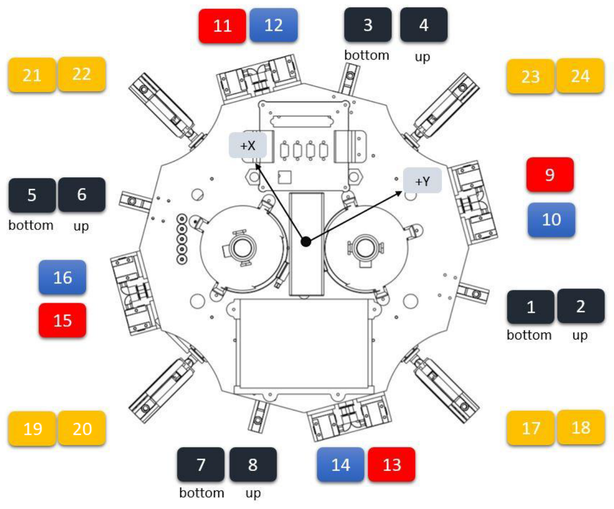

Figure 6.2 Location of the thrusters in the x and y axes. Yellow for TVC; red and blue for horizontal configuration, left and right side respectively; dark blue for vertical configuration, denoting both bottom or up.

the thrusters. Pressure sensors monitor the system's pressure levels, and relief valves are employed for safety in case of over-pressure. The opening and closing of the solenoid valves are regulated by Pulse Width Modulated (PWM) signals from the digital IO pins on the onboard computer. Figure 6.3 provides a schematic of this cold gas thruster configuration and propellant feed lines.

The flight computer used for the EASY Spacecraft is the PC/104 PCM-3355, which is integrated with the Emerald MM-4M-Port serial module and the Onyx MM Digital I/O module. This integration allows the spacecraft to include serial and analog input modules, along with digital I/O, which are used to actuate each of the solenoid valves to regulate the appropriate amount of air required for attitude control. An Inertial Measurement Unit (IMU) from Microstrain is employed to provide accurate measurements of attitude and angular rates, which are necessary for the controllers. The Microstrain IMU communicates with the flight computer through an RS232 communication protocol. A robust Nonlinear Dynamic Inversion (NLDI) controller provides the command control for attitude tracking.

The hardware components were tested individually to ensure full functionality before being mounted and integrated into the EASY Spacecraft. The testbed is connected to the

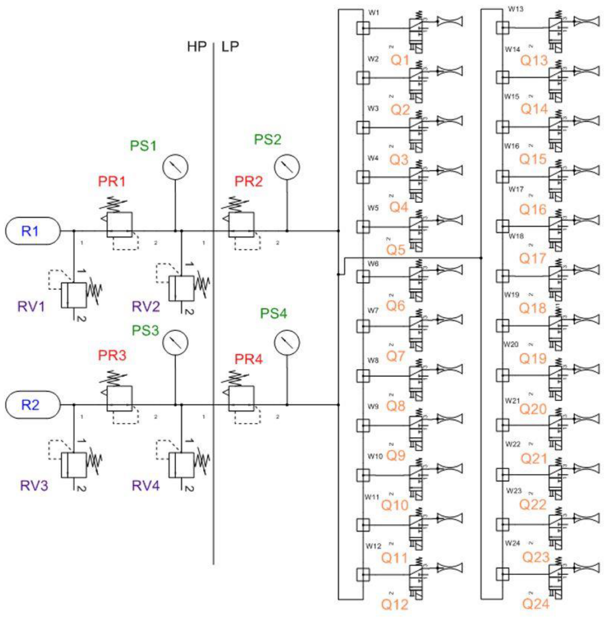

Figure 6.3 Propulsion System.

host computer via a high data rate Wi-Fi connection, which is crucial for online tuning of the controllers and signal monitoring. Figure 6.4 illustrates the EASY Spacecraft and lab control workstation setup.

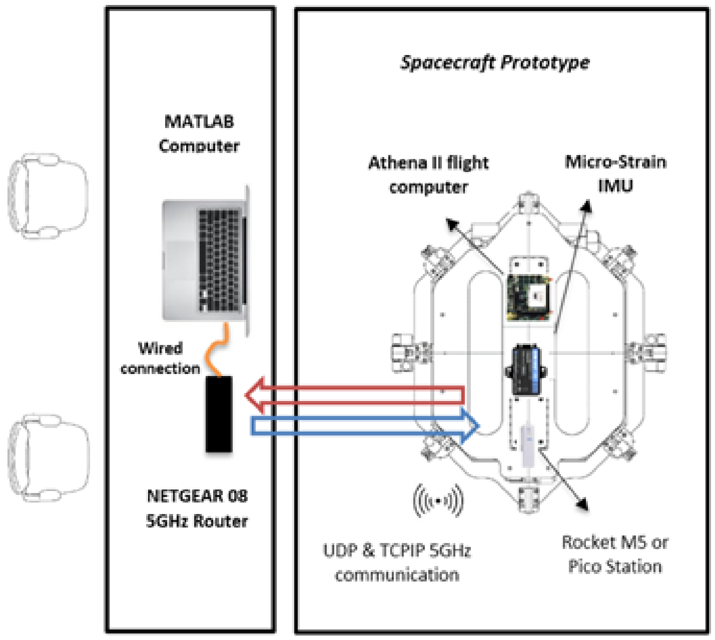

Figure 6.4 Schematic of Test Bed and Hardware used on EASY.

6.2 Spacecraft Testbed Data Acquisition

The application process of the A-MMAE fault detection system began with data accumulation for offline training. This involved conducting several nominal test cases with the testbed, enabling the creation of a training vector. This series of nominal test cases required the spacecraft to execute various roll maneuvers ranging between 28 and 35 degrees. It is important to note that the pitch and yaw axes of the spacecraft gimbal were locked in order to improve repeatability of the tests. The purpose of the tests was to generate a diverse set of data, representing a range of normal operating conditions for the spacecraft to perform this maneuver. This information was vital in the creation of a robust training vector for the A-MMAE, which would be essential for accurate fault detection during real-world operations.

Throughout the nominal test cases, several key data streams were recorded to provide a comprehensive understanding of the spacecraft's behavior during these maneuvers. These data streams created the selected features to represent the nominal dynamics of the testbed. Eighteen primary features used within this application are shown in Table 6.1.

Once the nominal data was gathered, the A-MMAE was trained on the selected feature spaces to ensure the model's adaptability to new and unseen scenarios. This training process created X possible model representations and enabled the A-MMAE to develop a comprehensive bank of models of the spacecraft's normal and abnormal behavior and, subsequently, enhance its ability to detect any deviations from this norm, indicating potential faults.

To test the A-MMAE's fault detection capability, a failure test case was conducted to simulate a fault within the spacecraft's attitude control thruster system. In this scenario, Thruster 1 was intentionally turned off, preventing the EASY spacecraft from performing a commanded roll maneuver. As an example, the roll rate time history response is provided in Figures 6.5 and 6.6 for a nominal and failure test case for comparison. The nominal data was acquired from a 32 degree roll maneuver test case scenario and the failure data was acquired from a 35 degree commanded roll maneuver with the injected thruster error in T_1.

Table 6.1 Selected features and states for the EASY Spacecraft.

	Feature	Desription
1	p	Roll Rate, deg/s
2	q	Pitch Rate, deg/s
3	r	Yaw Rate, deg/s
4	ϕ	Roll angle, deg
5	θ	Pitch angle, deg
6	ψ	Yaw angle, deg
7	$q_{0_{actual}}$	Quaternion actual q0
8	$q_{1_{actual}}$	Quaternion actual q1
9	$q_{2_{actual}}$	Quaternion actual q2
10	$q_{3_{actual}}$	Quaternion actual q3
11	T_1	Thruster 1 Actuation
12	T_2	Thruster 2 Actuation
13	T_3	Thruster 3 Actuation
14	T_4	Thruster 4 Actuation
15	T_5	Thruster 5 Actuation
16	T_6	Thruster 6 Actuation
17	T_7	Thruster 7 Actuation
18	T_8	Thruster 8 Actuation
19	T_9	Thruster 9 Actuation
20	T_{10}	Thruster 10 Actuation
21	T_{11}	Thruster 11 Actuation
22	T_{12}	Thruster 12 Actuation
23	T_{13}	Thruster 13 Actuation
24	T_{14}	Thruster 14 Actuation
25	T_{15}	Thruster 15 Actuation
26	T_{16}	Thruster 16 Actuation

6.3 Failure Scenarios

In this case, the focus of the analysis shifts to failures related to the power plant and sensor measurements within the spacecraft system. This change in the analyzed failures presents an opportunity to validate the effectiveness and versatility of the built architecture, showcasing its ability to handle diverse failure scenarios. By expanding the scope to include power plant-related failures and sensor measurement anomalies, a comprehensive evaluation can be conducted to assess the system's robustness and reliability across various fault scenarios.

- **Failure Case I** involves a power plant failure where Thruster 2 loses power at t=0s,

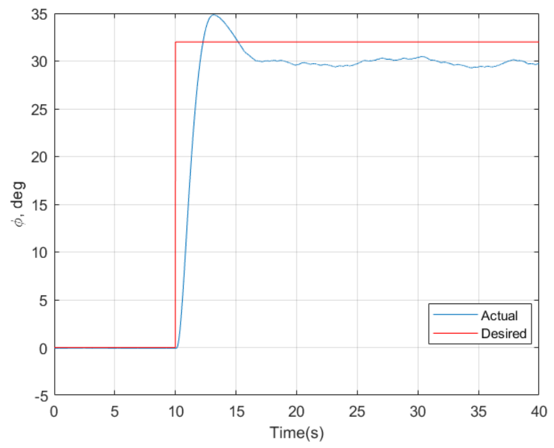

Figure 6.5 Nominal EASY spacecraft roll angle test data.

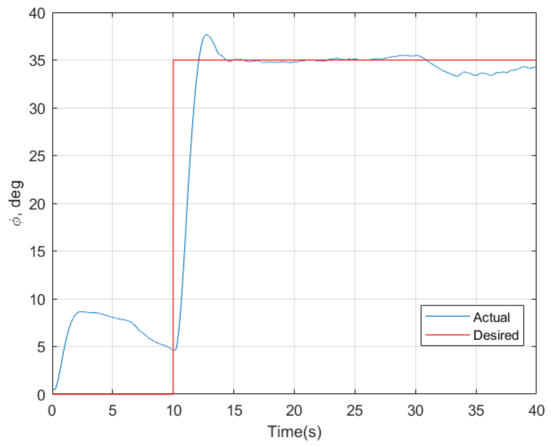

Figure 6.6 Nominal EASY spacecraft roll angle injected failure test data.

resulting in its complete cessation of operation. A failure in the power plant of a space system, specifically a thruster failure, can have significant implications for the spacecraft's maneuvering and control capabilities. When a single thruster fails, the spacecraft's ability to generate thrust and adjust its attitude is compromised. This can lead to difficulties in maintaining desired orientations, performing precise maneuvers, and overall mission success.

- **Failure Case II**, also occurring at time t=0s, is also a power plant failure, with both Thrusters 1 and 7 experiencing power loss and subsequently becoming inoperative.

When multiple thrusters fail, the impact on spacecraft control becomes even more severe. With the loss of multiple thrusters, the spacecraft may struggle to counteract external disturbances, maintain stability, or execute complex maneuvers. The failure of two thrusters can significantly limit the spacecraft's maneuverability and may necessitate alternative control strategies to mitigate the loss of thrust authority.

- **Failure Case III**, again at time t=0s, pertains to a sensor failure, specifically a bias in the roll angular measurement, where the measurement exhibits a constant bias of 15 degrees per second. These sensor failures can lead to erroneous attitude information, potentially resulting in compromised stability, control, and navigation of the spacecraft.

To gain further insights into these types of failures in power plants and sensor measurements in space systems, researchers have conducted extensive studies and investigations. Understanding the ramifications and challenges posed by said failures is crucial for developing robust fault detection and isolation algorithms, designing fault-tolerant control strategies, and ensuring the reliability and performance of space systems in demanding operational environments.

6.4 Results

In accordance with the previous section, each failure scenario is associated with a specific model. Model 1 corresponds to the Nominal data, Model 2 utilizes data from Failure Case I, Model 3 incorporates data from Failure Case II, and Model 4 is trained with data from Failure Case III. Each model is again accompanied by a corresponding autoencoder, tailor-made to reconstruct the data associated with its respective scenario.

6.4.1 Residual Calculation and Analysis

The maneuvers performed in this case involve various roll maneuvers ranging between 28 and 35 degrees, as illustrated in Figure 6.7. Due to the nature of the experimental setup, the acquired data differs from the previous section. The spacecraft undergoes approximately

60 seconds of flight. Data was split into training and testing, resulting in a set of real flight data of approximately 87 seconds being used to test the system.

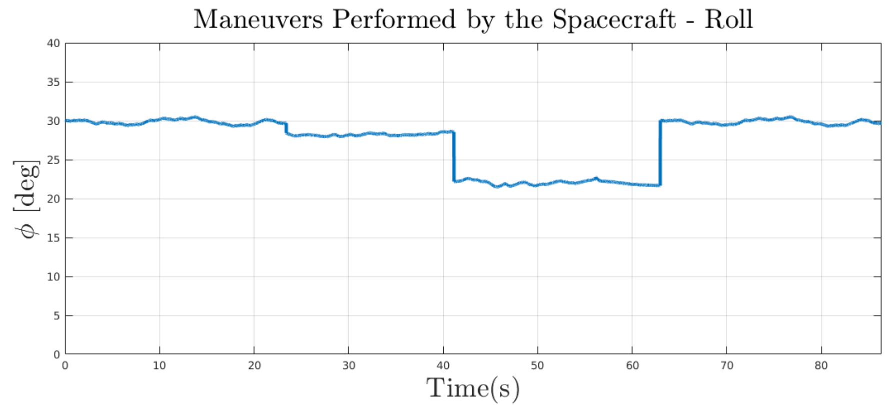

Figure 6.7 Maneuvers performed by the Spacecraft around 30 degrees in roll.

Following a similar arrangement as before, Figure 6.8 shows the testing data set organized in the order of nominal, failure 1, failure 2, failure 3, and nominal data segments. By adhering to this disposition, it is anticipated that similar results will be obtained, enabling the evaluation and comparison of the system's performance across different scenarios.

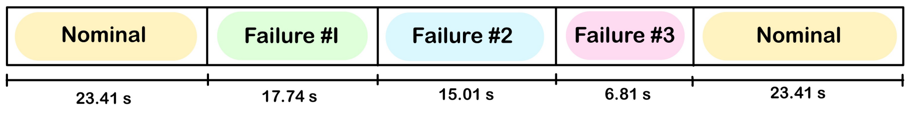

Figure 6.8 Disposition of the experimental data set used to detect failures.

After the auto-encoders were properly trained, residuals were then obtained by subtracting the original data from the reconstructed outputs of the auto-encoders. In terms of probabilities, if the residuals of a specific model are low, then the probability of the system behaving as that model shall be almost 1.

Figure 6.9 shows the time history of the residuals when the validation data (not used in the training process) corresponding to the nominal condition was passed through all auto-encoders. Since Model 1 represents the nominal condition, the lowest residuals clearly correspond to the first and last sections of the data, accordingly with Figure 6.8.

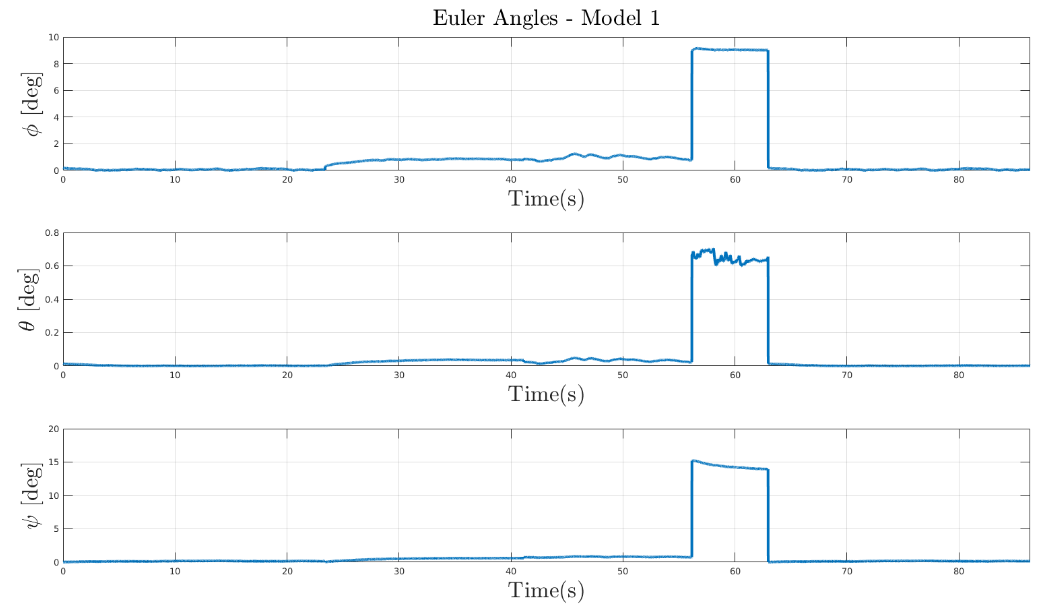

Figure 6.9 Residuals in Euler Angles obtained when passing new data through Model 1.

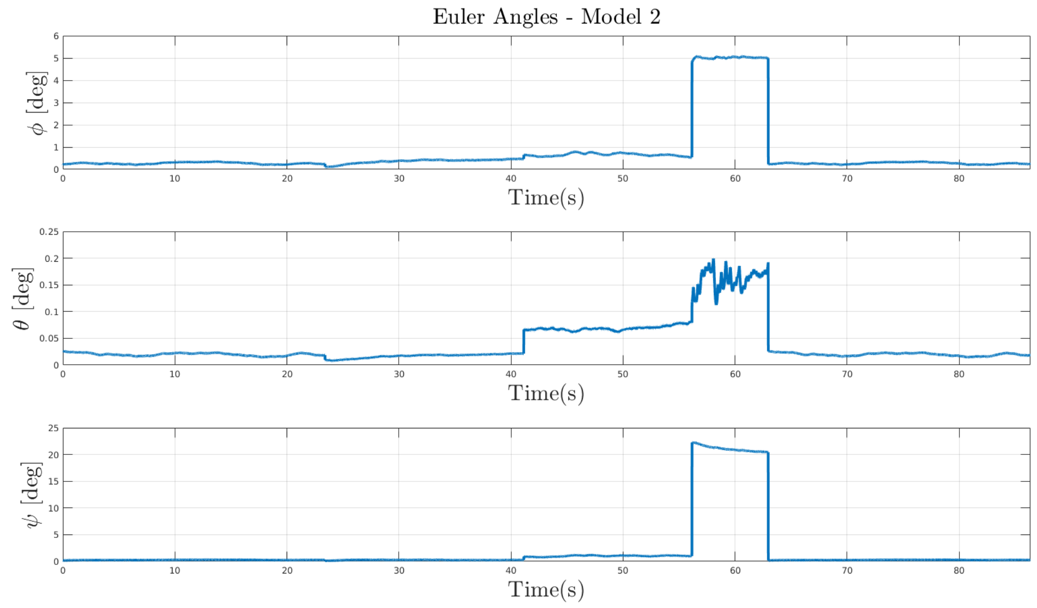

Figure 6.10 Residuals in Euler Angles obtained when passing new data through Model 2.

Figure 6.10 shows the time history of the residuals when failure 1 was passed through all auto-encoders. Since Model 2 was trained to detect Failure 1 condition, it is expected that residuals would be minimal in the second part of the data. It can be observed that

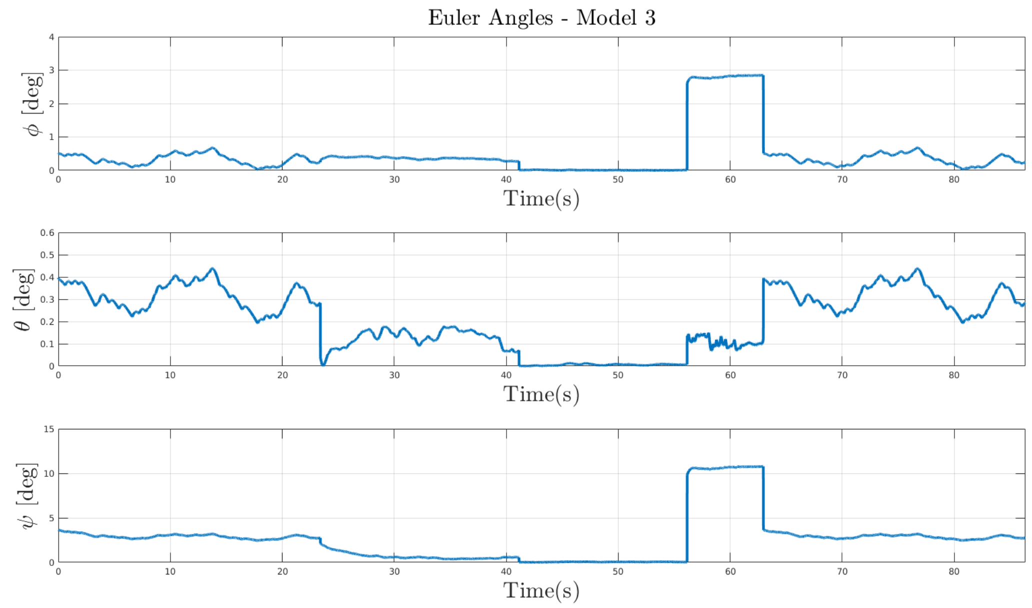

Figure 6.11 Residuals in Euler Angles obtained when passing new data through Model 3.

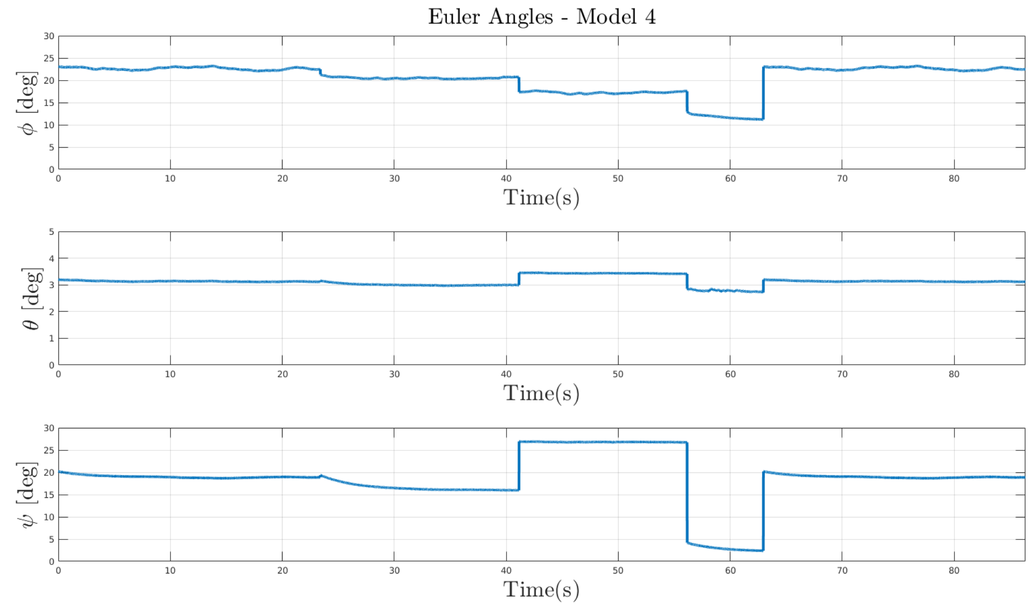

Figure 6.12 Residuals in Euler Angles obtained when passing new data through Model 4.

residuals are slightly lower after 23.41 seconds of flight, then higher at time 41.15s. In this particular case, the failure's effects are well-handled by the spacecraft's controller, resulting in a behavior almost nominal.

Figure 6.11 shows the time history of the residuals when the failure 2 data was passed through all auto-encoders. It is clear to note how the residuals decrease to almost zero between around 42sec and 56sec, which corresponds to the window of failure 2 behavior according to Figure 6.8.

Finally, Figure 6.12 shows the time history of the residuals when the failure 3 data was passed through all auto-encoders. It is clear to note how the residuals decrease with respect to other models between around 56sec and 62sec, which corresponds to the window of failure 3 behavior according to Figure 6.8.

6.4.2 Probabilities with the A-MMAE architecture

The results presented in this subsection were calculated using the same methodology as in Chapter 5. However, it is worth noting that slight differences were observed in the obtained results. These differences will be thoroughly discussed and analyzed in subsequent sections, allowing for a comprehensive examination of the variations and their implications. By investigating these discrepancies, a deeper understanding of the architecture's structure and the impact of different factors on the results can be gained. This subsequent discussion will provide valuable insights into the nuances and intricacies of the proposed architecture, enabling further refinement and improvement of its performance.

Figure 6.13 illustrates the probability analysis, offering insights into the occurrence of different scenarios within the proposed architecture. The first subplot, labeled as P1, represents the probability of Model 1, corresponding to the nominal condition, being active. From t=0s to approximately t=29s and from t=63s to t=87s, the probability value remains consistently close to 1, indicating a high likelihood of the spacecraft operating in its nominal state during these intervals.

During the subsequent 14-second period, the probability shifts towards Model 2, associated with Failure 1. This transition suggests an increased probability of encountering Failure 1 within this timeframe. Similarly, in the following 13 seconds, the probability shifts towards Model 3, corresponding to Failure 2, indicating a higher likelihood of experiencing Failure

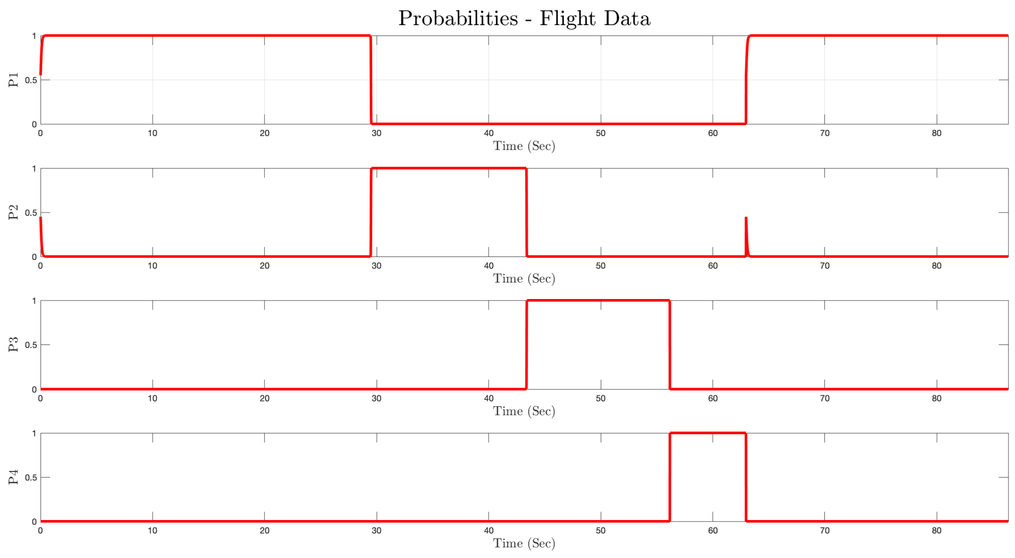

Figure 6.13 Probabilities obtained with the proposed architecture. Flight Test Data.

2 during that interval. Towards the end of the observation period, the probability analysis suggests that Failure 3 becomes the most probable scenario in about 7 seconds.

The observations made do not fully align with the patterns depicted in Figure 6.8. These differences can be influenced by various factors. One such factor could be an incorrect covariance matrix, which might require adjustments to ensure precise estimation and prediction within the architecture. Another disparity arises from the minimal variations in residuals observed for both Model 1 and Model 2. As previously mentioned, Failure 1 is categorized as a minor fault that the spacecraft's controllers adeptly mitigate, resulting in behavior that closely resembles the nominal condition. The controllers' efficient management of Failure 1 minimizes the impact on residuals, thereby leading to insignificant changes in the estimation process.

In the case of Model 3, which is trained using data from Failure 2, a noticeable 2-second lag is observed. This lag can potentially be attributed to the controller's robustness. The effective functioning of the controller in mitigating the effects of Failure 2 introduces a slight delay in the estimation process, causing the observed lag in the results.

The insights gained from the probability analysis presented in Figure 6.13 offer valuable information for identifying and addressing potential failures. By monitoring the evolving probabilities, proactive measures can be taken to mitigate the impact of failures and enhance the overall reliability and robustness of the spacecraft system.

6.5 Study of Unknown Failures

In this section, an exploration of ”unknown failures” (UF), which constitute either power plant or sensor failures, is undertaken. These failures share similarities with those that the autoencoders were originally trained on. The UF were subject to testing, albeit without the benefit of prior training data, with the objective of determining the A-MMAE model's ability to not only detect the occurrence of any failure but also accurately identify them. This study seeks to expand the understanding of the model's robustness and its capacity for detecting and classifying failures beyond its initial training scope.

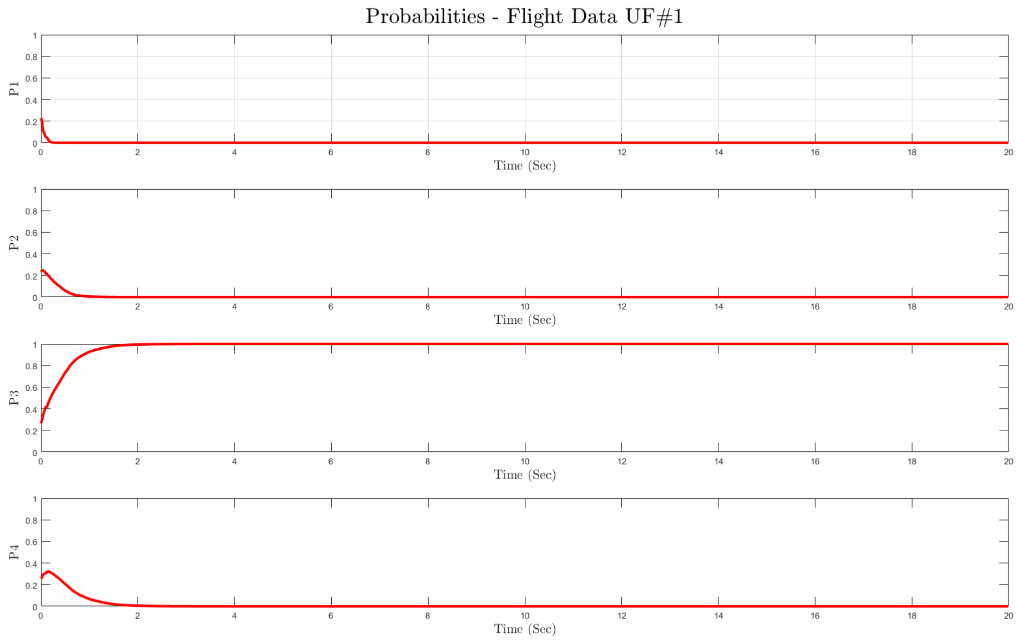

Figure 6.14 Probabilities of the system to behave as the trained models. Unknown Failure I.

In the evaluation of **Unknown Failure I**, a power plant failure is instigated with Thruster 1 losing power at t=0s, leading to its immediate halt. Similarities should be

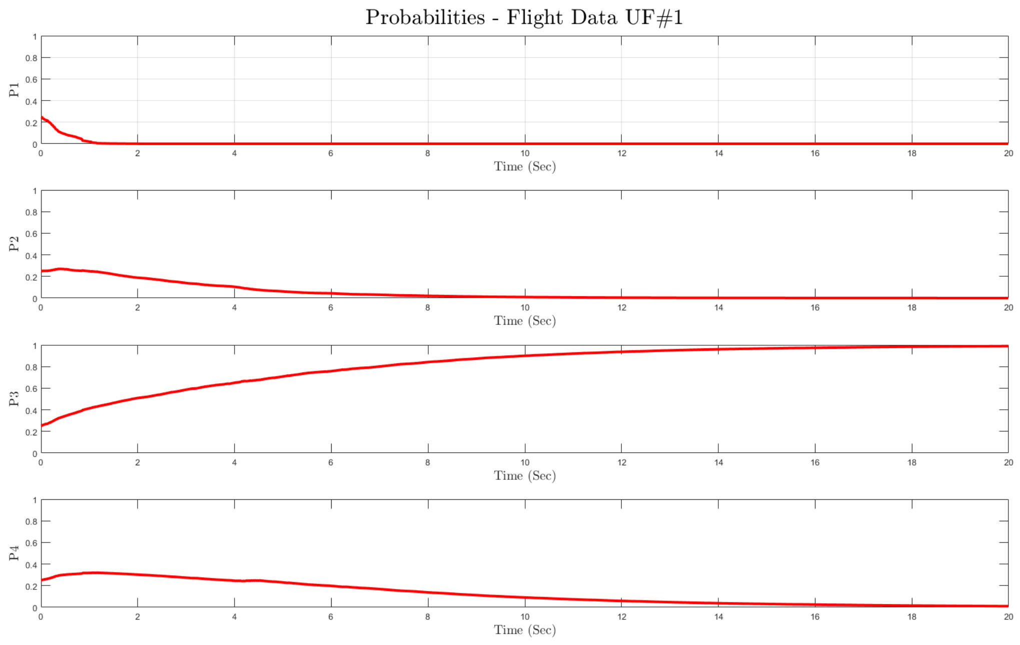

Figure 6.15 Probabilities of the system to behave as the trained models. Unknown Failure I. Different Covariance Matrix used to compute the probabilities.

detected with Failure case II, Model 3 due to the identical thruster failure. The A-MMAE produced varying outputs linked to the chosen covariance for probability computation, as depicted in Figure 6.14 and Figure 6.15. It is noted that this parameter was customized for optimized performance in each case. Even with the slight difference in initial flight seconds, the model successfully detected that UF1 bears similarities to Model 3.

Next, **Unknown Failure II** constitutes a power plant failure, where two thrusters, Thrusters 2 and 7, lose power at t=0s. This failure is expected to resemble Model 2 and Model 3, given that these models were trained on these specific thruster failures. However, none of the probabilities should be exact, as UF2 represents a combination of both models rather than a perfect match to either. Figure 6.16 confirms this prediction, indicating higher than zero probabilities for the system behaving as Model 2 and Model 3. Notably, the probability of behaving as Model 3 is higher due to the concurrent failure of two thrusters in both UF2 and Failure Case 2.

Unknown Failure III incurs a power plant failure with Thruster 5 losing power at t=0s.

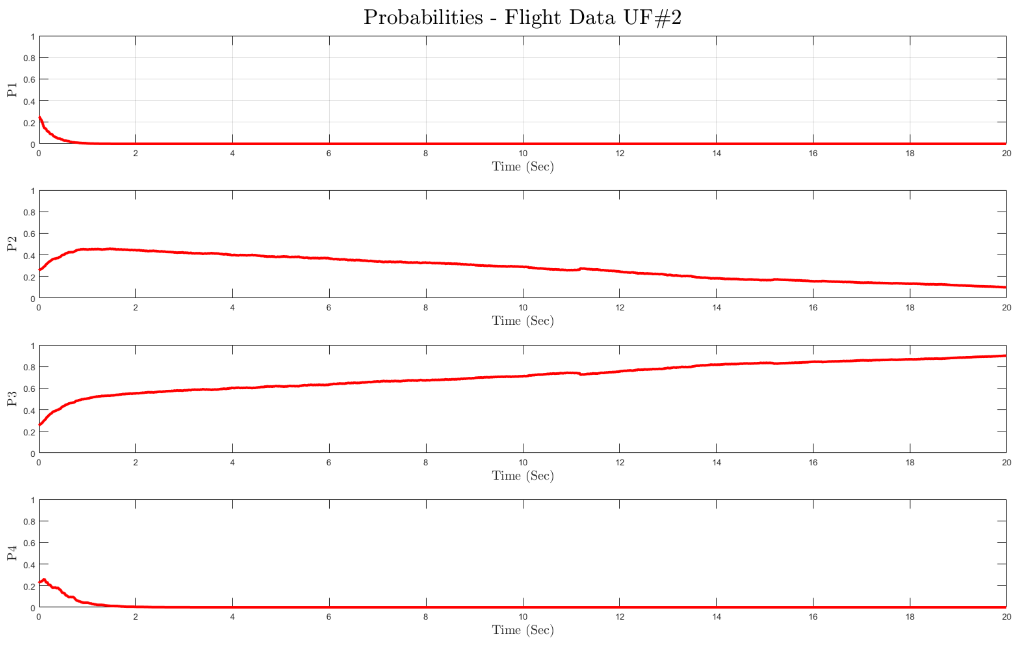

Figure 6.16 Probabilities of the system to behave as the trained models. Unknown Failure II.

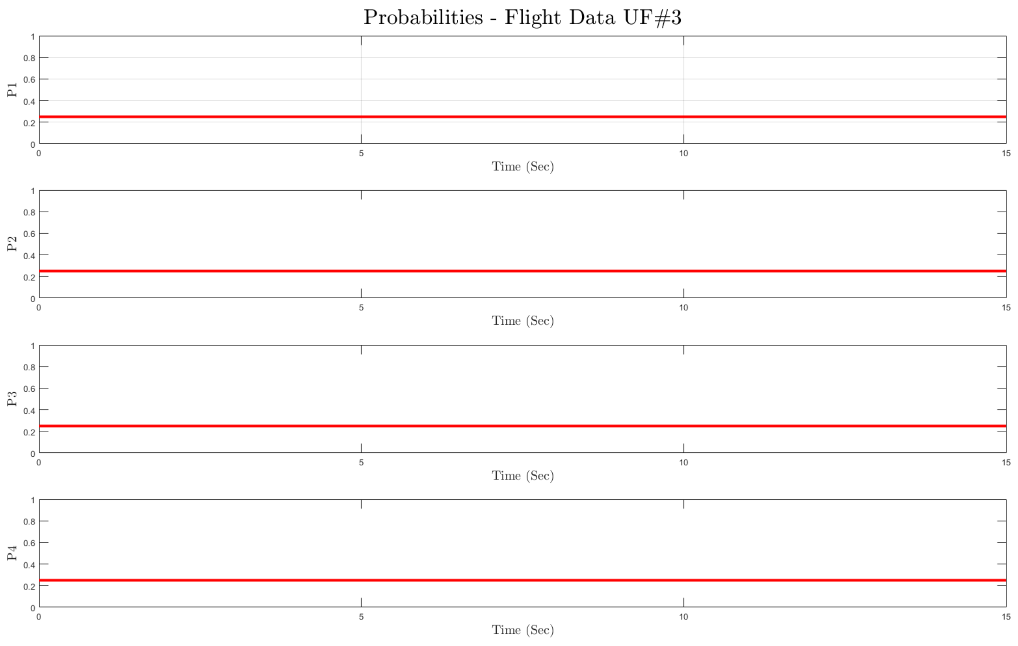

Figure 6.17 Probabilities of the system to behave as the trained models. Unknown Failure III.

This failure occurs in a thruster that was not considered in the A-MMAE's prior training, potentially leading to model dysfunction or detection failure. Contrary to expectations, Figure 6.17 illustrates that the A-MMAE model can identify abnormal system behavior, indicating failure. However, as predicted, no high probabilities are associated with any trained models given the distinctiveness of UF3.

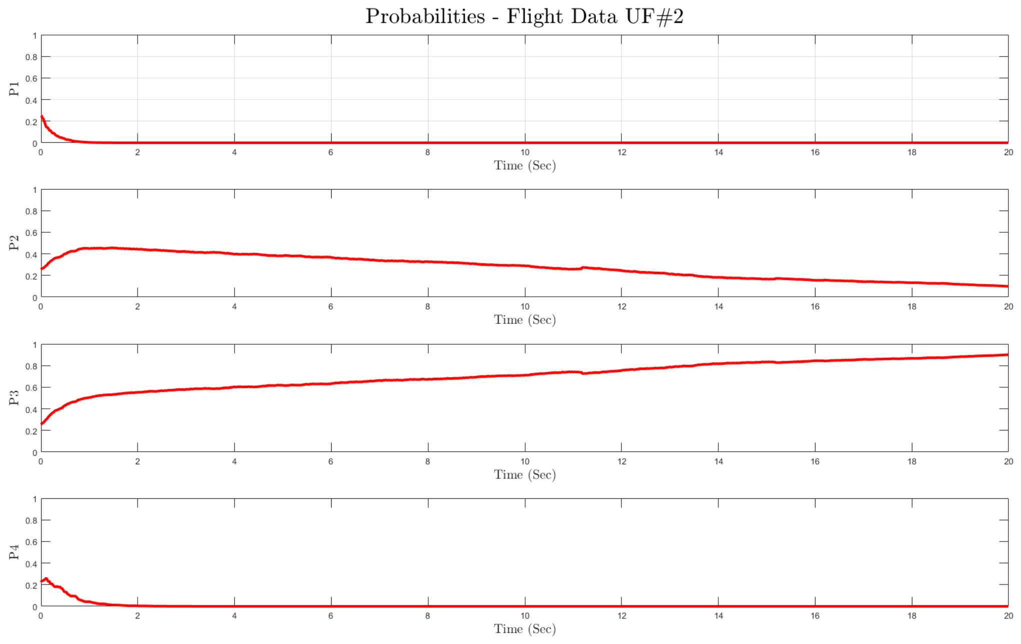

Figure 6.18 Probabilities of the system to behave as the trained models. Unknown Failure IV.

Unknown Failure IV signifies a power plant failure where two thrusters, 1 and 8, lose power at t=0s. This failure shares similarities with Failure Case II, which had Thrusters 1 and 7 malfunction. Hence, the probability of the system behaving as Model 3 is anticipated to be high. Figure 6.18 affirms this hypothesis, suggesting the system is experiencing a failure analogous to Model 3. This could infer a failure in either of thrusters 1 and 7 or both. However, the A-MMAE model's identification of such failures may not be as accurate as expected, given the potential for multiple concurrent failures.

Lastly, **Unknown Failure V** entails a sensor failure, specifically a bias in the roll angular measurement, introducing a consistent bias of 7 degrees per second. This failure mirrors the

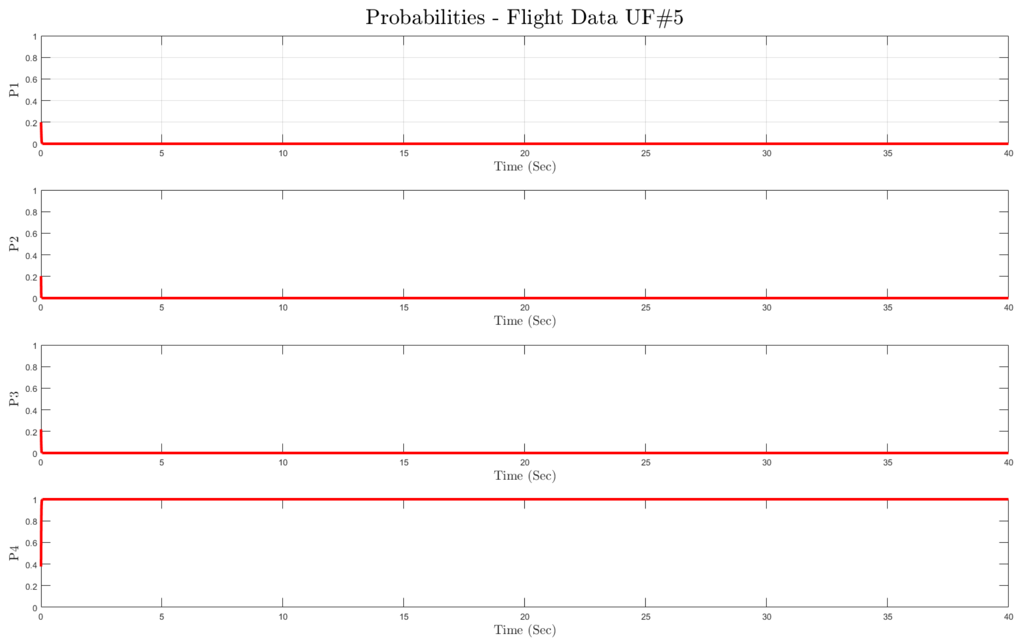

Figure 6.19 Probabilities of the system to behave as the trained models. Unknown Failure V.

behavior delineated by Model 4, hence the probability should be near 1, which is confirmed by Figure 6.19.

The results drawn from these experiments provide insights into the system's capabilities to detect and identify fault behavior that aligns with the failures previously trained with the autoencoders.

7 Conclusions and Future Work

The conclusions drawn from this thesis substantiate the efficacy of the proposed architecture in detecting and identifying various types of spacecraft failures. The case of trained failures reveals that despite a few instances where the model's predictions diverged from the actual system behavior, these discrepancies can be chiefly attributed to the intrinsic characteristics of the failures themselves. These were effectively counteracted by the spacecraft's control mechanisms, demonstrating the robustness of the proposed model. Notably, the architecture demonstrated its proficiency by promptly detecting and accurately pinpointing aggressive and critical failures, an essential capability for maintaining the system's operational integrity.

An extension of the analysis to encompass unknown failures was conducted for the experimental case. This thorough evaluation offered crucial insights into the architecture's performance when confronted with a wide array of untrained failure scenarios. It is thus evident that the system exhibits commendable resilience and adaptability, capable of detecting and identifying a significant portion of failures that resemble those on which it has been trained. However, minor discrepancies and overlaps, while expected, could be further mitigated by augmenting the number of models utilized to represent system behavior, thereby enhancing the model's overall robustness.

In summary, the research presented in this thesis has culminated in the successful development and validation of a fully data-driven Fault Detection and Identification (FDI) method. A noteworthy feature of this accomplishment is that the architecture operates without specific knowledge of the mathematical models underlying the systems under examination; instead, it relies extensively on the dataset of system features for input. By harnessing the potential of autoencoders coupled with a parallel bank of estimators, the architecture has conclusively demonstrated its proficiency in detecting a diverse array of failures. This significant advancement promises to bolster system reliability and safety, contributing to the broader field of spacecraft engineering and operations.

REFERENCES

[1] Müller, S., "Towards a Conceptual Data Model for Fault Detection, Isolation and Recovery in Virtual Satellite," 2018.

[2] Lu, P., Van Eykeren, L., van Kampen, E., and Chu, Q. P., "Selective-Reinitialization Multiple-Model Adaptive Estimation for Fault Detection and Diagnosis," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 8, 2015, pp. 1409–1424. https://doi.org/10.2514/1.G000587.

[3] "https://medium.com/@srijaneogi31/exploring-multi-class-classification-with-deep-learning-239cb42e69bf," , 2020.

[4] Vochozka, M., Horák, J., and Šuleř, P., "Equalizing seasonal time series using artificial neural networks in predicting the Euro–Yuan exchange rate," *Journal of Risk and Financial Management*, Vol. 12, No. 2, 2019, p. 76.

[5] "https://www.assemblyai.com/blog/introduction-to-variational-autoencoders-using-keras/," , 2022.

[6] Kalman, R. E., "A new approach to linear filtering and prediction problems," 1960.

[7] Tudoroiu, N., and Khorasani, K., "Satellite fault diagnosis using a bank of interacting Kalman filters," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 43, No. 4, 2007, pp. 1334–1350.

[8] Chen, J., Patton, R. J., Chen, J., and Patton, R. J., "Robust residual generation using unknown input observers," *Robust model-based fault diagnosis for dynamic systems*, 1999, pp. 65–108.

[9] Alwi, H., Edwards, C., and Marcos, A., "FDI for a Mars orbiting satellite based on a sliding mode observer scheme," *2010 Conference on control and fault-tolerant systems (SysTol)*, IEEE, 2010, pp. 125–130.

[10] Henry, D., "Fault diagnosis of microscope satellite thrusters using H-infinity/H_- filters," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 699–711.

[11] Jiang, C., Zhang, S.-B., and Zhang, Q.-Z., "A new adaptive h-infinity filtering algorithm for the GPS/INS integrated navigation," *Sensors*, Vol. 16, No. 12, 2016, p. 2127.

[12] Khalil, A., Janaideh, M. A., and Kundur, D., "Online fault classification in Connected Autonomous Vehicles using output-only measurements," *Mechanical Systems and Signal Processing*, Vol. 190, 2023, p. 110099.

[13] Cattin, L., Zanotto, M., and Savaresi, S. M., "Model-based fault diagnosis for robotic systems: A survey," *Robotics and Autonomous Systems*, Vol. 123, 2020, p. 103360.

[14] Codetta-Raiteri, D., and Portinale, L., "Dynamic Bayesian networks for fault detection, identification, and recovery in autonomous spacecraft," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 45, No. 1, 2014, pp. 13–24.

[15] Dezan, C., Zermani, S., and Hireche, C., "Embedded bayesian network contribution for a safe mission planning of autonomous vehicles," *Algorithms*, Vol. 13, No. 7, 2020, p. 155.

[16] Eide, P., and Maybeck, P., "An MMAE failure detection system for the F-16," *IEEE Transactions on Aerospace and Electronic systems*, Vol. 32, No. 3, 1996, pp. 1125–1136.

[17] Hanlon, P. D., and Maybeck, P. S., "Multiple-model adaptive estimation using a residual correlation Kalman filter bank," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 36, No. 2, 2000, pp. 393–406.

[18] Carbone, M. A., Csank, J. T., Tomko, B. J., Follo, J. C., and Muscatello, M. J., "A Multiple Model Based Approach for Deep Space Power System Fault Diagnosis," *AIAA SciTech Forum 2019*, 2019.

[19] Cortes, C., and Vapnik, V., "Support vector machine," *Machine learning*, Vol. 20, No. 3, 1995, pp. 273–297.

[20] Jolliffe, I. T., and Cadima, J., "Principal component analysis: a review and recent developments," *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, Vol. 374, No. 2065, 2016, p. 20150202.

[21] Oja, E., and Hyvarinen, A., "Independent component analysis: algorithms and applications," *Neural networks*, Vol. 13, No. 4-5, 2000, pp. 411–430.

[22] Comon, P., and Jutten, C., *Handbook of Blind Source Separation: Independent component analysis and applications*, Academic press, 2010.

[23] Liu, S., Wang, B., and Zhang, L., "Blind source separation method based on neural network with bias term and maximum likelihood estimation criterion," *Sensors*, Vol. 21, No. 3, 2021, p. 973.

[24] Wang, Y., Jiang, B., Lu, N., and Pan, J., "Hybrid modeling based double-granularity fault detection and diagnosis for quadrotor helicopter," *Nonlinear Analysis: Hybrid Systems*, Vol. 21, 2016, pp. 22–36.

[25] Khoukhi, A., and Khalid, M. H., "Hybrid computing techniques for fault detection and isolation, a review," *Computers & Electrical Engineering*, Vol. 43, 2015, pp. 17–32.

[26] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, Vol. 60, No. 6, 2017, pp. 84–90.

[27] Hochreiter, S., and Schmidhuber, J., "Long short-term memory," *Neural computation*, Vol. 9, No. 8, 1997, pp. 1735–1780.

[28] Radford, A., Metz, L., and Chintala, S., "Unsupervised representation learning with

deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[29] Kingma, D. P., and Welling, M., "Auto-Encoding Variational Bayes," , 2022.

[30] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A., "Extracting and composing robust features with denoising autoencoders," *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.