Automated Anomaly Detection Ground Station for Small Satellite Attitude Dynamics Control System

by

Yujia Huang

A thesis
presented to the University of Manitoba
in fulfilment of the
thesis requirement for the degree of
Master of Science
in
Mechanical Engineering

Winnipeg, Manitoba, Canada 2022

Abstract

This thesis presents an automated ground-based anomaly detection system for an attitude dynamics system of a small satellite. In a space mission, a ground station works as the operation center to communicate and monitor the satellite. Human operators are required to stay at the ground station at all times to collect, track, and analyze the performance of the satellite. With the growing popularity of the Artificial Intelligence (AI) application in recent years, processes relating to data collection, transmission, and analysis have been replaced by machine learning which reduces human labour, expense and time, whereas shortcomings in the traditional human expert systems are improving, and applications to space systems fault detection are now possible.

Therefore, in this research, machine learning algorithms were applied to accomplish the proposed anomaly detection system. The detection system aimed to detect subtle failures in the spacecraft's attitude dynamics system, mainly in the reaction wheel subsystem, by only learning from nominal behavioural data from the spacecraft. The system was developed from a small satellite attitude dynamics control system using reaction wheels that may exhibit bearing failures. There were two types of anomaly detection systems introduced, including a two-sided learning anomaly detection system and a one-sided learning anomaly detection system.

For this study, I first developed the two-sided learning anomaly detection system using the Logistic Regression (LR) method, which provided a background of how the training process would be undertaken using a machine learning method. By learning from both nominal and failure behaviours from the satellite, the system was expected to detect small reaction wheel friction failures. Then, the one-sided learning anomaly detection system was built by only learning from nominal behaviours from the satellite and was expected to detect the same reaction wheel failures. The methods used to develope the one-sided learning system were: One-Class Support Vector Machine (OC-SVM) and One-Class Linear Regression (OC-LR). Two simulation tests with different friction failures were given to the two-sided and one-sided learning systems. Detection performance for each system was discussed. After demonstrating the proposed system in simulation, the one-sided learning system was verified by a real motor test. Similar to the simulations, the detection system was only trained by nominal behaviours from the motor and was expected to detect friction failures to distinguish between normal and abnormal motor motion. The simulation test and the motor anomaly test illustrated the feasibility and generality of the proposed one-sided learning fault detection system for space systems.

Acknowledgements

First, I would like to thank my parents and my husband for your love and support all the time. Thank you for always being with me, I could not take this journey without you all.

A huge thank you to my advisor, Dr. Philip Ferguson. The best mentor, professor, friend, and listener that I have ever met. Your patience, suggestions, and encouragement made me arrive here. Your courses, aerodynamics, space system, and space dynamics, are the enlightenment for me on space engineering.

I am also grateful to my committee members, Dr. Ahmed Ashraf and Dr. Igor Telichev, for providing me with valuable advice and guidance in the machine learning field and space engineering. It has provided me an insight into how machine learning knowledge can be used in a different area and to combine the machine learning knowledge with space engineering.

I would like to thank all my friends and lab members, especially to Mitesh Patel, Jaime Campos, Morgan May, and Ali Barari. Thanks for the edits, comments, feedback, and helping me with my physical test setup. I appreciate your time, patience, suggestions, and knowledge all the time.

Contents

List of Tables

List of Figures

Nomenclature

Acronyms/Abbreviations

ADCS	Attitude Dynamics Control System
AI	Artificial Intelligence
AUC	Area Under Curve
EER	Equal Error Rate
FFNN	Feed-Forward Neural Network
FPR	False Positive Rate
LAD	Logical Analysis of Data
LEO	Low Earth Orbit
MOC	Mission Operation Center
OC-LR	One-Class Linear Regression
OC-SVM	One-Class Support Vector Machine
PHM	Prognostic and Health Management
RF	Radio Frequency
ROC	Receiver Operating characteristic
SDR	Software-Defined Radio
SMO	Sequential Minimal Optimization
SVM	Support Vector Machine

TPR	True Positive Rate

Chapter 1

Introduction

1.1 Motivation[1]

A satellite ground station is a signal receiving and transmitting station for interacting with satellites in space. The ground station provides an interface with the communication system of the satellites by using Radio-frequency (RF) signals to telecommunicate with the satellites and to receive telemetry data from the satellites [3]. A typical ground station works as the Mission Operation Center (MOC) for a small satellite mission such as the UM station for Iris spacecraft (the spacecraft currently being designed and built in the STARLab at University of Manitoba) [4]. As the satellite and ground station communicate when the satellite is in a certain range, the MOC uplinks commands to the satellite, downlinks telemetry data from the satellite and monitors real-time performance of the satellite during contact passes - when the satellite is locationally over the contact ground station [4].

[1] This section is a slightly modified version based on the paper [1], [2] published in AIAA SciTech 2021 Forum and 35th Small 35th Annual Small Satellite Conference by Yujia Huang and Philip Ferguson, copyright © 2021 by Yujia Huang, reproduced with permission.

To monitor the performance of the satellite, telemetry data from the satellite will be inspected and analyzed by the operations team (Figure 1.1) [4]. Generally, ground station operation assumes that the normal behaviours, and operating information of the satellites are well-known and modelled. If the monitored information for the satellite shows signs of failures, the operators at the ground station will observe it right away and assess the situation.

Figure 1.1: Ground Station Operation Team at NASA [4]

A disadvantage for using this conventional ground station paradigm is that it needs technical operators and engineers to continuously monitor and track the satellites at the ground station at all times, requiring large expenses in labour, management and operations. Another common situation for the conventional ground station is that even highly-trained operators may only notice an issue after it has become large enough to cause a critical failure [5].

On the other hand, Prognostic and Health Management (PHM) has gained attention in space engineering as the size and prevalence of satellite constellations grow [6]–[8]. In keeping with the small-satellite philosophy to maintain low design, manufacturing and operating costs, the small-satellite community is interested in efficient ground operations and fault management that does not require excessive labor from trained space systems experts [9]–[12]. The expanding scale of small-satellite constellations has also posed a significant challenge for ground operations: how to find a sustainable way to monitor and manage a large amount of satellites efficiently with minimal cost? Therefore, the concept of an automated ground station was developed to reduce human workload and operational expenses, particularly for constellations of satellites where the monitoring task is even more onerous [13].

Automated satellite ground stations have become popular in recent years. Such ground stations are designed to automatically control and track a satellite's orbital motion, attitude and position around Earth [14]–[18]. In 2000, an automated ground station was designed which would detect the abnormal behaviour of satellites in off-hours and send back the information to remote operators by a pager or phones [19]. More recently, an automated ground station was able to detect and correct abnormal behaviours of satellites autonomously [20]. With the rapid development and widespread application of Artificial Intelligence (AI) technologies in space engineering, processes relating to data collection, transmission, and analysis have been replaced by machine learning systems to reduce human labour, expense, and time. As the number of AI application fields increase, shortcomings in traditional human expert systems are improving, and applications to space systems fault detection are now possible. Many researchers have relied upon machine learning to detect faults [1], [14]–[17], [21], [22], but this approach suffered from the limitation that it could only detect failures that have been previously trained or modelled. While ground station fault detection technology has become more mature and sophisticated, they still can only detect deviations from a well-known and fully understood model of nominal spacecraft operations. Sometimes, the spacecraft behaviour may not be well-known prior to launch and may even evolve as the mission progresses. Further, subtle failures may be difficult for experienced engineers to detect before they turn into catastrophic failures.

In many cases, the most subtle and dangerous failures are the ones without consideration prior to the mission, such as reaction wheel degradation failures. This type of the failure is one of the most dangerous anomalies that can happen in a satellite but is hard to predict as bearing degradation gradually grows with time [23]. Hence, being able to autonomously detect unmodeled faults is critical to the health of a constellation of small satellites, given that many spacecraft operators cannot afford to dedicate specialized staff to monitor all telemetry continuously to look for faults.

An interesting question then arises: how to design an autonomous ground station that could, over time, learn the normal behaviour of a spacecraft (without any prior knowledge of the spacecraft system) and report subtle deviations when detected? If we could implement such a ground station, how could the ground station learn from only normal behaviour of that unmodeled satellite and how could we differentiate between "normal" and "abnormal" behaviours in real-time, having never trained on abnormal data?

Therefore, this thesis aims to answer these questions by finding a way to design an autonomous, ground-based fault detection system that is trained using only nominal data, without requiring expert knowledge of the spacecraft systems. By observing nominal data during the commissioning phase of the satellite, the fault detection algorithm learns how to distinguish normal data from abnormal data without requiring a labelled set of abnormal data. Specifically, this research demonstrates the utility of one-sided learning methods by autonomously detecting faults in reaction wheel bearing friction, without any prior exposure to the failures.

1.2 Hypotheses

The principal purpose of my research was to design an automated anomaly detection system for a ground station monioring satellite data that is capable of learning the normal behaviour of a spacecraft over a period of time without having any prior knowledge of the spacecraft (trained on only normal data) and report subtle deviations and anomalies when detected. Two hypotheses have been posed below to direct this research project:

Hypotheses 1: I hypothesize that a ground station can learn the nominal behaviour of a spacecraft attitude control system after observing a specific failure-free period of a spacecraft operations using an artificial neural network or other suitable deep learning method.

Hypotheses 2: I hypothesize that a ground station can detect un-trained failures from the performance of the spacecraft attitude control system after a primary learning phase from a specific failure-free period.

1.3 Research Contributions

This thesis presents how a ground-based fault-detection system was developed. Multiple machine learning algorithms were applied in this research to achieve the proposed automated detection system. To develop the fault detection system, my research followed the three phases (Figure 1.2), including model simulation, two-sided learning-based anomaly detection system development and one-sided learning-based anomaly detection system development.

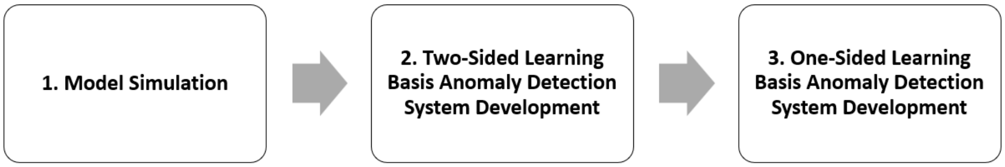
1. Model Simulation
2. Two-Sided Learning Basis Anomaly Detection System Development
3. One-Sided Learning Basis Anomaly Detection System Development

Figure 1.2: Principal Research Phases

An Attitude Dynamics Control System (ADCS) of a small satellite was simulated and utilized as the major subsystem for anomaly detection at ground station. A two-sided learning-based anomaly detection system was firstly developed to provide an insight into training and learning procedure for a machine-learning problem.

Major contributions of this thesis include:

1. A mathematical model of a spacecraft attitude control system, including a cluster of reaction wheels that enable custom bearing failures to be injected.
2. An evaluation of the performance of a two-sided logistic-function-based machine learning algorithm for detecting subtle changes in reaction wheel bearing friction based only on simulated downlinked satellite data.
3. An analysis and a validation of the performance of a one-sided logic-regression-based and a one-sided-classification-based machine learning algorithms for detecting subtle changes in reaction wheel bearing friction based on both simulated downlinked satellite data and physical motor experiments data.

1.4 Thesis Summary

Chapter 1 (this chapter) provided the motivation of this research, including a brief introduction and background of the satellite ground station. This chapter also included the research purpose, hypotheses, and major contributions.

Chapter 2 presents the background of the conventional and modern automated satellite ground stations. Algorithms and approaches that enabled the automated satellite ground station are discussed and summarized in this chapter.

Chapter 3 introduces the attitude dynamics control system model used to demonstrate the machine learning algorithms considered in this thesis. It describes the methodology of building a slew simulator and explains the basic motion dynamics of the satellite. This chapter also details how bearing degradation failures for reaction wheels were simulated for my research.

Chapter 4 introduces three different machine learning algorithms applied in this research, including Logistic Regression for Binary Classification, One-Class Support Vector Machine (OC-SVM) and One-Class Linear Regression (OC-LR) methods.

Chapter 5 presents the training data and training process for each proposed methods in Chapter 4 as well as the anomaly detection results from simulated test datasets with various types of bearing degradation failures. The anomaly detection test results and performance for each method are analyzed and discussed in this chapter.

Chapter 6 provides details of the physical motor tests, including the test setup and procedure, to validate the proposed machine learning methods in Chapter 4. The anomaly detection test results and performances are compared and discussed.

Chapter 7 summarizes the main research contributions, conclusion, and recommendations for future work.

Chapter 2

Background

This chapter provides background information on conventional ground stations and modern automated satellite ground stations. A literature review of the operating system in the conventional ground stations is provided as well as details of the data transmission process between ground stations and satellites. An overview of algorithms and methodology that have been implemented to achieve the automated satellite stations are presented.

2.1 Conventional Ground Stations

Currently, the conventional ground station architecture is commonly being used in most satellite missions. A typical conventional ground station is comprised of a receiver, modem, transmitter, and antennas to allow the uplinking and downlinking processes to take place during a space mission (Figure 2.1) [24]. As data transmitting only occurs during each contact pass, there is a time limitation on communication between the satellite and the ground station [9]. The operators are required to be trained priorly and stay on-site 24 hours to operate equipment, send uplink commands to the satellite, and interpret downlinked telemetry data from the satellite. [2], [13], [24].

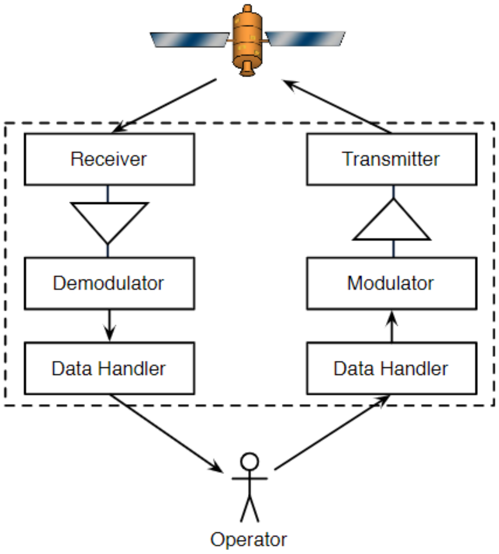

Figure 2.1: Traditional Ground Station Block Diagram [24]

All uplinked commands are planned and tested before the satellite launches to space, and the feedback performances from the satellite are rehearsed [13]. Any failures or signs of failures from the satellite rely on the data monitoring and observing from the on-site ground station operators. As previously mentioned, with the growing number of small-satellite constellations, numerous satellites need the operators to periodically monitor and examine the health of the satellites [2], [13]. The increasing number of space missions also results in increasing security needs, effective costs, operational efficiency and system complexity of the ground station [25], [26]. However, limited capacity of the ground station (e.g., staff, infrastructure, etc.) makes these difficult to accomplish [27]–[29].

To achieve efficient control and operation, some researchers [26], [30]–[32] used a remote-control approach without updating the current ground station architecture.

A separate ground station control system or a centralized interface was developed and utilized to remotely connect to the real operation ground station by cables or remote access software (e.g., TeamViewer, TCP/IP, etc.).

In [26], a remote ground station control system, Mercury, was developed by Stanford University's Space Systems Development Laboratory (SSDL) to provide a centralized software interface to control all ground station hardware. The operators were able to operate all equipment without being present in the ground station control room. A similar remote control system was established in [32]: an end-to-end ground station network worked as a common server to connect all ground station facilities together.

Another type of remote-controlled ground station was developed and tested by Budapest University of Technology and Economics for MaSat-1 CubeSat project [31]. This remote-controlled ground station worked as a secondary ground station for the satellite mission and was able to handle commands and receive telemetry data from the satellite. Done et al. [30] also introduced an online Software-Defined Radio (SDR) remote control ground station for Low Earth Orbit (LEO) amateur satellites. The SDR ground station receiver was connected to the internet which enabled the ground station to collect telementry data from various satellites simutaneously and provide an online data access to the operators.

To some extent, the remote control ground station achieves the automation of operations and significantly reduces labour, operating time, and costs since it no longer requires 24-hour on-site operation and the operators can control and access the telemetry data anywhere. Although different types of remote control ground stations improve the operational efficiency for a conventional ground station, it still needs human operators to monitor and capture anomalies from the downlinked data to track the satellite health.

2.2 Automated Ground Stations[2]

With the growing interest in machine learning and deep learning technology, there is an increased interest in developing attitude and orbit control tracking systems that are capable of autonomously detecting and diagnosing anomalies for various kinds of satellites [14]–[18]. Moreover, anomaly detection now takes a critical part in Prognostics and Health Management (PHM) and has been employed in many areas of space engineering [6]–[8], [33]. Such systems have the capability to reduce human labour leading to significant cost savings. One method of implementing autonomous fault detection is through the use of ground stations. There has been lots of research investigating the benefits of autonomous ground stations. This subsection explores some previous research done into autonomous ground stations.

An early autonomous ground station was developed as the Deep Space Tracking Station (DS-T) in 1999 [34], which was designed as a fully autonomous spacecraft ground station. The DS-T was able to autonomously track and operate the spacecraft and was capable of receiving and predicting the telemetry signals at the same time over a weekly time scale. This DS-T method succeeded in telemetry tracking and it could make subtle trajectory adjustments when it detected a deviation between the real and the predicted performance of the spacecraft.

One of the most popular approaches to detect satellite anomalies from an autonomous ground station is to set a value threshold for the satellite system. This kind of a system works as a "monitor" to observe the operational performance of the satellite [14], [35]. Once the satellite's performance exceeds the limits deemed to be "normal", the ground station will inform the operators. The operators then look

[2] This section is a slightly modified version based on the paper [1], [2] published in AIAA SciTech 2021 Forum and 35th Small 35th Annual Small Satellite Conference by Yujia Huang and Philip Ferguson, copyright © 2021 by Yujia Huang, reproduced with permission.

into the situation and provide solutions. This kind of ground station is “partially autonomous” and works only for a specific satellite with limitations when applied to different satellites. A more advanced approach is built upon on a model-based method where the ideal performance of the satellite is modeled and failures are detected from the nature of the deviations between the actual performance and ideal performance of the satellite [36]–[38]. The model-based system requires that an expert be extremely familiar with the satellite and would be able to precisely predict the ideal performance of the satellite in the absence of failures.

The DS-T method, the value-threshold method and the modeled-based method described above all have the advantages that they reduce human labour and expense, but there are still limitations. The disadvantages for such detection systems are that they only work for a specific type of satellite and require experts with prior knowledge to design and import numerous possibilities of failures and anomalies into the system for detection. While ground station fault detection technology has become more mature and sophisticated, they still can only detect deviations from a well-known and fully understood model of nominal spacecraft operations. In many cases, the nominal behaviour of a spacecraft may either not be known perfectly prior to launch, or it may evolve over time as parts age. Further, while it is often easy to detect significant failures such as the loss of an entire battery or solar cell string, other more subtle failures, such as a gradual increase in wiring harness resistance or bearing degradation in a reaction wheel, are much more difficult to detect and require more advanced modelling that may not be available.

To solve such limitations, in [15], [16], space-based fault detection systems are designed as “data-based” model systems. The performance of the satellite over a specific time period is collected and learned by the detection systems. The detection systems learn the behaviour from past data of the satellite and monitors the satellite in real-time to diagnose anomalies by comparing with the previously learned data

[17]. High-accuracy and high-confidence fault detection systems are developed in the machine learning and deep learning fields, which generally do not require a qualitative model or value threshold and has the capability for real-time fault detection. Ibrahim et al. [21] proposed a fault diagnosis method based on a combination of machine learning and data mining technologies to identify failures and anomalies of satellite subsystems for the Egyptsat-1 satellite. This paper did the investigation on three important subsystems of the satellite: communication subsystem, on-board computer subsystem and power subsystem. In this research, a Support Vector Machine (SVM) was applied to learn and predict the expected bus voltage of the satellite from received time-series telemetry parameters. With comparison of the predicted results, data mining techniques such as Logical Analysis of Data (LAD) were utilized to classify the binary categories of the satellite's behaviour ("normal" and "abnormal") and generate the behaviour patterns. From the behaviour patterns, a Fault Tree Analysis was used to determine the root cause and occurrence possibility of the failures occurred in each subsystem.

Omran et al. [22] developed a fault detection and identification system for reaction wheels in an attitude dynamics system. The fault detection system was created based on a feed-forward neural network (FFNN) with a back-propagation algorithm to detect if there was an anomaly in the reaction wheel voltage, current, and temperature. The system firstly used FFNN to predict the desired torque for a reaction wheel under a commanded bus voltage. Then, the normal operational torque curve was used as a comparable reference for anomaly detection. The difference in torques resulting from the failures (e.g., over and under-voltage, current gain and loss, and temperature increase and decrease) would then be observed.

A "data-based" model system does not require prior knowledge from experts and is capable of detecting anomalies based on data learning and prediction. This kind of system is widely applied in ground stations now to monitor satellites and identify anomalies. However, such a system is limited to learn a satellite's behaviour from two categories of data where both positive data and negative data are required to be given during learning phase.

Other approaches for fault detection were to identify and label data that deviated from acceptable ranges [16], [17]. Specific features were selected from the flight measurements of the spacecraft and collected over a designated time. Filters were added before the training process to remove the abnormal data beyond the pre-defined lower and upper limits of satellite performance. The system was trained using the filtered data contained within most of the normal data. The detection system then would predict a theoretical model from the trained data to detect anomalies in real-time operations of the satellite.

Most of the previous fault-detection systems have been trained using datapoints containing both normal behavioural data and abnormal behavioural data of the satellite (i.e, both positive data and negative data were given). Then, the system learned the "knowledge" from the two-sided training data and developed a binary classifier once the learning stage completed. Finally, the output binary classifier was used to predict failures and differentiate normal data and abnormal data during testing.

In this thesis, a ground-based automated novelty detection system was developed for a small satellite attitude dynamics control system using both two-sided and one-sided learning algorithm: Logistic Regression (two-sided), One-Class Support Vector Machine (OC-SVM) and One-Class Linear Regression (OC-LR) methods.

The Logistic Regression methodology is widely applied to develop novelty detection systems in many research fields [39]–[43], and it is a typical method to solve for the Binary Classification problem. Normally, to train a novelty detection system developed from the Logistic Regression method, it requires a "two-sided" dataset where the dataset contains data in two different categories or classes. Each of the datapoint in the training dataset will be labelled based on its categories. As the Logistic Regression is evolved from Linear Regression method, a linear seperable boundary is learned by the system from the labelled data. The classifier for the detection system is built from Sigmoid function. By applying this method in my research, normal and abnormal behavioural data of the satellite were labelled and given to the system to train the classifier.

The OC-SVM method was originally proposed by Schölkopf et al.[44] in 1999 and has been used in a variety of one-class classification problems [45]–[51]. The OC-SVM method only requires ”one-sided” data for training where all training data belongs to a single category. It assumes that majority of the training data is in a ”small” region [44]. To distinguish the training data with other data, a hyperplane with the maximum margin from the ”small” region will be learned from this method. The hyperplane works as the classifier for the OC-SVM method. If a datapoint falls in one side of the hyperplane is in one class, and a datapoint falls in the other side of the hyperplane is in another class. By using this approach, the fault-detection system was developed to only learn from nominal behaviour of the satellite during the commissioning phase and to identify and detect anomalies when there was a subtle behavioural failure in the attitude control system. The detection system was trained by only observing the nominal attitude dynamics behaviour of a small satellite for a period of time. Training data was obtained from reaction wheel outputs in a healthy attitude control system, and reaction wheel currents and angular velocities were selected as training features. A one-class classifier was built from a hyperplane decision function during training. An adaptive Sequential Minimal Optimization (SMO) method was utilized to solve the quadratic problem in the application of OC-SVM algorithm to provide an optimal solution for the hyperplane decision function.

The One-Class Linear Regression is a modified method in this research from Linear Regression method in statistics [52], [53]. This method also only requires ”one-sided” data for training. An approximate function will be obtained between the data after training. The one-class classifer for this method is built from Empirical Rule (68-95-99.7 Rule), where the limits to classify datapoints are from 1σ, 2σ and 3σ from the training data mean.

Two tests were performed on the developed anomaly detection systems to validate its feasibility and detection accuracy. Untrained reaction wheel bearing failures were added into the attitude control system validation tests to examine whether the fault-detection systems were capable of detecting and diagnosing the reaction wheel failures. Details of training and testing performances of the fault-detection system will be presented in the following chapters.

Chapter 3

Attitude Dynamics Control System Simulation

This chapter povides details of how an attitude dynamics control system model was established. Two subsections are included in this chapter. Section 3.1 presents the methodology of developing a slew simulator of the satellite. The dynamics of the reaction wheel are presented in Section 3.2. Section 3.2 also illustrates how the bearing degradation failures were added into the reaction wheel subsystem.

3.1 Spacecraft Attitude Dynamics Control System Simulation[3]

In this section, an attitude control system model is developed. The attitude control system mainly focuses on the orientation and pointing direction of the spacecraft

[3] This section is a slightly modified version based on the paper [1], [2] published in AIAA SciTech 2021 Forum and 35th Small 35th Annual Small Satellite Conference by Yujia Huang and Philip Ferguson, copyright © 2021 by Yujia Huang, reproduced with permission.

[54]. The development of the attitude control system model started from the basic rotational motion dynamics of the spacecraft. Next, I designed a closed-loop feedback system to control the spacecraft dynamics (shown in Figure 3.1)).

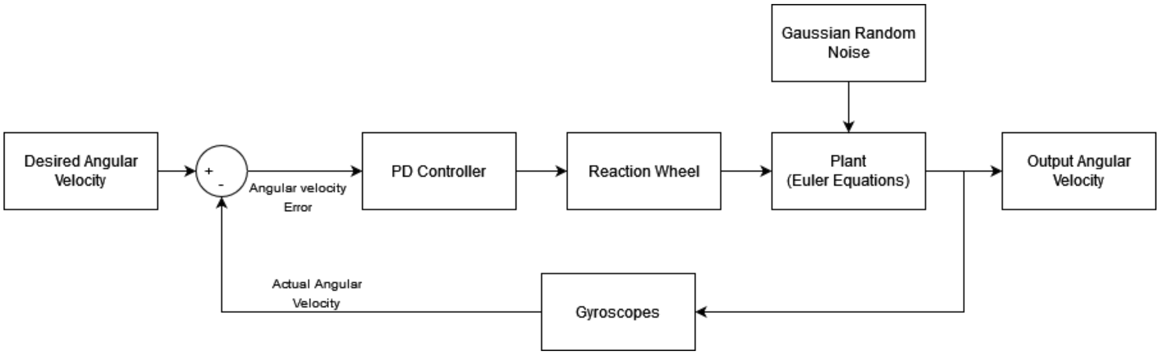

Figure 3.1: Closed-Loop Attitude Control System of the Spacecraft

The desired angular velocity of the spacecraft was first set as an input to the closed-loop system. To achieve the desired motion of the spacecraft, reaction wheels were utilized to generate and apply torque to control the spacecraft. There were three reaction wheels in the attitude control system. The wheels were oriented along the spacecraft's x, y and z body axes to provide sufficient torque to the spacecraft. The transformation between angular momentum and torque were governed by Euler's theorem [54] where time differentiation of angular momentum was equal to the torque (shown in Eq. 3.1).

$$\boldsymbol{T}(t) \;= \dot{\boldsymbol{h}}(t) \tag{3.1}$$

Where, t is time, $\boldsymbol{T}(t)$ is the torque, $\boldsymbol{h}(\mathrm{t})$ is the angular momentum.

Gaussian random noise was added into the system as an external noise to simulate disturbances on the spacecraft to obtain a more realistic model. With the actuation from reaction wheels, the spacecraft was able to achieve the desired attitude motion. The total applied torque to the system was

$$\boldsymbol{T}_{total}(t) = \boldsymbol{T}_{rw}(t) + \boldsymbol{T}_{noise}(t) \tag{3.2}$$

Where, $\boldsymbol{T}_{total}(t)$ is the total torque of the attitude control system, $\boldsymbol{T}_{rw}(t)$ is the torque from reaction wheel system, $\boldsymbol{T}_{noise}$ is the torque from added Gaussian random noise.

As the magnitude of the torque from the noise was very small (2.5 $\times 10^{-6}$ Nm), $\boldsymbol{T}_{noise}$ was assumed to be neglected in the Eq. 3.2. Therefore, the total torque was equal to the torque from the reaction wheel system (Eq. 3.3). The reaction wheel equation of motion is shown in Eq. 3.4.

$$\boldsymbol{T}_{total}(t) = \boldsymbol{T}_{rw}(t) \tag{3.3}$$

$$\boldsymbol{T}_{rw}(t) = \boldsymbol{I}_{rw}(t)\dot{\boldsymbol{\omega}}_{desire}(t) \tag{3.4}$$

Where, $\boldsymbol{I}_{rw}$ is the moment of inertia of the reaction wheel and $\boldsymbol{\omega}_{desire}(t)$ is the desired angular velocity of the reaction wheel.

Then, the momentum of the system can be modeled as the sum of the spacecraft and reaction wheel momentum as shown in Eq. 3.5 [55].

The system torque can then be expressed in terms of the system momentum as in Eq. 3.6 [55].

$$\boldsymbol{h}(t) = \boldsymbol{h}_{spacecraft}(t) + \boldsymbol{h}_{rw}(t) = \boldsymbol{I}\boldsymbol{\omega}(t) + \boldsymbol{h}_{rw}(t) \tag{3.5}$$

$$\boldsymbol{T}(t) = \dot{\boldsymbol{h}}(t) + \boldsymbol{\omega}(t) \times \dot{\boldsymbol{h}}(t) = \boldsymbol{I}\dot{\boldsymbol{\omega}}(t) + \boldsymbol{\omega}(t) \times (\boldsymbol{I}\boldsymbol{\omega}(t) + \boldsymbol{h}_{rw}(t)) \tag{3.6}$$

Where, $\boldsymbol{h}_{spacecraft}(t)$ is the angular momentum of the spacecraft, $\boldsymbol{I}$ is the moment of inertia of the spacecraft, $\boldsymbol{\omega}(t)$ is the angular velocity of the spacecraft, $\boldsymbol{h}_{rw}(t)$ is the angular momentum of the reaction wheels.

The frame of the momoment of inertia was defined from the center of mass of the spacecraft where the principal axes frame was assumed as the body frame. Therefore,

$$\mathbf{I} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \tag{3.7}$$

Where, $\boldsymbol{I}$ is a diagonal matrix, I_1, I_2, and I_3 are the principal moments of intertia.

A feed-forward torque was added to the attitude control system as shown in Eq. 3.8, which was controlled by the wheel speed measurements and the gyro measurements. The feed-forward torque was then generated from the desired angular rate of the spacecraft in Eq. 3.9 [55].

$$\boldsymbol{I}\dot{\boldsymbol{\omega}}(t) = \boldsymbol{T}_{rw_{real}}(t) + \boldsymbol{T}_{FF}(t) \tag{3.8}$$

$$\boldsymbol{T}_{FF}(t) = \boldsymbol{I}\dot{\boldsymbol{\omega}}_{desire}(t) + \boldsymbol{\omega}_{desire}(t) \times (\boldsymbol{I}\boldsymbol{\omega}_{desire}(t) + \boldsymbol{h}_{rw_{measured}}(t)) \tag{3.9}$$

Where, $\boldsymbol{T}_{rw_{real}}(t)$ is the real output torque required by the reaction wheels, $\boldsymbol{T}_{FF}(t)$ is the feed-forward torque, $\dot{\boldsymbol{\omega}}_{desire}$ is the desired angular acceleration of the spacecraft, $\boldsymbol{\omega}_{desire}(t)$ is the desired angular velocity of the spacecraft, $\boldsymbol{h}_{rw_{measured}}(t)$ is the measured angular momentum of the reaction wheels.

The real spacecraft angular velocity from the system was measured by gyroscopes. A PD controller was used to control, track, and correct the difference between the real and desired motions of the spacecraft [56]. The angular velocity error was inputted into the PD controller to output a torque command as shown in 3.11.

$$\boldsymbol{\omega}_{error}(t) = \boldsymbol{\omega}_{desire}(t) - \boldsymbol{\omega}(t) \tag{3.10}$$

$$\boldsymbol{T}_{PD}(t) = k_p \boldsymbol{\omega}_{error}(t) + k_d \frac{d}{dt} \boldsymbol{\omega}_{error}(t) \tag{3.11}$$

Where, $\boldsymbol{T}_{PD}(t)$ is the torque generated from a PD controller, $\boldsymbol{\omega}_{error}(t)$ is the angular velocity difference between the desired value and the real output of the spacecraft angular velocity. The proportional gain (k_p) and the derivative gain (k_d) were tuned to obtain the desired spacecraft performance [56].

From the closed-loop system diagram shown in Figure 3.1, the commanded reaction wheel torques were the combination between the PD control torque and the feed-forward torque. Thus,

$$\boldsymbol{T}_{rw}(t) = \boldsymbol{T}_{FF}(t) + \boldsymbol{T}_{PD}(t) \tag{3.12}$$

Where, $\boldsymbol{T}_{PD}(t)$ was the torque generated from a PD controller.

In the next section, details of how a reaction wheel subsystem has been developed will be illustrated.

3.2 Dynamics of Reaction Wheel[4]

As shown in Figure 3.1, the reaction wheels work as actuators in the attitude control system of the spacecraft. A diagram shows the interior structure of the reaction wheel is presented in Figure 3.2, where a typical reaction wheel system contains a top bearing, a bottom bearing, two fly wheels and a DC brushless motor. An example of a commercial reaction wheel product from Microsat Systems Canada Inc. is shown in Figure 3.3.

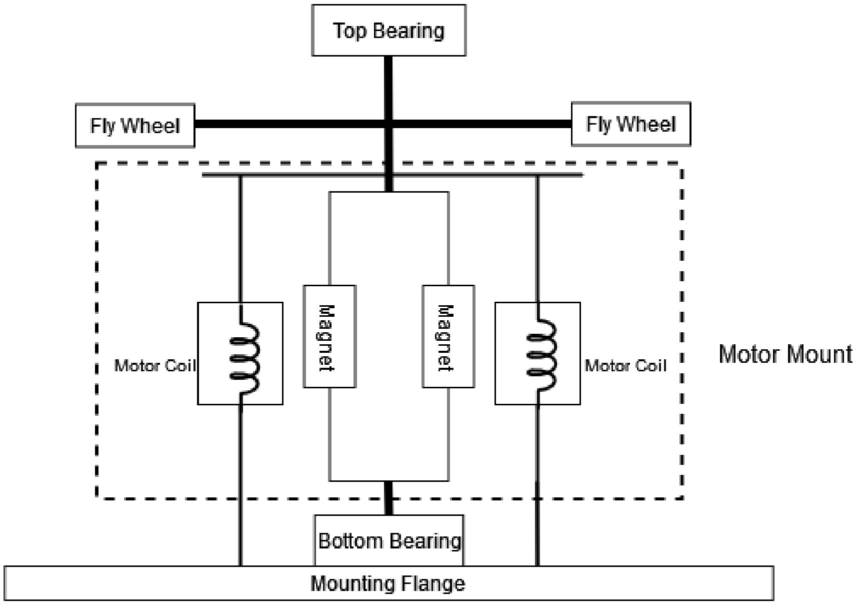

Figure 3.2: Reaction Wheel Diagram

[4] This section is a slightly modified version based on the paper [1], [2] published in AIAA SciTech 2021 Forum and 35th Small 35th Annual Small Satellite Conference by Yujia Huang and Philip Ferguson, copyright © 2021 by Yujia Huang, reproduced with permission.

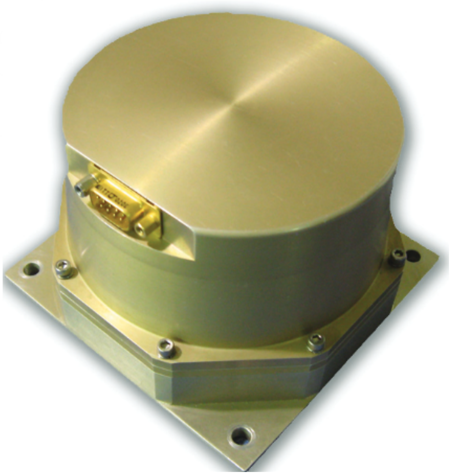

Figure 3.3: MSCI MicroWheel 1000 [57]

The reaction wheel model was directly generated from the nominal reaction wheel torque from Eq. 3.12. As mentioned earlier, three reaction wheels were simulated in the attitude control system. A similar closed-loop feedback system was designed for the reaction wheel speed controller in Simulink where a motor dynamic system was also included to simulate a DC brushless motor in each reaction wheel. The modelled system is shown below in Figure 3.4.

The commanded torque for the reaction wheel system was the torque intended to be applied to the spacecraft (Eq. 3.12). Thus, the command reaction wheel torque was expressed as:

$$\boldsymbol{T}_{command}(t) = -\boldsymbol{T}_{rw}(t) \tag{3.13}$$

Where, $\boldsymbol{T}_{command}(t)$ is the input command reaction wheel torque for the reaction wheel system.

Then, the commanded angular velocity rate for the reaction wheel system was attained from the nominal moment of inertias for the reaction wheels,

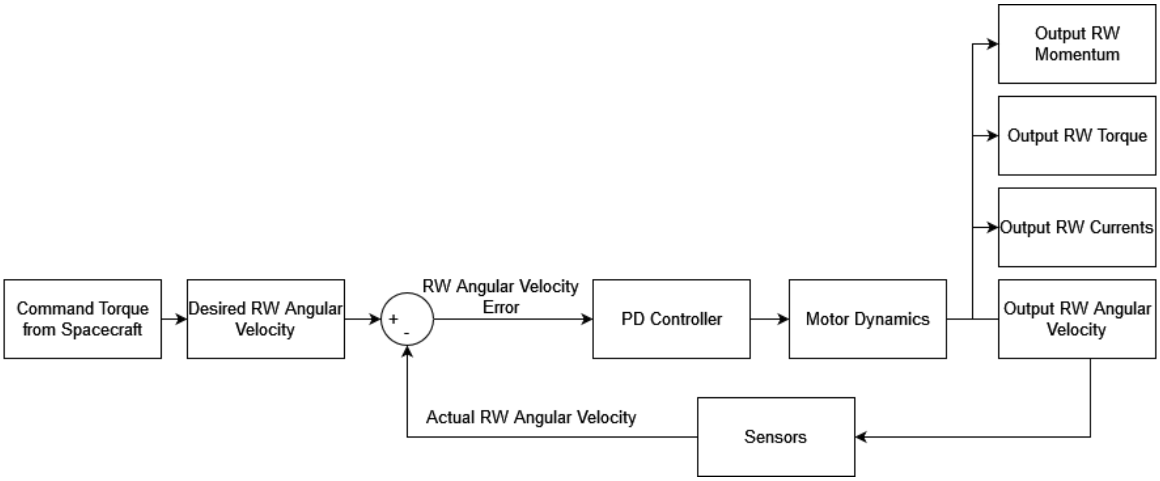
Output RW Momentum
Output RW Torque
Output RW Currents
Command Torque from Spacecraft
Desired RW Angular Velocity
+
-
RW Angular Velocity Error
PD Controller
Motor Dynamics
Output RW Angular Velocity
Actual RW Angular Velocity
Sensors

Figure 3.4: Closed-Loop Feedback System for Reaction Wheel

$$\dot{\boldsymbol{\omega}}_{rw_{command}}(t) = \frac{\boldsymbol{T}_{rw}(t)}{\boldsymbol{I}_{rw}} \tag{3.14}$$

Where, $\dot{\boldsymbol{\omega}}_{rw_{command}}(t)$ is the commanded angular acceleration of the reaction wheel system, and $\boldsymbol{I}_{rw}$ is the moment of inertia of reaction wheel.

The commanded angular velocity of reaction wheel then was computed as the time integration of the commanded wheel acceleration,

$$\boldsymbol{\omega}_{rw_{command}}(t) = \int \dot{\boldsymbol{\omega}}_{rw_{command}}(t)\ dt \tag{3.15}$$

Where, $\boldsymbol{\omega}_{rw_{command}}(t)$ is the commanded angular velocity of the reaction wheel.

The PD controller for the reaction wheel took the calculated difference between the commanded reaction wheel angular velocity and the real output reaction wheel angular velocity as an input error to output a voltage [56]. The motor voltage was then obtained from the wheel angular velocity difference with the PD tuning gains (shown in Eq. 3.17).

$$\boldsymbol{\omega}_{rw_{error}}(t) = \boldsymbol{\omega}_{rw_{command}}(t) - \boldsymbol{\omega}_{rw_{real}}(t) \tag{3.16}$$

$$\boldsymbol{V}(t) = k_{rw_p}\boldsymbol{\omega}_{rw_{error}}(t) + k_{rw_d}\frac{d}{dt}\boldsymbol{\omega}_{rw_{error}}(t) \tag{3.17}$$

Where, $\boldsymbol{\omega}_{rw_{real}}(t)$ is the real output angular velocity of the reaction wheel, $\boldsymbol{V}(t)$ is the motor voltage computed from the PD controller, k_{rw_p} and k_{rw_d} are the proportional and derivative gains of the PD controller respectively in the reaction wheel system [56]. With the computed motor voltage from the PD controller, the electrical circuit for the motor is presented in Figure 3.5.

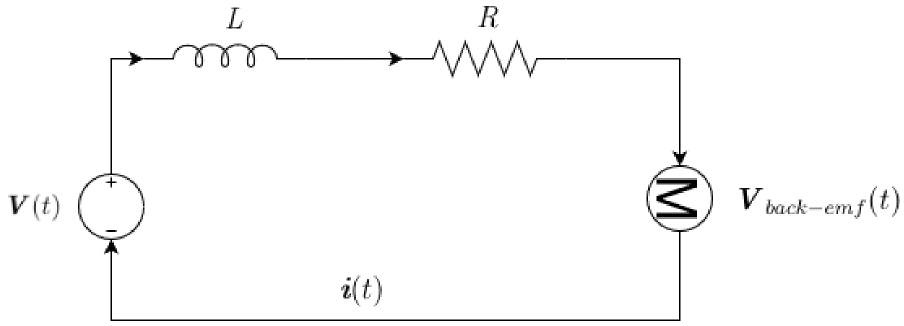

Figure 3.5: Motor Electric Circuit

From the motor electric circuit, the relatioship used to calculate the output current of the reaction wheel can be expressed as shown in Eq. 3.18.

$$\boldsymbol{V}(t) = R\boldsymbol{i}(t) + L\frac{di}{dt} + \boldsymbol{V}_{back-emf}(t) \tag{3.18}$$

Where, R is the resistance of the motor, $\boldsymbol{i}(t)$ is the reaction wheel current, L is the inductance, $\boldsymbol{V}_{back-emf}(t)$ is the back EMF which is proportional to the motor speed. The equation of the $\boldsymbol{V}_{back-emf}(t)$ is shown below:

$$\boldsymbol{V}_{back-emf}(t) = k_e \boldsymbol{\omega}_{motor}(t) = k_e \boldsymbol{\omega}_{rw_{real}}(t) \tag{3.19}$$

Where, k_e is the back EMF coefficient, $\boldsymbol{\omega}_{motor}(t)$ is the motor speed which is equal to the reaction wheel real speed. Since the inductance of a motor is significantly smaller than its resistance, the effect of the inductance of the motor on its dynamics were assumed to be negligible and hence ignored in the reaction wheel model [58]. Then, the motor electric circuit became:

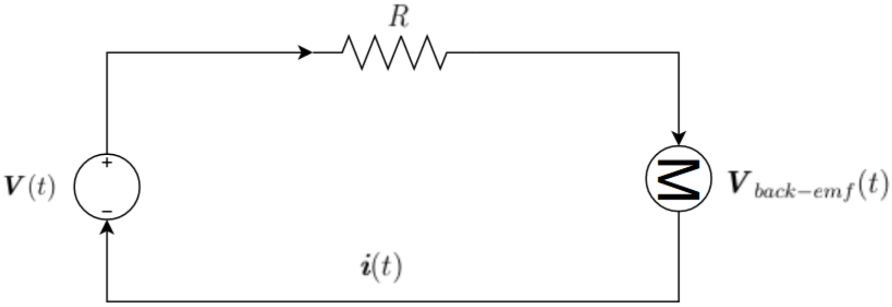

Figure 3.6: Motor Electric Circuit (After Neglecting the Inductance Effect)

And Eq. 3.18 became:

$$\boldsymbol{V}(t) = R\boldsymbol{i}(t) + \boldsymbol{V}_{back-emf}(t) \tag{3.20}$$

The provided toque from the motor was:

$$\boldsymbol{T}_m(t) = K_t \boldsymbol{i}(t) \tag{3.21}$$

Where, $\boldsymbol{T}_m(t)$ is the motor torque, K_t is the motor torque constant.

The actual provided torque generated by the motor in the reaction wheels was calculated after subtracting the total friction from the system (shown in Eq. 3.22).

$$\boldsymbol{T}_{rw_{provide}}(t) = \boldsymbol{T}_m - \boldsymbol{f}_{total} \tag{3.22}$$

$\boldsymbol{T}_{rw_{provide}}(t)$ is the real torque provided from motor to the reaction wheel. $\boldsymbol{f}_{total}$ is the total friction.

The total friction was composed of Coulombic and viscous friction and was added to the motor to model energy loss in the motor dynamics. The total friction is shown in 3.23.

$$\boldsymbol{f}_{total} = \boldsymbol{f}_{coulomb} + \boldsymbol{f}_{viscous} \tag{3.23}$$

Where, $\boldsymbol{f}_{coulomb}$ is the Coloumbic friction, and $\boldsymbol{f}_{viscous}$ is the viscous friction.

The nominal viscous fiction was calculated based on the real reaction wheel speed, where:

$$\boldsymbol{f}_{viscous_{nominal}} = k_f \boldsymbol{\omega}_{rw_{real}} \tag{3.24}$$

In this equation, k_f is the viscous friction coefficient, $\boldsymbol{f}_{viscous_{nominal}}$ is the nominal viscous friction which is calculated from the real reaction angular velocity and the viscous friction coefficient to compute all normal slews.

Hence, the function of calculating the total friction was then determined to be [59]:

$$\boldsymbol{f}_{total} = sign(\boldsymbol{\omega}_{rw_{real}}) \cdot (\boldsymbol{f}_{coulomb}) + k_f \boldsymbol{\omega}_{rw_{real}} \tag{3.25}$$

Since this research only focused on the viscous friction failure, the Coloumbic friction was assumed to be a constant (5 $\times 10^{-5}$ Nm/(rad/s)). Therefore, the function of calculating the viscous friction was determined to be proportional to the nomimal viscous friction:

$$\boldsymbol{f}_{total} = \boldsymbol{f}_{viscous} = m\boldsymbol{f}_{viscous_{nominal}} \tag{3.26}$$

$\boldsymbol{f}_{total}$ is the total friction in the motor dynamics, $\boldsymbol{f}_{viscous}$ is the viscous friction, m ($m \geq 1$) is a scaled factor which will be identified as a failure indicator through training process (if $m = 1$, $\boldsymbol{f}_{viscous} = \boldsymbol{f}_{viscous_{nominal}}$).

Then the real angular acceleration and velocity outputs for the reaction wheel system were calculated from Eq. 3.27 to Eq. 3.28

$$\dot{\boldsymbol{\omega}}_{rw_{real}}(t) = \frac{-\boldsymbol{T}_{rw_{provide}}(t)}{\boldsymbol{I}_{rw}} \tag{3.27}$$

$$\boldsymbol{\omega}_{rw_{real}}(t) = \int \dot{\boldsymbol{\omega}}_{rw_{real}}(t)\ dt \tag{3.28}$$

Where, $\dot{\boldsymbol{\omega}}_{rw_{real}}(t)$ is the real reaction wheel angular velocity rate output, $\boldsymbol{I}_{rw}$ is the moment of inertia of the reaction wheel and $\boldsymbol{\omega}_{rw_{real}}(t)$ is the real reaction wheel angular velocity output.

The real reaction wheel angular momentum and torque can be calculated using Eq. 3.29 and Eq. 3.30.

$$\boldsymbol{h}_{rw_{real}}(t) = \boldsymbol{I}_{rw}\boldsymbol{\omega}_{rw_{real}}(t) \tag{3.29}$$

$$\boldsymbol{T}_{rw_{real}}(t) = \boldsymbol{I}_{rw}\dot{\boldsymbol{\omega}}_{rw_{real}}(t) \tag{3.30}$$

Where, $\boldsymbol{h}_{rw_{real}}(t)$ is the real angular momentum output from the reaction wheel system, $\boldsymbol{T}_{rw_{real}}(t)$ is the real torque output from the reaction wheel system provided to the attitude control system. Note that this signal is only available in simulation and not available as a piece of telemetry from the reaction wheel.

An example of a healthy closed-loop attitude control system for a 1000 second slew period at the commissioning phase of a representative spacecraft has been developed based upon a specific slew trajectory in MATLAB and Simulink (Shown in Figure 3.7 and Figure 3.8). Figure 3.7 represents the input desired angular velocity of the spacecraft. Figure 3.8 present the measured output slew rates of the spacecraft from the gyroscopes with a random noise of ±0.0000025 rad/s.

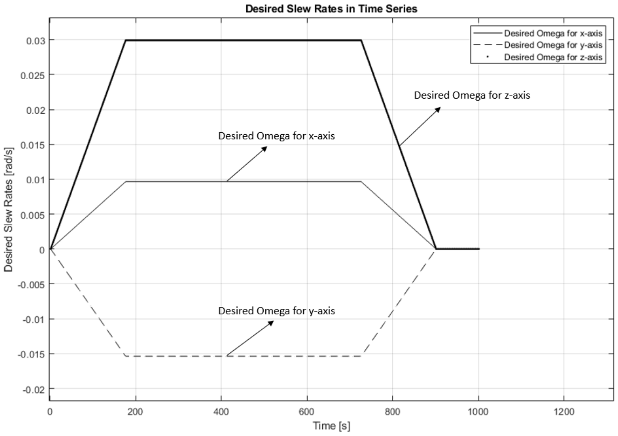

Figure 3.7: Desired Angular Velocity of the Simulated Spacecraft

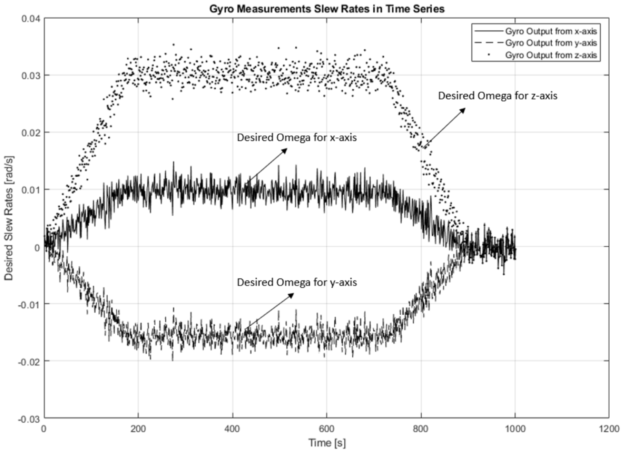

Figure 3.8: Output Body Rates of the Simulated Spacecraft

Chapter 4

Machine Learning Methods

This chapter explains how the proposed anomaly detection system was developed for the automated ground-based anomaly detection system as well as the methodoly and algorithms applied to accomplish the system. Two types of anomaly detection systems will be introduced in this chapter, including two-sided learning basis anomaly detection system and one-sided learning basis anomaly detection system. Three methods will be introduced in this chapter, including Logistic Regression, One-class Support Vector Machine (OC-SVM) and One-Class Linear Regression (OC-LR).

4.1 Two-Sided Learning Basis Detection System[5]

In this section, an automated anomaly detection system has been developed based on a two-sided machine learning algorithm as a precursor to the one-sided learning task for the next research phase. Although the two-sided learning algorithm is not

[5] This section is a slightly modified version based on the paper [1], [2] published in AIAA SciTech 2021 Forum and 35th Small 35th Annual Small Satellite Conference by Yujia Huang and Philip Ferguson, copyright © 2021 by Yujia Huang, reproduced with permission.

very practical in the real world as such algorithm requires both normal and abnormal dataset for training where the abnormal data is from expected failure that can be predicted, it is still a good baseline in this research to compare with the proposed one-sided algorithms. The training dataset was created to have both normal and abnormal data from the simulated model. The abnormal training dataset was created from a simulated failure in one of reaction wheels. The failure was selected as bearing degradation in the reaction wheel. With bearing wear, the reaction wheel viscous friction constant increased with time.

I started by collecting data from the normal slews and the slews with failures into one singular dataset. To train the system, the first step was to select the appropriate spacecraft (and reaction wheel) telemetry to use for training. The output data available from the attitude control system included:

- Measured reaction wheel current (Amps)
- Measured reaction wheel angular velocity (rad/s)
- Measured reaction wheel angular accleration (rad/s^2)
- Commanded reaction wheel torque (Nm)
- Computed reaction wheel momentum, derived from the measured angular velocity (Nm/s)

For this research, I chose three features as the most effective for learning viscous friction: measured reaction wheel current, measured reaction wheel angular velocity and measured reaction wheel angular acceleration. The reason for the selection was the implicit dependence of commanded reaction wheel torque and reaction wheel momentum on the reaction wheel current, angular velocity and angular acceleration (Shown in Eq. 3.21, Eq. 3.22, Eq. 3.29, and Eq. 3.30). Figure 4.1 shows the relationship of reaction wheel current, angular velocity and angular acceleration. As the reaction wheel accelerates, it takes more currents and vice versa. In certain instances, during normal slews, when there is more friction, the reaction wheel also takes more current.

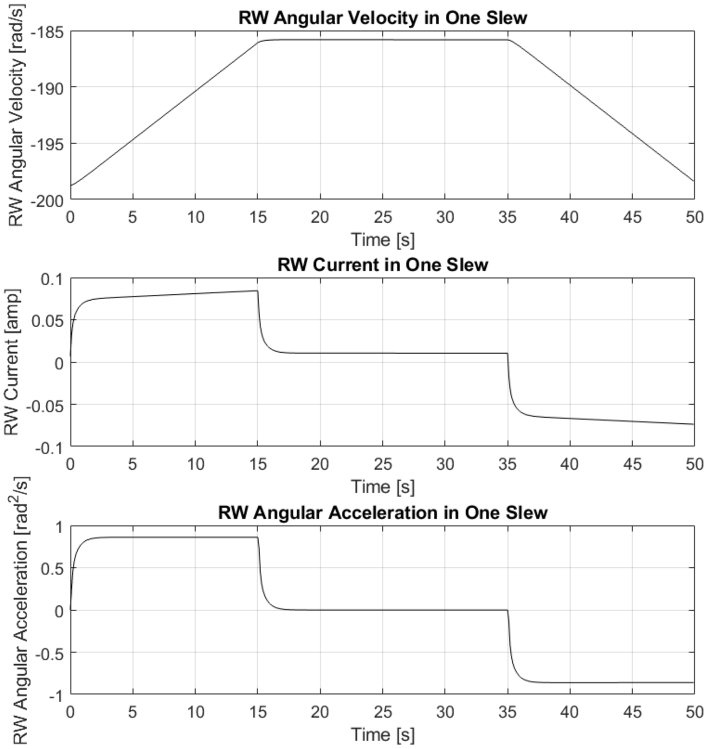

Figure 4.1: Relationship of Reaction Wheel Current, Angular Velocity and Angular Acceleration

By observing a brief time history, the classifier should be able to differentiate between currents associated with viscous friction, versus those associated with rotor acceleration.

The second step was to add labels to the dataset to differentiate the behaviour of the spacecraft between “normal” and “abnormal”: label 0 represented the normal data; label 1 represented the data with failures (shown in Eq. 4.1).

$$y = \begin{cases} 0 & \text{normal} \\ 1 & \text{abnormal} \end{cases} \tag{4.1}$$

For every time point, three measurements: reaction wheel current, angular velocity and angular acceleration were taken as one training data point with one response label.

$$\boldsymbol{x}_{tc} = \{(i_{tc}, \omega_{tc}, \dot{\omega}_{tc}, y_{tc})\} \tag{4.2}$$

$$\boldsymbol{X}_{tc} = \{(\boldsymbol{x}_{tc_1}, \boldsymbol{x}_{tc_2}, ..., \boldsymbol{x}_{tc_N})\} \tag{4.3}$$

Where $\boldsymbol{x}_{tc}$ is a subset of training data for two-sided learning method at a single timepoint, which contains both normal and failure data. i_{tc} is the reaction wheel current data at the timepoint. ω_{tc} and $\dot{\omega}_{tc}$ are the reaction wheel angular velocity data respectively at the same timepoint. y_{tc} is the assigned label for each training data from Eq. 4.1. $\boldsymbol{X}_{tc}$ is a vector that contains all training data with a total number of N training data sets.

The labelled behaviour dataset was utilized as the whole training set and was learned using Linear Logistic Regression for Binary Classification [60]. The system was trained with this set of training data and predicted the parameters of a decision boundary function that relates the reaction wheel current, the angular velocity and the angular acceleration to a nominal value of viscous friction. The learned decision boundary function could be represented as:

$$\boldsymbol{H}(\boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{X}_{tc} \tag{4.4}$$

Where, $\boldsymbol{H}$ is the prediction function which contains all predicted value from learned function for each training data, and $\boldsymbol{\theta}$ is a vector contains all learned function parameters.

The learned function was fitted with each training data point and imported to the Sigmoid function for labelling. The Sigmoid function is shown in Eq. 4.5 [60]:

$$\boldsymbol{S}(\boldsymbol{\theta}) = \frac{1.0}{(1 + e^{-\boldsymbol{H}(\boldsymbol{\theta})})} \tag{4.5}$$

Where, $\boldsymbol{S}(\boldsymbol{\theta}) \in \mathbb{R}$ [0, 1].

Failure detection and identification during real-time normal operations followed the learning phase. The system was tested with different sets of time-series testing data to examine its detecting capability. The system monitored the spacecraft and diagnosed the behaviour with learned model, where

$$y = \begin{cases} 1 & \boldsymbol{S}(\boldsymbol{\theta}) \geq 0.5 \\ 0 & \boldsymbol{S}(\boldsymbol{\theta}) < 0.5 \end{cases} \tag{4.6}$$

If $\boldsymbol{S}(\boldsymbol{\theta})$ was equal or greater than 0.5, the predicted label was designated to be 1 (denoting a failure); else, the predicted label was 0 [60].

4.2 One-Sided Learning Basis Detection System[6]

In this section, a one-sided learning-based detection system has been created. As mentioned in the previous section, the one-sided learning algorithm is more practical

[6] This section is a slightly modified version based on the paper [1], [2] published in AIAA SciTech 2021 Forum and 35th Small 35th Annual Small Satellite Conference by Yujia Huang and Philip Ferguson, copyright © 2021 by Yujia Huang, reproduced with permission.

than the two-sided learning method. The reason for this is that the one-sided learing algorithm does not require any abnormal dataset for training, which indicates that even without knowing any specific failure ahead, the system should be still capable of capturing failures. On the other hand, in the commissioning phase for any space mission, data collected from the satellite is all normal. Therefore, there is more interest to develop a one-sided learning basis anomaly detection system rather than building a two-sided learning basis anomaly detection system. The designed one-sided learning basis system only used nominal slews as a primary training dataset from the simulated model where the training dataset did not contain any failures. Two methods were proposed to develop the one-sided learning-based detection system. The first method was to apply One-Class Support Vector Machine method (OC-SVM) ([38], [44], [61], [62]) to train the detection system. The second method applied a One-Class Linear Regression algorithm (OC-LR) combined with statistical analysis to accomplish the detection system ([52], [53]).

4.2.1 One-Class Support Vector Machine (OC-SVM)

The principle of using the OC-SVM methodology is that labels or data responses are not required for training. Therefore, the training process only included normal behavioural attitude control system data. For this method, I started with data collection following the same steps when developing the two-sided learning basis detection system, except this time, the training data did not include any failures. Reaction wheel current, angular velocity and angular acceleration were selected as the training features to enable the algorithm to deduce relevant time-series features. The data of reaction wheel current, angular velocity and angular acceleration were collected at each time point.

Then, the training data then will be given as:

$$\boldsymbol{x}_{oc} = \{(i_{oc}, \omega_{oc}, \dot{\omega}_{oc})\} \tag{4.7}$$

$$\boldsymbol{X}_{oc} = \{(\boldsymbol{x}_{oc_1}, \boldsymbol{x}_{oc_2}, ..., \boldsymbol{x}_{oc_N})\} \tag{4.8}$$

Where, $\boldsymbol{x}_{oc}$ is a subset of training data for OC-SVM method, i_{oc} is the nominal reaction wheel current at one timepoint, ω_{oc} is the corresponding reaction wheel angular velocity, and $\dot{\omega}_{oc}$ is the corresponding reaction wheel angular acceleration. $\boldsymbol{X}_{oc}$ is the whole training data set of nominal data with a total of N training data sets.

All nominal data were used to train the detection system using OC-SVM to differentiate the behavioural labels. The OC-SVM algorithm trained the system to learn the decision hyperplane with maximum margin to separate normal behavioural data points with untrained abnormal behavioural data points from the origin in its feature space. Before training the system, the input training data was first mapped by a Kernel function into the feature space ([44], [63]):

$$K(\boldsymbol{X}_{oc}, \boldsymbol{X}'_{oc}) = \Phi(\boldsymbol{X}_{oc})\Phi(\boldsymbol{X}'_{oc}) \tag{4.9}$$

Where, K is the Kernel function (e.g., Gaussian, linear, sigmoid, polynomial Kernel functions), and Φ is a mapping function.

General Kernel functions that can be used for OC-SVM are listed in Table 4.1. σ, a, c, and d shown in the functions below are the tuning parameters in the process of training.

In this research, the Gaussian Kernel function was used to map the training features. As the Gaussian Kernel function is a commonly used mapping function for

Table 4.1: General Kernel Functions for OC-SVM ([45], [46], [64])

Name	**Kernel Functions**
Gaussian	$$k(x_i, x_j) = exp(-\frac{\|x_i, x_j\|}{2\sigma^2}) \tag{4.10}$$
Linear	$$k(x_i, x_j) = \mathrm{a}{x_i}^T x_j + c \tag{4.11}$$
Sigmoid	$$k(x_i, x_j) = tanh(\mathrm{a}{x_i}^T x_j + c) \tag{4.12}$$
Polynomial	$$k(x_i, x_j) = (\mathrm{a}{x_i}^T x_j + c)^d \tag{4.13}$$

OC-SVM as it shows a better performance than other Kernel functions without having prior knowledge of the data [65], [66]. To obtain the decision function of the separable hyperplane between normal and untrained-abnormal attitude behavioural data of the satellite, a quadratic function was defined as [44]:

$$\begin{aligned} \min_{w \in F, \xi \in \mathbb{R}^N, \rho \in \mathbb{R}} \quad & \frac{1}{2}\|w\|^2 + \frac{1}{\nu N}\sum_{i=1}^{N} \xi_i - \rho \\ \text{Subject to} \quad & (w \cdot \Phi(x_i)) \geq \rho - \xi_i \\ & \xi_i \geq 0, \forall i = 1, 2, ..., N \end{aligned} \tag{4.14}$$

Where, w and ρ are the parameters of the hyperplane decision function and are computed after iterations in training, w is the coefficient of the decision function and ρ is the offset of the decision function. $\frac{1}{2}\|w\|^2$ is a regularizer term of the function [67]. ν is a trade-off regularization parameter with a range of (0,1) [50]. ξ is the slack variable in the training data.

To solve the quadratic problem above, a Lagrange function was then proposed to solve [44]:

$$L(w, \xi, \rho, \alpha, \beta) = \frac{1}{2}\|w\|^2 + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho - \sum_{i=1}^{N}\alpha_i((w \cdot \Phi(x_i)) - \ \rho + \xi_i) - \sum_{i=1}^{N}\beta_i\xi_i \tag{4.15}$$

Where, α_i and β_i are the positive multiplier parameters in the Lagrange function [61].

After taking partial derivatives of the Lagrange function with regard to w, ρ and ξ, both w and α_i then could be solved by setting the derivative functions to zero (shown in Eq. 4.16 and Eq. 4.17 [44]):

$$w = \sum_{i=1}^{N}\alpha_i\Phi(x_i) \tag{4.16}$$

$$\alpha_i = \frac{1}{\nu N} - \beta_i \leq \frac{1}{\nu N}\ , \sum_{i=1}^{N}\ \alpha_i = 1 \tag{4.17}$$

By substituting Eq. 4.16 and Eq. 4.17 into Eq. 4.15, the dual problem becomes [44]:

$$\begin{aligned} &\min_{\alpha}\ \frac{1}{2}\sum_{i=1,\ j=1}^{N}\alpha_i\alpha_j K(x_i,\ x_j) \\ &\text{Subject to}\ \ 0 < \alpha_i \leq \frac{1}{\nu N}, \sum_{i=1}^{N}\ \alpha_i = 1 \\ &\forall i = 1, \ldots, N,\ \forall j = 1, \ldots, N \end{aligned} \tag{4.18}$$

With $\alpha_i \epsilon (0, \frac{1}{\nu N}]$ and $\beta_i \neq 0$, the last parameter required for the decision function of the hyperplane, ρ, can be computed by combining Eq. 4.9 and Eq. 4.16 as [44]:

$$\rho = (w \cdot \Phi(x_i)) = \sum_{j=1}^{N}\alpha_j K(x_j,\ x_i) \tag{4.19}$$

Therefore, the general function of the separable hyperplane to determine the behavioral labels of the satellite with Kernel mapping functions was:

$$\begin{aligned} f(x) &= (w \cdot \Phi(x_i)) \geq \rho \\ f(x) &= sgn\left(\sum_{i=1}^{N} \alpha_i k(x_i, x) - \rho\right) \end{aligned} \tag{4.20}$$

Where, x_i in N number of training data is treated as a positive datapoint which is responsible for computing the decision function of the hyperplane for satellite attitude behaviour classification.

After learning the hyperplane decision function from all nominal training data, the detection system should be able to identify failures. Test data is fed into Eq. 4.20 with tuned parameters: α_i, w and ρ. Then, labels are assigned to each test datapoint. Label 0 is assigned to all normal data during testing (decision function indicates positive), and label 1 is assigned to all abnormal data (decision function indicates negative).

$$f(x_{test}) = sgn\left(\sum_{i=1}^{N} \alpha_i K(x_i, x_{test}) - \rho\right) \tag{4.21}$$

$$y = \begin{cases} 0 & f(x_{test}) \geq 0 \\ 1 & f(x_{test}) < 0 \end{cases} \tag{4.22}$$

Where, x_{test} is the test dataset. When $f(x_{test}) \geq 0$, it represents that the test datapoint falls inside of the defined decision hyperplane. When $f(x_{test}) < 0$ it represents that the test datapoint falls outside of the defined decision hyperplane. Therefore, if $f(x_{test})$ shows a positive value, the predicted label for the test datapoint will be 0 (no failure), otherwise the datapoint will be labelled as 1 (failure detected).

4.2.1.1 Sequential Minimal Optimization

To better define the decision function of the hyperplane, the Sequential Minimal Optimization (SMO) method was used in this research. The origin of the SMO method was proposed by Platt [68] to solve the quadratic problem in the general Support Vector Machine (SVM) method and to provide a faster and more efficient solution. An adaptive SMO algorithm was applied in this research to provide an efficient approach to solve the quadratic problem in the OC-SVM method. The adaptive SMO algorithm was modified by Schölkopf et al. and Platt ([44], [68]).

As mentioned in the previous section, α_i, w and ρ were the tuned parameters through the training process where w and ρ were dependent on α_i. Thus, the principal optimization problem for the OC-SVM method is to optimize α_i. By applying the SMO method, a pair of α(s) will be selected and optimized for each iteration. For example, the optimizing function for α_1 and α_2 are as follows:

$$\begin{aligned} \min_{\alpha_1,\alpha_2} \quad \frac{1}{2} \sum_{i=1,\ j=1}^{2} \alpha_i \alpha_j K_{ij} + \sum_{i=1}^{2} \alpha_i C_i + C \\ C_i = \sum_{j=3}^{N} \alpha_j K_{ij}, C = \sum_{i=3,\ j=3}^{N} \alpha_i \alpha_j K_{ij} \\ \text{Subject to } 0 \ \le \alpha_1 \le \frac{1}{\nu N},\ 0 \ \le \alpha_2 \le \frac{1}{\nu N} \end{aligned} \tag{4.23}$$

Where, α_1 and α_2 are the selected pair from α_i for optimization. From the linear equality constraint of α_1 and α_2 , the summation of α_1 and α_2 is the same before and after optimization ([68]–[70]). Therefore,

$$\alpha_1 + \alpha_2 = s \tag{4.24}$$

Where s represents the sum of the selected α_1 and α_2 before optimization. Then, the optimized α_2 could be updated as:

$$\alpha_{2_{new}} = \frac{s(K_{11} - K_{12}) + C_1 - C_2}{K_{11} + K_{22} - 2K_{12}} \tag{4.25}$$

From Eq. 4.24 the optimized α_1 could be attained:

$$\alpha_{1_{new}} = s - \alpha_{2_{new}} \tag{4.26}$$

For every step of optimization, the offset parameter of the decision function would require re-calculated and updated using Eq. 4.27.

$$\rho = \sum_{j=1}^{N} \alpha_j k(x_j,\ x_i) \tag{4.27}$$

The stopping criteria for SMO were the filter conditions for α optimization following the Karush-Kuhn-Tucker (KKT) conditions [71]. The first α selected for optimization was from the whole training dataset where any α that violated one of the KKT conditions would be chosen for SMO (shown in Eq. 4.28 and Eq. 4.29).

$$(f(x_i) - \rho) \cdot \alpha_i > 0 \tag{4.28}$$

or,

$$(\rho - f(x_i)) \cdot \left(\frac{1}{\nu N} - \alpha_i \right) > 0 \tag{4.29}$$

Where $f(x_i)$ is derived from Eq. 4.20.

$$f(x_i) = K_{1i}\alpha_1 +\ K_{2i}\alpha_2 + C_i \tag{4.30}$$

Where, C_i is denoted as $C_i = \sum_{j=3}^{N} \alpha_j K_{ij}$.

The second α, α_j, can be selected using the argument in Eq. 4.31:

$$j = arg \max_{n \in N} |f(x_i) - f(x_n)| \tag{4.31}$$

Once every α has satisfied the KKT conditions (listed in Eq. 4.28 and Eq. 4.29), the learning process is terminated. With the optimized output α, tuning parameters, w and ρ, were then could be computed by Eq. 4.16 and Eq. 4.17.

4.2.2 One-Class Linear Regression (OC-LR)

The second method I used to accomplish the one-sided learning basis anomaly detection system was the OC-LR method. Similar to the OC-SVM method, only normal behavioural data of the reaction wheel were used for training. As shown in Figure 4.1, there was an approximate functional relationship between reaction wheel current, angular velocity and angular acceleration that can describe the normal behaviour of the spacecraft. Therefore,

$$\boldsymbol{i} = f(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}) \tag{4.32}$$

Where, the reaction wheel current, $\boldsymbol{i}$, is a function of reaction wheel angular velocity, $\boldsymbol{\omega}$, and reaction wheel angular acceleration, $\dot{\boldsymbol{\omega}}$.

As the reaction wheel accelerates, the angular velocity of the reaction wheel increases, so does the current. Thus, a polynomial functional relationship for reaction wheel currents, angular velocity and angular acceleration was assumed. The relationship can be expressed as:

$$\boldsymbol{i} = a\boldsymbol{\omega} + b\boldsymbol{\omega}^2 + d\boldsymbol{\omega}^3 + e\dot{\boldsymbol{\omega}} + f\dot{\boldsymbol{\omega}}^2 + k\dot{\boldsymbol{\omega}}^3 + c \tag{4.33}$$

Where, a, b, c, d, e, f, k are the constant parameters of the function. In this method, the reaction wheel current, $\boldsymbol{i}$, the reaction wheel angular velocity, $\boldsymbol{\omega}$, the reaction wheel angular acceleration, $\dot{\boldsymbol{\omega}}$, still took account for training the system. The reaction wheel was the expected output of the function from training. The output reaction wheel current was then utilized as the failure-diagnosing measurement to detect failures. Since $\boldsymbol{i}$, $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ were collected in time series, the function can also be expressed as shown in Eq. 4.34:

$$\boldsymbol{i} = \boldsymbol{X}_{LR} \cdot \boldsymbol{A} \tag{4.34}$$

Where,

$$\boldsymbol{X}_{LR} = \begin{bmatrix} \boldsymbol{\omega} & \boldsymbol{\omega}^2 & \boldsymbol{\omega}^3 & \dot{\boldsymbol{\omega}} & \dot{\boldsymbol{\omega}}^2 & \dot{\boldsymbol{\omega}}^3 & \boldsymbol{1} \end{bmatrix} = \begin{bmatrix} \omega_1 & \omega_1^2 & \omega_1^3 & \dot{\omega}_1 & \dot{\omega}_1^2 & \dot{\omega}_1^3 & 1 \\ \omega_2 & \omega_2^2 & \omega_2^3 & \dot{\omega}_2 & \dot{\omega}_2^2 & \dot{\omega}_2^3 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_N & \omega_N^2 & \omega_N^3 & \dot{\omega}_N & \dot{\omega}_N^2 & \dot{\omega}_N^3 & 1 \end{bmatrix} \tag{4.35}$$

N is the total number of the training data. $\boldsymbol{X}_{LR}$ is a $N \times 7$ matrix which contains all training data of $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ for the OC-LR method in time series (present in Eq. 4.35). $\boldsymbol{A}$ is the matrix that comprises of all function parameters (a, b, c, d, e, f, k). Therefore, Eq. 4.34 becomes:

$$\begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_N \end{bmatrix} = \begin{bmatrix} \omega_1 & \omega_1^2 & \omega_1^3 & \dot{\omega}_1 & \dot{\omega}_1^2 & \dot{\omega}_1^3 & 1 \\ \omega_2 & \omega_2^2 & \omega_2^3 & \dot{\omega}_2 & \dot{\omega}_2^2 & \dot{\omega}_2^3 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_N & \omega_N^2 & \omega_N^3 & \dot{\omega}_N & \dot{\omega}_N^2 & \dot{\omega}_N^3 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ d \\ e \\ f \\ k \\ c \end{bmatrix} \tag{4.36}$$

To solve for $\boldsymbol{A}$, the problem became to solve a system of linear equations. The inverse of a matrix was used:

$$\boldsymbol{A} = \boldsymbol{X}_{LR}^{-1} \cdot \boldsymbol{i} = \begin{bmatrix} \omega_1 & \omega_1^2 & \omega_1^3 & \dot{\omega}_1 & \dot{\omega}_1^2 & \dot{\omega}_1^3 & 1 \\ \omega_2 & \omega_2^2 & \omega_2^3 & \dot{\omega}_2 & \dot{\omega}_2^2 & \dot{\omega}_2^3 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_N & \omega_N^2 & \omega_N^3 & \dot{\omega}_N & \dot{\omega}_N^2 & \dot{\omega}_N^3 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_N \end{bmatrix} \tag{4.37}$$

As $\boldsymbol{X}_{LR}$ is not a square matrix $(n \times n)$, it was not invertible. Hence, in Eq. 4.38, Moore-Penrose Pseudo Inverse was used instead of using inverse matrix of $\boldsymbol{X}_{LR}$ [72].

$$\boldsymbol{X}_{LR}\dagger = (\boldsymbol{X}_{LR}^T \cdot \boldsymbol{X}_{LR})^{-1} \boldsymbol{X}_{LR}^T \tag{4.38}$$

Where, $\boldsymbol{X}_{LR}\dagger$ is the pseudo inverse matrix of $\boldsymbol{X}_{LR}$. $\boldsymbol{X}_{LR}^T$ is the transpose matrix of $\boldsymbol{X}_{LR}$.

Then, by using the pseudo interse matrix of $\boldsymbol{X}_{LR}$, the parameters of the system of linear equations were calculated as:

$$\boldsymbol{A} = \boldsymbol{X}_{LR} \dagger \cdot \boldsymbol{i} \tag{4.39}$$

From Eq. 4.39, the expected reaction wheel current can be predicted through the computed parameters, $\boldsymbol{A}$.

$$\boldsymbol{i}_{predict} = \boldsymbol{X}_{LR} \cdot \boldsymbol{A} \tag{4.40}$$

Where, $\boldsymbol{i}_{predict}$ is the predicted reaction wheel current computed by using the Moore-Penrose Pseudo Inverse of $\boldsymbol{X}_{LR}$.

The difference between the predicted and measured reaction wheel current from the training data was then calculated and used as the disgnosing metric for the following given tests to determine if there was a failure. The current difference was calculated as:

$$\boldsymbol{i}_{diff} = \boldsymbol{i}_{predict} - \boldsymbol{i} \tag{4.41}$$

Where, $\boldsymbol{i}_{diff}$ is the difference between the predicted current and the measured current of the reaction wheel.

After I calculated the current difference, the Empirical Rule (68-95-99.7 Rule) was selected as the failure detection criteria. By using the Empirical Rule, it was assumed that the collected training data followed a normal distribution. The Empirical Rule implies that in a normal-distributed data, 68.27% of the data falls within 1 standard deviation of the data mean, 95.45% of the data falls within 2 standard deviations of the data mean, and 99.7% of the data falls within 3 standard deviations of the data mean. Figure 4.2 shows an example of the 68-95-99.7 rule application on a normal distributed data population.

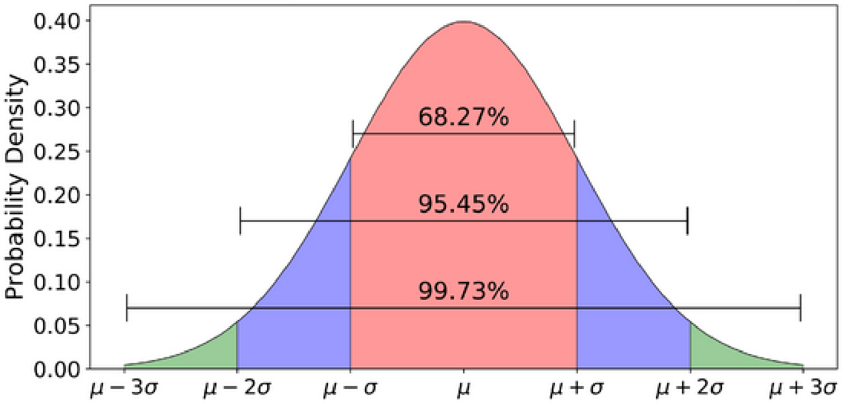

Figure 4.2: Empirical Rule (68-95-99.7) Plot [73]

By applying the 68-95-99.7 rule to the calculated current differences, as the whole training dataset was from normal behaviour of the spacecraft, it indicated that:

- Within 1 standard deviation of the data mean, there is a probablity that 68.27% of the data is normal.
- Within 2 standard deviations of the data mean, there is a probablity that 95.45% of the data is normal.
- Within 3 standard deviations of the data mean, there is a probablity that 99.7% of the data is normal.

Then, the failure detection criteria were calculated from the standard deviations of the data mean based on the 68-95-99.7 rule. Three detection criteria were: 1 standard deviation of the current difference mean, 2 standard deviations the current difference mean, and 3 standard deviations of the current difference mean. Where,

$$\begin{aligned} CI_{limit_1} &= \mu_{diff} \pm 1\sigma_{diff} \\ CI_{limit_2} &= \mu_{diff} \pm 2\sigma_{diff} \\ CI_{limit_3} &= \mu_{diff} \pm 3\sigma_{diff} \end{aligned} \tag{4.42}$$

μ_{diff} is the current difference mean, σ_{diff} is the standard deviation of the current differnce. CI_{limit_1} is the limit of $1\sigma_{diff}$ of the current difference mean, CI_{limit_2} is the limit of $2\sigma_{diff}$ of the current difference mean, CI_{limit_3} is the third limit of $3\sigma_{diff}$ of the current difference mean. μ_{diff} and σ_{diff} were calculated from the current differences from traning data (shown in Eq. 4.41), where

$$\mu_{diff} = \frac{\sum i_{diff}}{N} \tag{4.43}$$

i_{diff} is the calculated current difference at one timepoint from training data.

After calculating the current difference mean, the current difference standard deviation was determined:

$$\sigma_{diff} = \sqrt{\frac{\sum |i_{diff} - \mu_{diff}|^2}{N}} \tag{4.44}$$

The probability of the data that fell into the normal behavioural data range was:

$$\begin{aligned} P_{limit_1}(\mu_{diff} - 1\sigma_{diff} \leq x \leq \mu_{diff} + 1\sigma_{diff}) \approx 68.27\% \\ P_{limit_2}(\mu_{diff} - 2\sigma_{diff} \leq x \leq \mu_{diff} + 2\sigma_{diff}) \approx 95.45\% \\ P_{limit_3}(\mu_{diff} - 3\sigma_{diff} \leq x \leq \mu_{diff} + 3\sigma_{diff}) \approx 99.73\% \end{aligned} \tag{4.45}$$

Where, P_{limit_1} is the probability of 68.27% that data falls into the range of $\mu_{diff} \pm 1\sigma_{diff}$, P_{limit_2} is the probability of 95.45% that data falls into the range of $\mu_{diff} \pm 2\sigma_{diff}$, P_{limit_3} is the probability of 99.73% data falls into the range of $\mu_{diff} \pm 3\sigma_{diff}$.

After calculating the limits through the training data, the detection system was capable of diagnosing failures from the test data. With the limit of 1σ of the data mean, labels were assigned as:

$$y = \begin{cases} 0 & \mu_{diff} - 1\sigma_{diff} \leq x_{test} \leq \mu_{diff} + 1\sigma_{diff} \\ 1 & x_{test} < \mu_{diff} - 1\sigma_{diff} \ and \ x_{test} > \mu_{diff} + 1\sigma_{diff} \end{cases} \tag{4.46}$$

With the limit of 2σ of the data mean, labels were assigned as:

$$y = \begin{cases} 0 & \mu_{diff} - 2\sigma_{diff} \leq x_{test} \leq \mu_{diff} + 2\sigma_{diff} \\ 1 & x_{test} < \mu_{diff} - 2\sigma_{diff} \ and \ x_{test} > \mu_{diff} + 2\sigma_{diff} \end{cases} \tag{4.47}$$

With the limit of 3σ of the data mean, labels were assigned as:

$$y = \begin{cases} 0 & \mu_{diff} - 3\sigma_{diff} \leq x_{test} \leq \mu_{diff} + 3\sigma_{diff} \\ 1 & x_{test} < \mu_{diff} - 3\sigma_{diff} \ and \ x_{test} > \mu_{diff} + 3\sigma_{diff} \end{cases} \tag{4.48}$$

Where, label 0 was assigned to all normal test data (within the limits), and label 1 was assigned to all failure test data (outside the limits).

Chapter 5

Simulated Tests Results and Discussion

In this chapter, training data, training process and simulated anomaly-detection tests for the two-sided learning and one-sided learning anomaly detection systems are presented. The details of how the training data were created and how the training process were undertaken will be addressed. The simulated anomaly-detection-test datasets will be presented. Two types of the simulated tests were performed; test results for each anomaly detection system will be inlcluded. Test performance for each method will also be analyzed and discussed in this chapter.

5.1 Two-Sided Logistic Regression[7]

5.1.1 Training Dataset for Two-Sided Logistic Regression

As mentioned in the previous chapter, the Logistic Regression method was adopted to accomplish the two-sided learning basis anomaly detection system. To begin with training this system, I used the simulated spacecraft attitude dynamics control system model developed in Chapter 3, with variable viscous bearing friction in one of the reaction wheels.

Using the model, I created a dataset of 100 different slew trajectories with random slew rates and slew periods over a time-span of 10,000 seconds as the training dataset used to train the two-sided learning basis detection system. The simulated slews in time series are shown in Figure 5.1 which presents the desired slew rates of the spacecraft in time series. Figure 5.2 presents the measured slew rates of the reaction from gyroscopes with a random noise of 0 rad/s mean and a variance of 2.5 $\times 10^{-6}$ rad/s.

To reduce and control the impacts from variable factors, all slews had the same slew axes and acceleration rates. The applied slew axis was $[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]$ and the slew acceleration rate was 0.001 rad/s^2 for each of the simulated slew profile.

[7] This section is a modified version based on the paper [1], [2] published in AIAA SciTech 2021 Forum and 35th Small 35th Annual Small Satellite Conference by Yujia Huang and Philip Ferguson, copyright © 2021 by Yujia Huang, reproduced with permission.

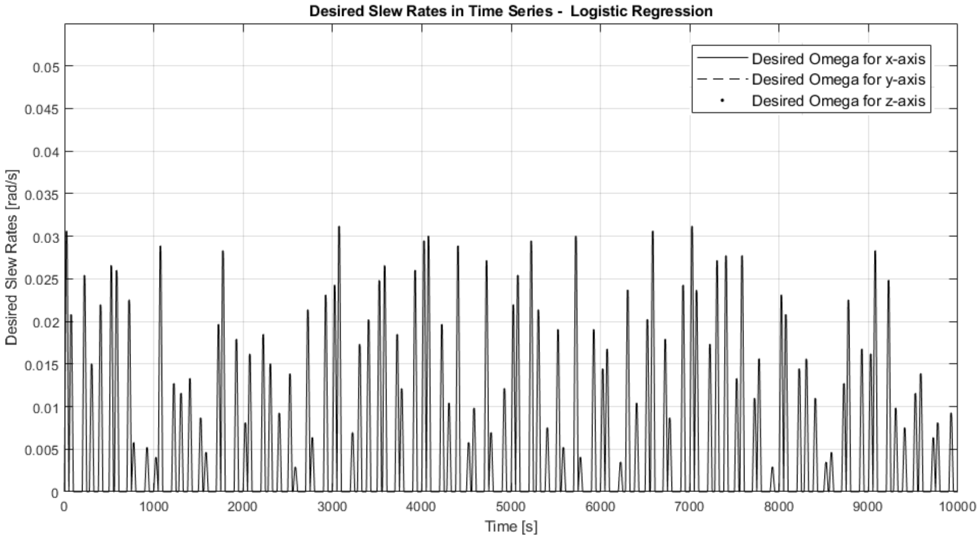

Figure 5.1: Desired Slew Rates in Time Series for Logistic Regression

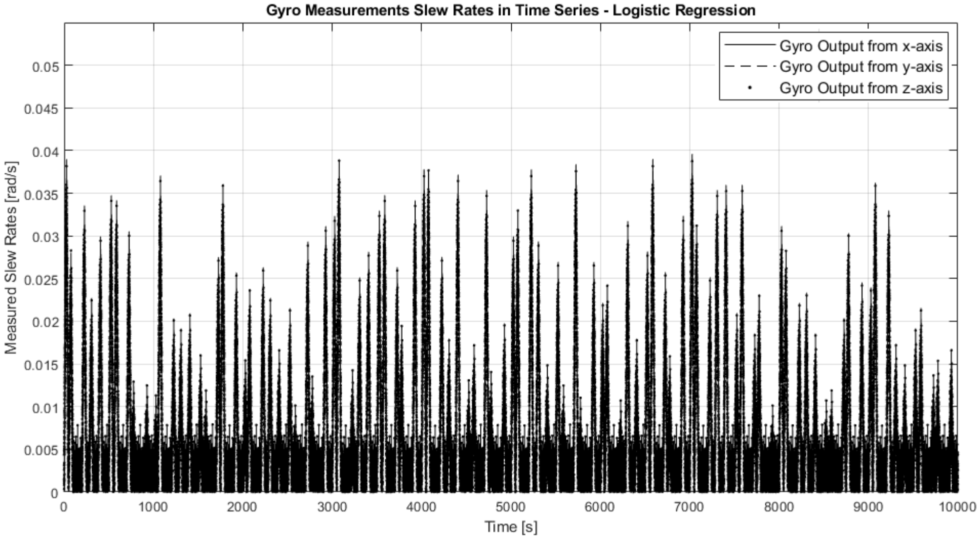

Figure 5.2: Gyro Measurements Slew Rates in Time Series for Logistic Regression

The training data included both normal and abnormal data. The first 5000-second set of training data was obtained from the healthy trajectories of the spacecraft where the nominal viscous friction of reaction wheel was 1×10^{-5} Nm/(rad/s). The next 5000-second set of data was added after the first set of training data where viscous friction failures were incorporated. During simulation, degradation in the reaction wheel bearing was selected to be the research object. When the reaction wheel bearing degrades, the lubrication condition in the reaction wheel reduces. Therefore, the simulated failures into the system were created by increasing the viscous friction where the reaction wheel friction would grow with bearing wear.

The failure model was simulated with changes of viscous friction in random constant values (see in Eq. 3.21), starting at the 5000^{th} second of the total slews and changing every 10 seconds. Each 10 second period contained a different abnormal value of viscous friction (greater than the nominal value, m was randomly selected and greater than 1), with nominal friction periods between the abnormal periods. An example of how the friction failures were generated is shown in Figure 5.3. In this plot, it shows an example of how the friction failures were generated in 1000 seconds. Data points located at 1×10^{-5} Nm/(rad/s) were the nominal viscous friction dedicated to a normal reaction wheel. Other data points greater than 1×10^{-5} Nm/(rad/s) were when the reaction wheel bearing degradation failures happened.

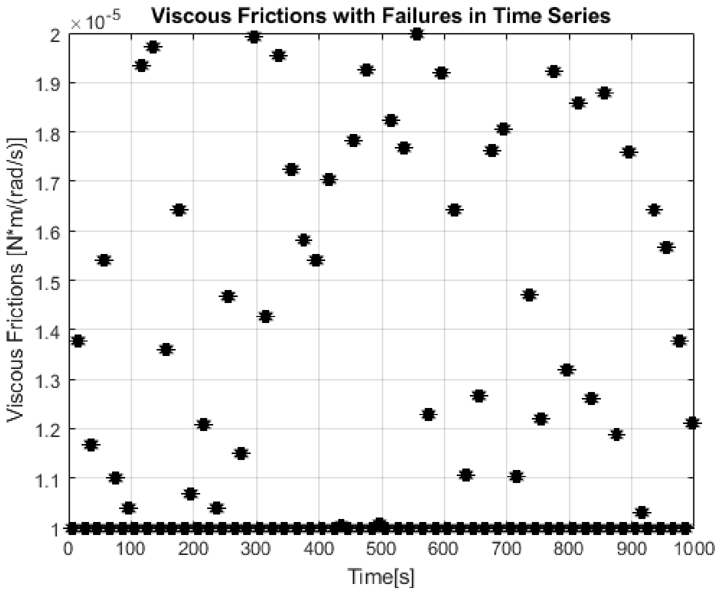

Figure 5.3: Viscous Frictions with Failures in 1000 Seconds

To analyze the behaviour of the spacecraft, reaction wheel current, reaction wheel angular velocity, and reaction wheel angular acceleration data were considered as the most efficient training features from all available outputs from the simulated model. The current, angular velocity, and angular acceleration were all collected from the model with a time step of 10 Hz. Therefore, there was a total of 100,0000 datapoints collected for each training feature.

Figure 5.4, Figure 5.5, and Figure 5.6 show the collected reaction wheel current, angular velocity, and angular accleration from simulated slews failures in time series. The viscous friction failures started at 5000^{th} second and were applied for every 10 seconds. It is important to note that some failures occurred during the coast phase of the slews, while others occurred during the acceleration phase. This training set was designed to provide a rich enough data set such that the Logistic Regression method could distinguish between abnormal viscous friction and a normal wheel acceleration (since the reaction wheel current naturally increases during normal operations).

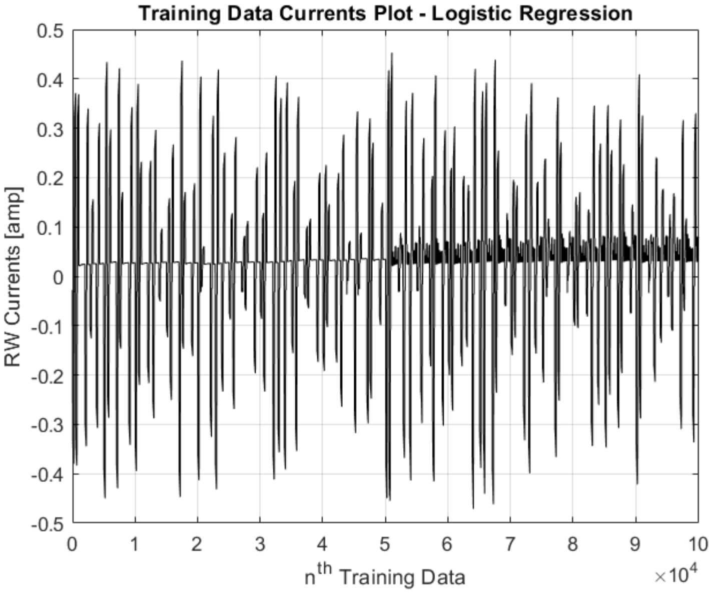

Figure 5.4: Collected Reaction Wheel Currents in 10,000 s

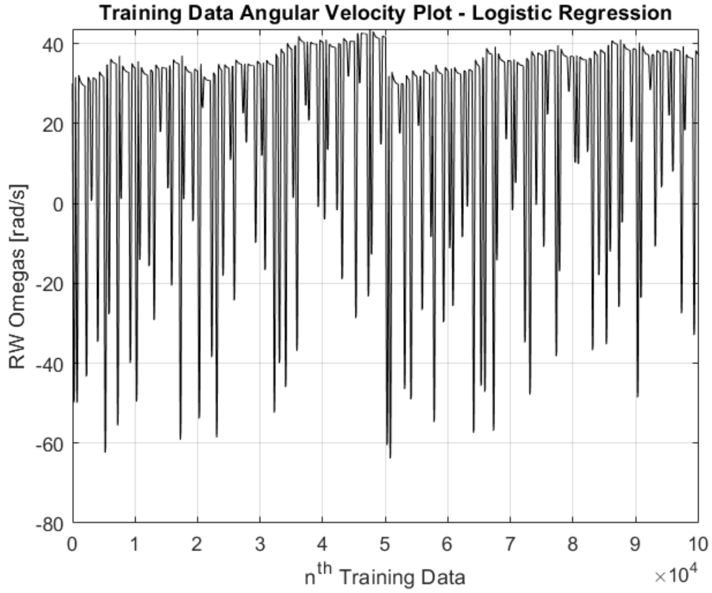

Figure 5.5: Collected Reaction Wheel Angular Velocity in 10,000 s

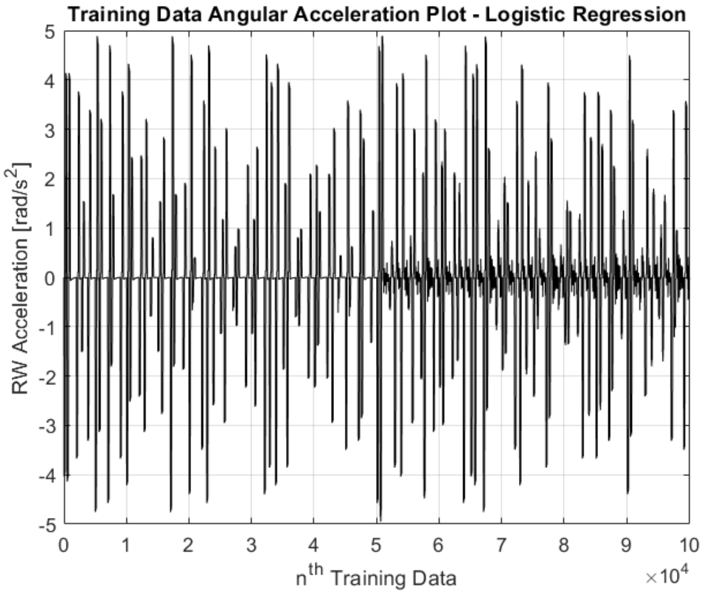

Figure 5.6: Collected Reaction Wheel Angular Acceleration in 10,000 s

Then, the collected data from the simulated models were prepared and processed for learning. The behaviours for the collected data were labelled as 0 or 1 before training the system to distinguish the normal data and the abnormal data. The data with no failure was labelled as "0", and the data with failure was labelled as "1". The normal and abnormal training datasets were stored as one dataset. For every timepoint, each training task involved three data points (in reaction wheel current, angular velocity and angular acceleration) for a single corresponding label. The training process is shown in Figure 5.7.

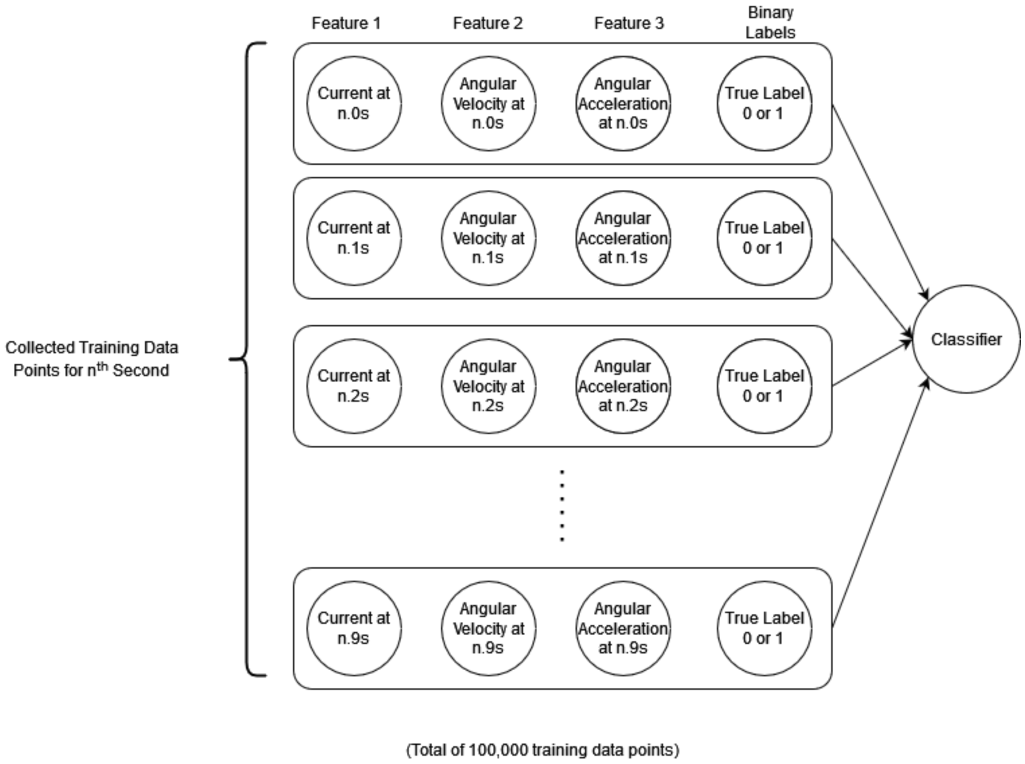

Figure 5.7: Training Process of Logistic Regression

The consecutive data was required to provide enough information for the training process. With a total of 100,000 training data with 3 features and 100,000 corresponding behaviourial labels, the classifier was trained to learn the logistical relationship of the decision boundary between the data with no failure and with failure. The input training data in a 3D plot can be seen in Figure 5.8. It was observed that decision boundary between the normal and failure data formed an inclined plane. During the learning process, the parameters for the decision boundary plane could be obtained where $\boldsymbol{W}$ = [-0.1145, -9.4851, -1.4441, 110.4964]. With the most weighted parameter from the computed parameters of the decision boundary, the reaction wheel angular velocity took the most important role through the three training features. The cost fuction with respect to the number of learning iteration is shown in Figure 5.9.

With the number of iteration increases, the cost of the training process decreased and gradually converged.

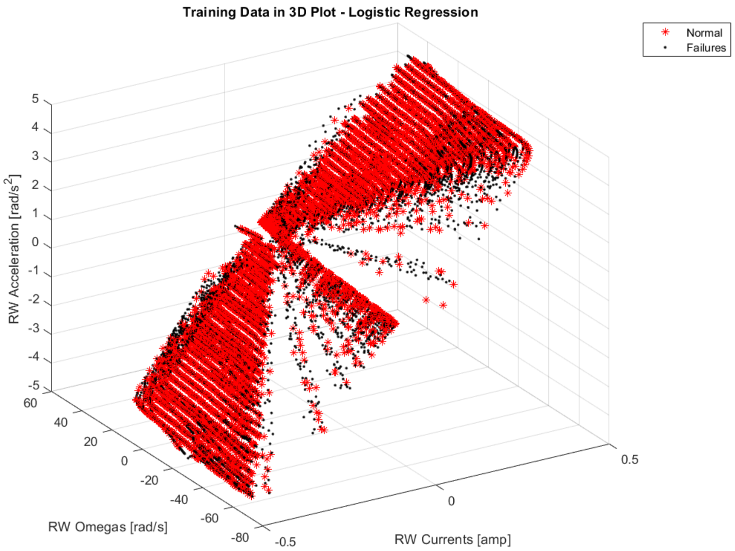

Figure 5.8: Training Data in 3D Plot for Logistic Regression

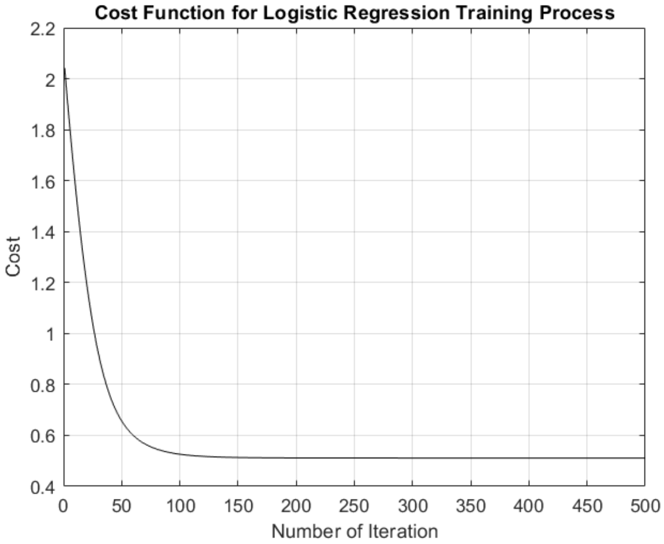

Figure 5.9: Cost Function for Logistic Regression during Training

To examine the trained classifer for the Logistic Regression method, 75% of the total collected data was used for training, the remaining 25% of the data was used for cross validation. The detection accuracy of the validation dataset was 87.65%. The receiver operating characteristic curve (ROC) with varied examine thresholds is plotted in Figure 5.10 where the area under the ROC curve (AUC) was 83.28%. The detection accuracy and the AUC value from the validation dataset indicated a good performance of the trained classifer by the Logistic Regression method. The random guess line was labelled in the ROC plot (shown in Figure 5.11), where the AUC under the random guess line is 0.5. The random guess line provides a reference criterion to assess a trained classifer which represents the detection ability of the classifier to distinguish between normal and abormal datapoint are equal[74]. If the detection results are above the random guess line, it indicates that the True Positive Rate (TPR) for the test is greater than False Positive Rate (FPR) and the detection results are acceptable [74]. If the detection results are below the random guess line, it indicates that the FPR is greater than TPR where the trained classifier has low ability to distinguish behaviours. An Equal Error Rate (EER) line was also labelled in the plot where the EER point was found as the interception point between the ROC curve and the EER line. The EER point showed the best threshold could be chosen for the trained classifier where the TPR (True Postive Rate) was around 0.8 with the FPR (False Positive Rate) of 0.22. It can be observed that the trained classifier by using Logistic Regression method had a low ability to differentiate the abnormal behaviour for the validation dataset before reaching the EER point.

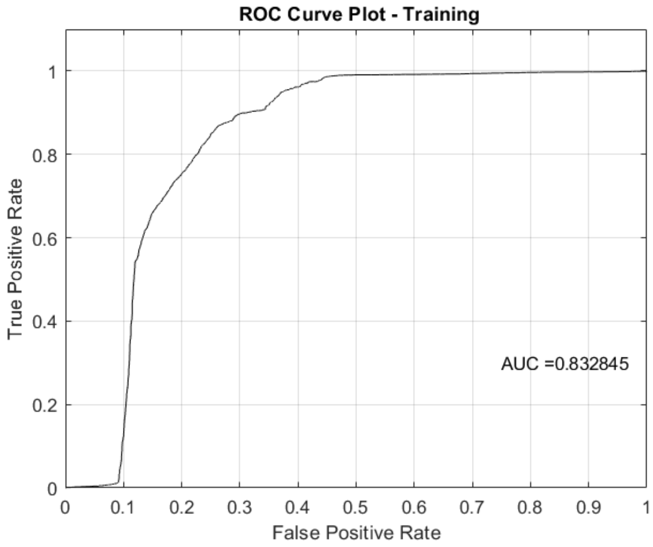

Figure 5.10: Receiver Operating Characteristic Curve of the Cross Validation Dataset of Logistic Regression

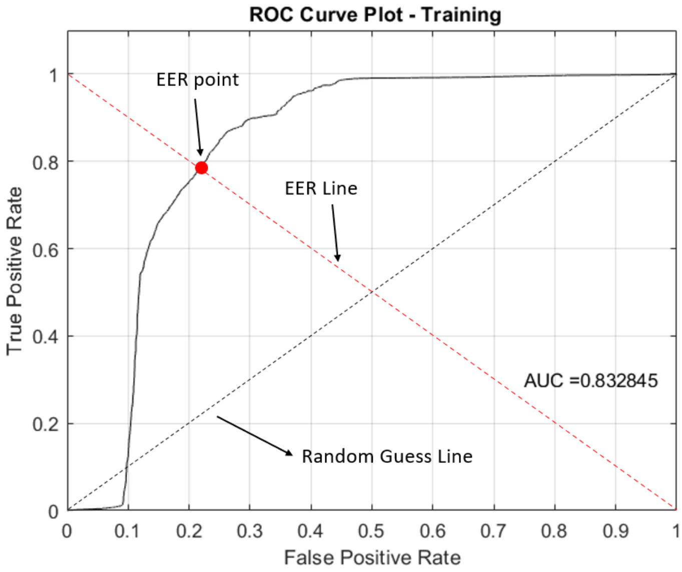

Figure 5.11: Receiver Operating Characteristic Curve of the Cross Validation Dataset of Logistic Regression with Equal Error Rate Point

5.1.2 Simulation Tests for Logistic Regression

After the training and cross validation stage, two tests were given to the system for failure detection. The first test was to examine if the designed detection system is capable of detecting slow changes in friction parameters over a long time period. The second test was to examine if the detection system is capable of detecting intermittent friction failures during slews.

In the first test, ten different random slews were developed for testing. The friction failure started at the midpoint of the slew (at the 500^{th} second). The slew axis and acceleration rate were identical with the slew axis and acceleration simulated in the training dataset. The slew axis was $[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]$ and the slew acceleration rate was 0.001 rad/s^2. At the 500th second, the friction failure occurred and began to increase gradually with respect to time in a rate of 0.02. Then,

$$f_{visous_{fail_{test1}}} = f_{viscous_{nominal}}(1 + 0.02t) \tag{5.1}$$

Where, $f_{visous_{fail_{test1}}}$ is the friction failure added in test one dataset. t is the time when the failure began, starting from the 500^{th} second to the 1000^{th} second in the first test. Figure 5.12 shows how the viscous friction changed when creating the first test dataset.

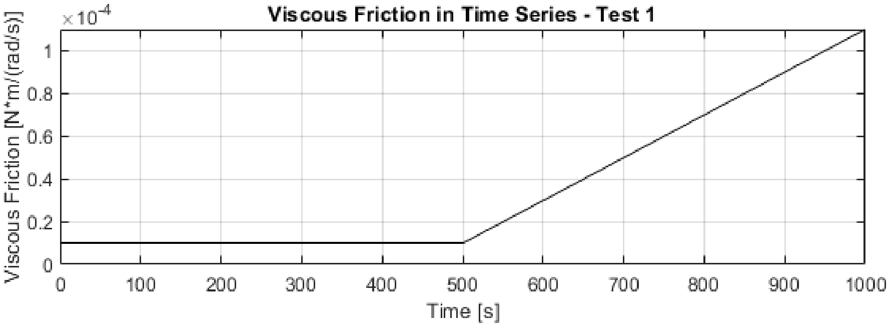

Figure 5.12: Viscous Friction in Test 1

The corresponding reaction wheel current, angular velocity and angular acceleration are shown in Figure 5.13, Figure 5.14 and Figure 5.15. When there was more friction in the reaction wheel, more current was taken by the reaction wheel. Since a PD controller was applied in the reaction wheel subsystem, there was no obvious increase or change in the behaviour of the reaction wheel angular velocity and acceleration.

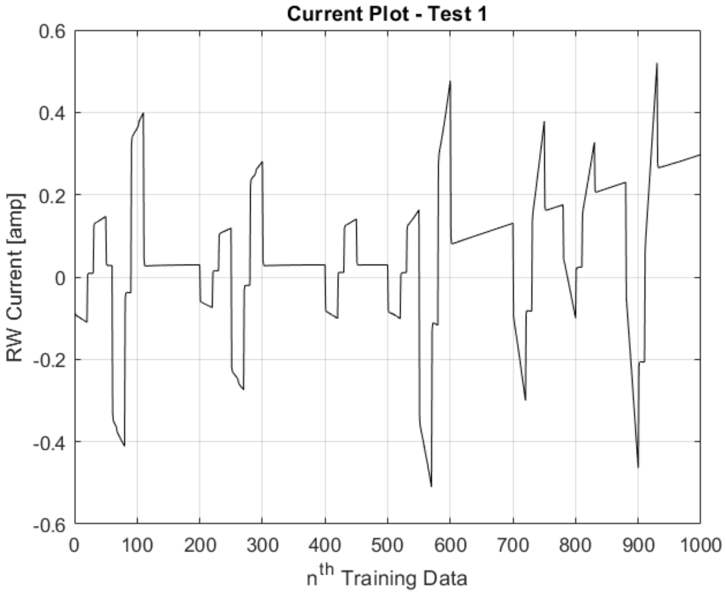

Figure 5.13: Reaction Wheel Current Data Plot for Test 1

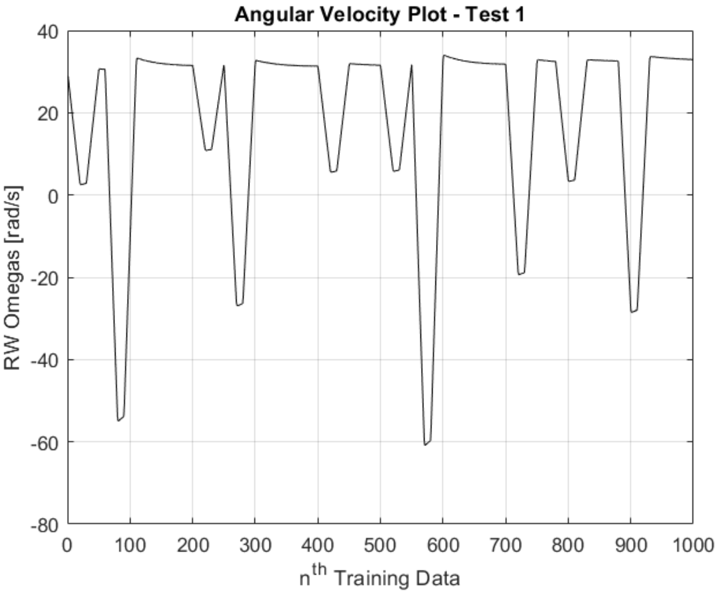

Figure 5.14: Reaction Wheel Angular Velocity Data Plot for Test 1

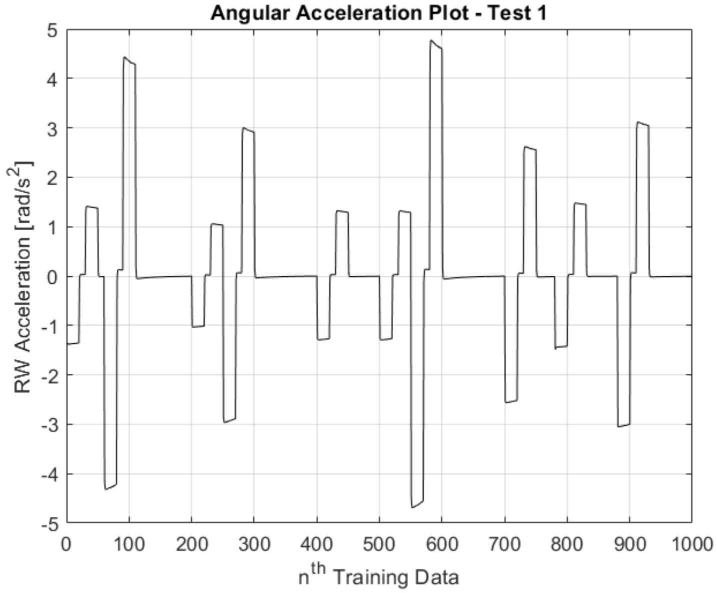

Figure 5.15: Reaction Wheel Angular Acceleration Data Plot for Test 1

The predicted results of test one for real-time spacecraft behaviour are presented in Figure 5.16. The normal and abnormal behaviour of the spacecraft were represented as “0” and “1” respectively.

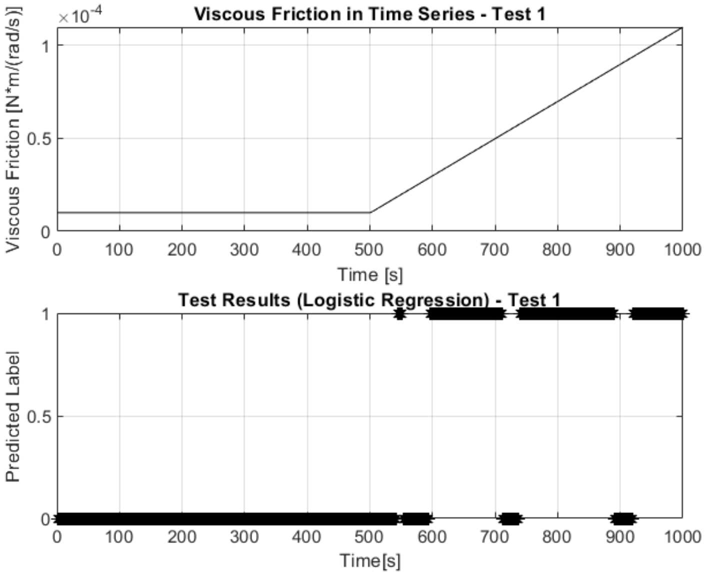

Figure 5.16: Prediction Results of Test 1 for Logistic Regression

By observing the predicted test results in Figure 5.16, the false label predictions were mainly taken place at around 575^{th} second, 705^{th} second, and 900^{th} second. After comparing these timepoints with the reaction wheel angular velocity plot and the angular acceleration plot in Figure 5.17 and Figure 5.19, it was found that the prediction errors all happened when the reaction wheel was accelerating or deaccelerating. As the viscous friction failure was increasing with time at a small rate (0.02) at the same time, it was reasonable for the detection system to determine the reaction were having a normal accelerating/deaccelerating motion. Comparatively, the detection system showed a better performance when the reaction wheel was coasting.

Figure 5.17 shows the plot of ROC curve for test one as well as the AUC. The detection accuracy for test one was 85.9% with an AUC of 0.7981. The random guess line and the EER line are presented in Figure 5.18. The best threshold for this test could be selected from the EER point where the TPR was about 0.78 with the FPR of 0.22.

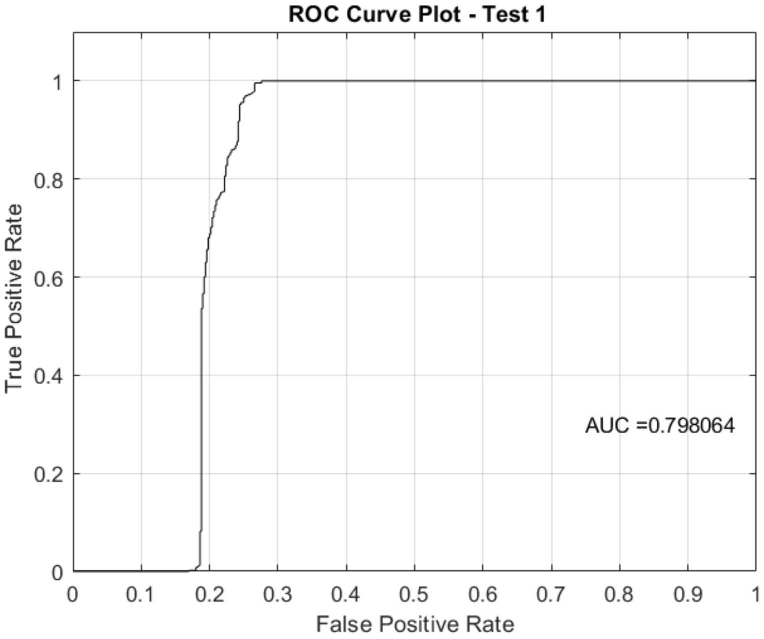

Figure 5.17: Receiver Operating Characteristic Curve of Test 1 for Logistic Regression

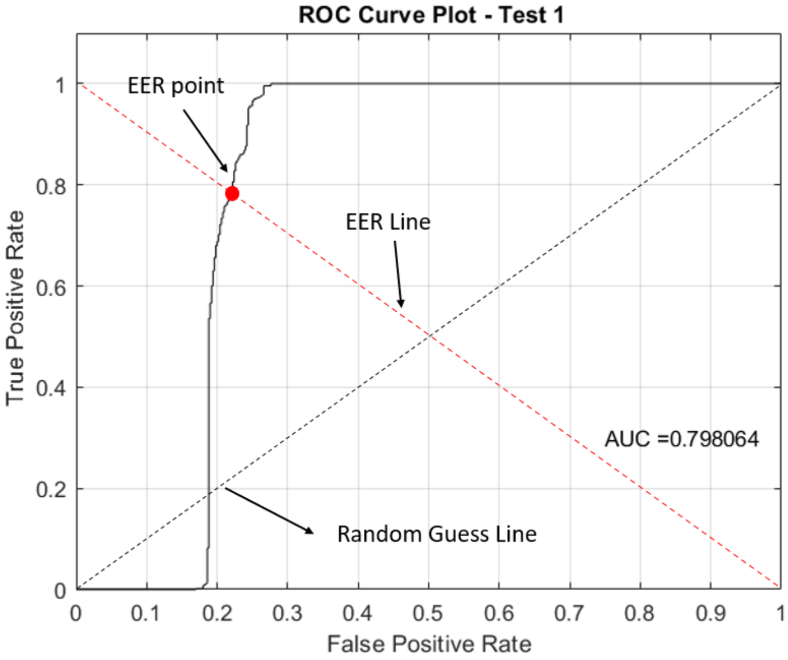

Figure 5.18: Receiver Operating Characteristic Curve of Test 1 for Logistic Regression with Equal Error Rate Point

In the second test, another set of ten random slews was used for testing. In this test, I implemented trapezoidal-like viscous friction profiles. The magnitude of the friction failure from each phase was increased with time in an increasing rate of 0.03 rad/s^2, then remained at a constant level for a short time (20-30s). Then, the friction failure decreased back to the nominal value of the viscous friction to 1 $\times 10^{-5}$ Nm/(rad/s) with a decreasing rate of -0.03 rad/s^2. The change of the viscous friction profile can be expressed as:

$$f_{visous_{fail_{test2}}} = \begin{cases} f_{viscous_{nominal}}(1 + 0.03t) & t_{fail_{start}} \leq t \leq t_{coast} \\ f_{viscous_{nominal}}(1 + 0.03t_{coast}) & t = t_{coast} \\ f_{viscous_{nominal}}(1 + 0.03t_{coast} - 0.03t) & t_{coast} \leq t \leq t_{fail_{end}} \end{cases} \tag{5.2}$$

Where, $f_{visous_{fail_{test2}}}$ is the friction failure added in test two dataset. $t_{fail_{start}}$ is the initial time when the friction failure started, t_{coast} is the time when the friction was expected to be maintained at a constant level, and $t_{fail_{end}}$ is the end time when the friction failure dropped back to the nominal viscous friction. Figure 5.19 shows the two viscous failure profiles in test two.

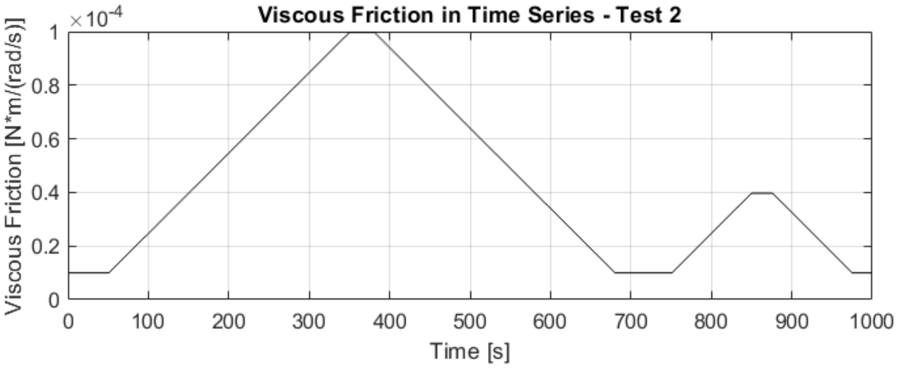

Figure 5.19: Viscous Friction in Test 2

Figure 5.20, Figure 5.21, and Figure 5.22 show the datasets of reaction wheel current, angular velocity and angular acceleration corresponding to the change of the viscous friction for test two.

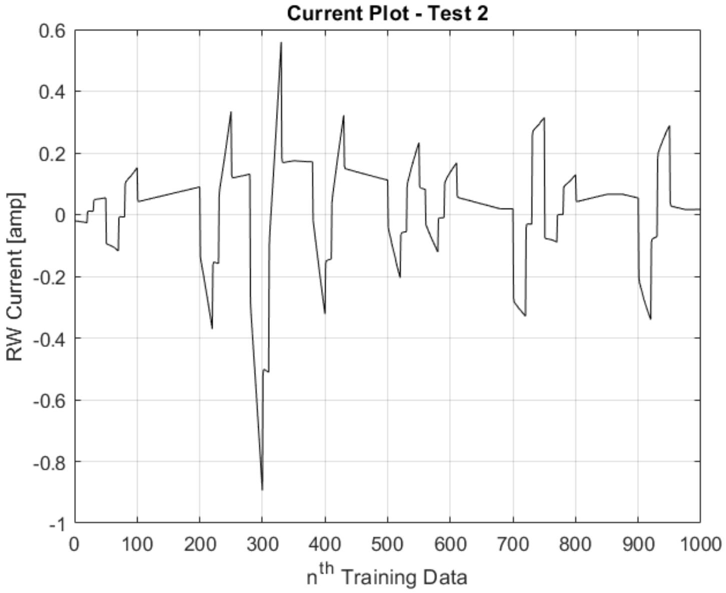

Figure 5.20: Reaction Wheel Current Data Plot for Test 2

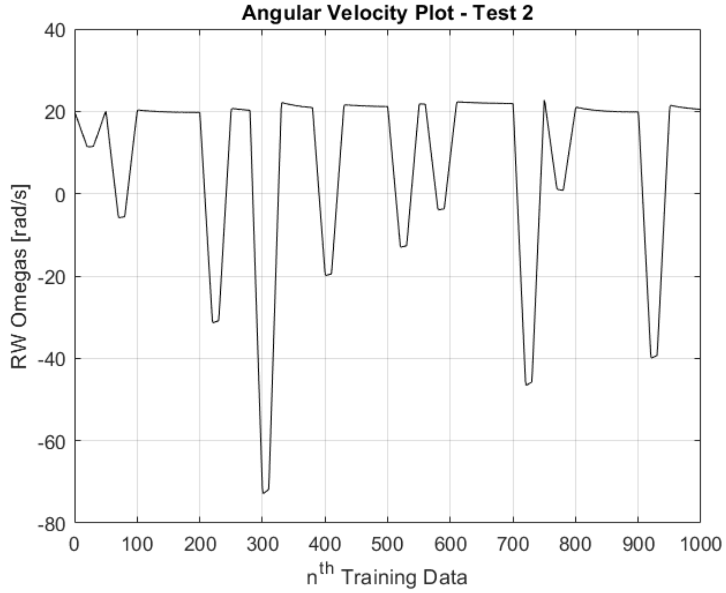

Figure 5.21: Reaction Wheel Angular Velocity Data Plot for Test 2

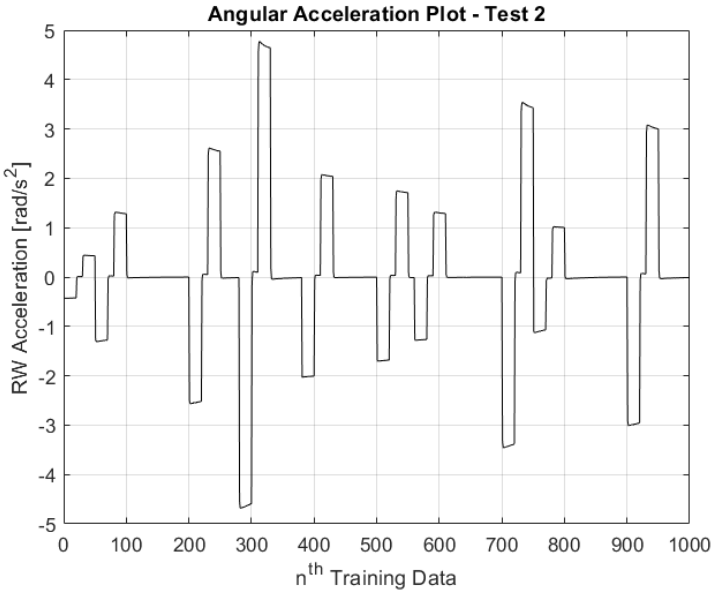

Figure 5.22: Reaction Wheel Angular Acceleration Data Plot for Test 2

Figure 5.23 below shows the testing results from test two with the detection accuracy of 62.5%. Figure 5.24 shows the ROC curve plot and the AUC of 0.7219 and Figure 5.25 shows the random guess line and the EER line for the ROC plot for the test two. The false predictions in test two were shown at around 205^{th} second, 300^{th} second, 400^{th} second, 510^{th} second and 600^{th} second. Similar to the test one performance, these false behavioural predictions were also taken place during acceleration and deacceleration motions of the reaction wheel. The best threshold selected for the trained classification model in this test was selected when the TPR was about 0.71 and the FPR was about 0.28.

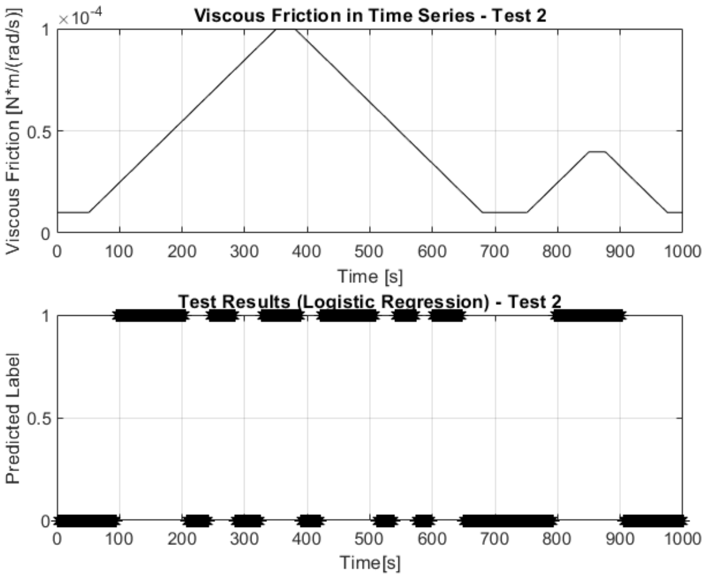

Figure 5.23: Prediction Results of Test 2 for Logistic Regression

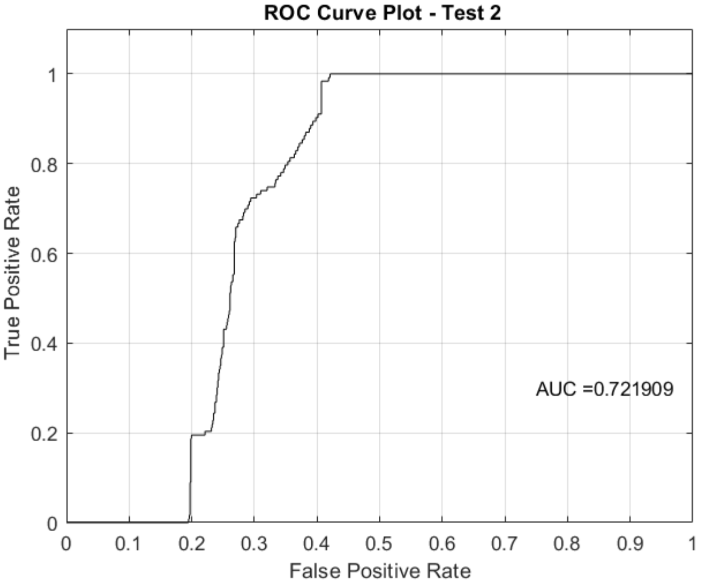

Figure 5.24: Receiver Operatig Characteristic Curve of Test 2 for Logistic Regression

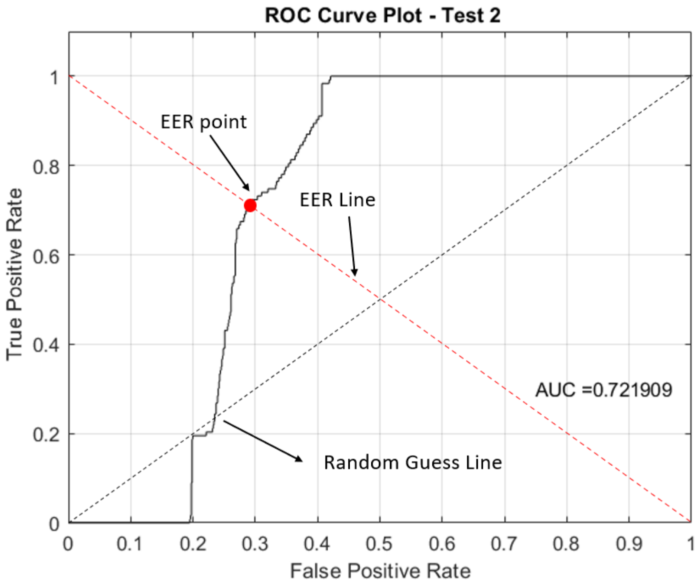

Figure 5.25: Receiver Operatig Characteristic Curve of Test 2 for Logistic Regression with Equal Error Rate Point

With the detection results of the spacecraft's behaviour (Figure 5.16 and Figure 5.24), the two-sided learning detection system showed good performance for detecting long-term visous friction failures during the simulated anomaly testing, but showed poor performance when detecting intermittent friction failures. Moreover, the false anomaly detections mainly appeared during the acceleration and deacceleration phases of the reaction wheel behavioural profiles. By comparing the ROC curves for each test with the random guess line, the classifier trained by the Logistic Regression method showed a relatively poor ability to distinguish the normal and abnormal cases for the simulation tests before reaching the ERR point.

5.2 Two-Sided OC-SVM and OC-LR[9]

5.2.1 Training Dataset for OC-SVM and OC-LR

All the training data for the OC-SVM and the OC-LR methods was from healthy spacecraft slews. To make the proposed two-sided and one-sided anomaly detection systems comparable with each other, the training dataset for the one-sided learning anomaly detection system was taken from the first 5000 second data from Logistic Regression training dataset, where the first 5000-second data was deemed to be normal data with the viscous friction of 1 $\times 10^{-5}$ Nm/(rad/s).

Figure 5.26, Figure 5.27, and Figure 5.28 present the datasets of the reaction wheel current, angular velocity and angular accelration to train the one-sided learning system using OC-SVM and OC-LR. The datasets were identical to the first 50,000 datapoints of the two-sided Logistical Regression training datasets and were all normal.

[9] This section is a modified version based on the paper [1], [2] published in AIAA SciTech 2021 Forum and 35th Small 35th Annual Small Satellite Conference by Yujia Huang and Philip Ferguson, copyright © 2021 by Yujia Huang, reproduced with permission.

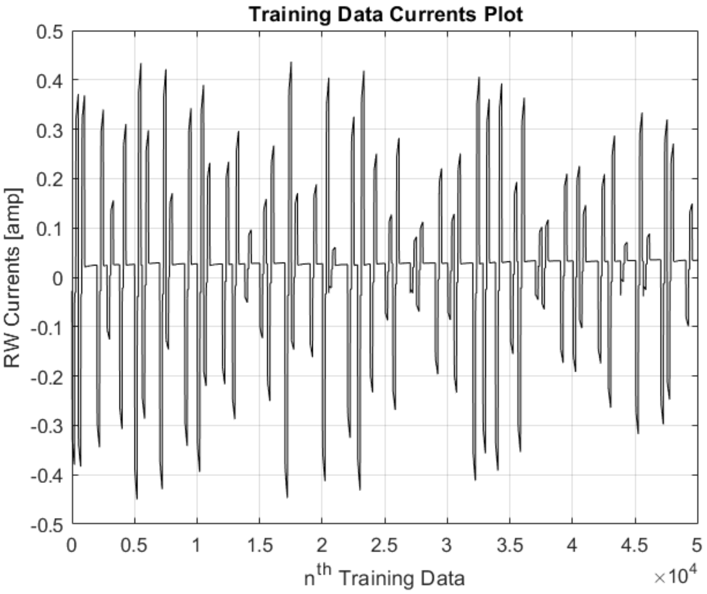

Figure 5.26: Reaction Wheel Currents Training Dataset for OC-SVM and OC-LR

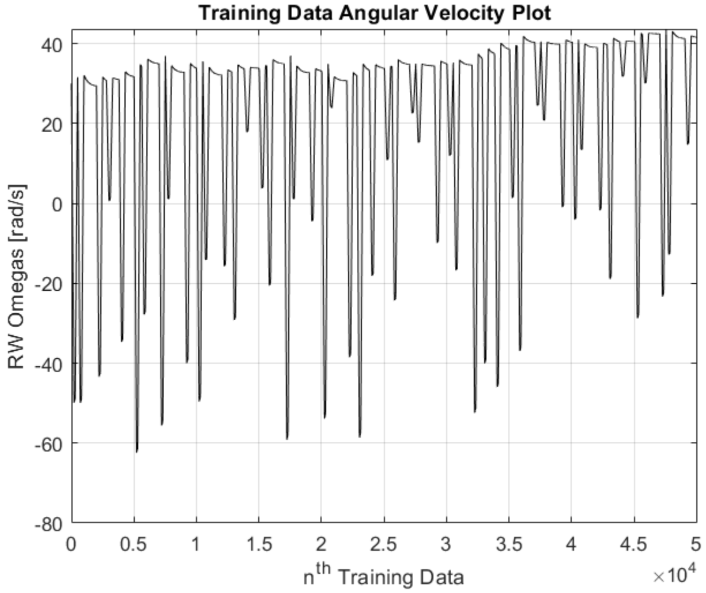

Figure 5.27: Reaction Wheel Angular Velocity Training Dataset for OC-SVM and OC-LR

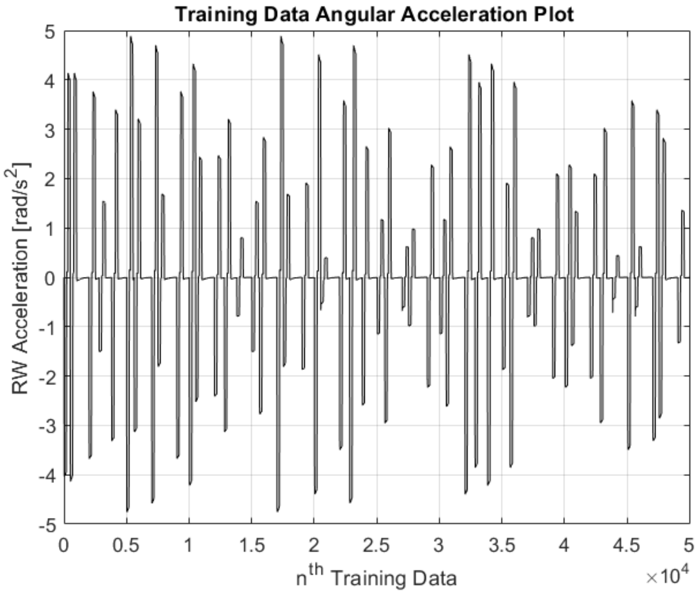

Figure 5.28: Reaction Wheel Angular Acceleration Training Dataset for OC-SVM and OC-LR

5.2.2 Simulation Tests for OC-SVM and OC-LR

After developing the training dataset for OC-SVM and OC-LR, identical simulation tests were used to examine both systems. The first test was when the viscous friction failure started at the 500^{th} second, and the second test was when the intermittent viscous friction failures were added. Testing results for OC-SVM are presented in section 5.2.2.1, and the testing results for OC-LR are presented in section 5.2.2.2.

5.2.2.1 Simulation Results for OC-SVM

The anomaly detection results from test one for the OC-SVM method is shown in Figure 5.29, and the ROC Curve for this test is shown in Figure 5.30. The detection accuracy for test one was 83.0% and the AUC was 0.8206. The random guess line and the EER line for the ROC curve is presented in Figure 5.31, where the best threshold was selected from the ERR point with the TPR of 0.85 and the FPR of 0.12.

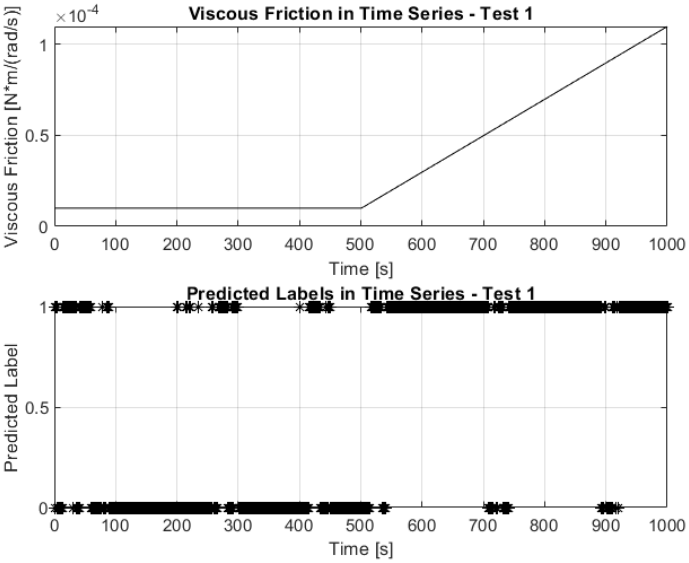

Figure 5.29: Prediction Results of Test 1 for OC-SVM

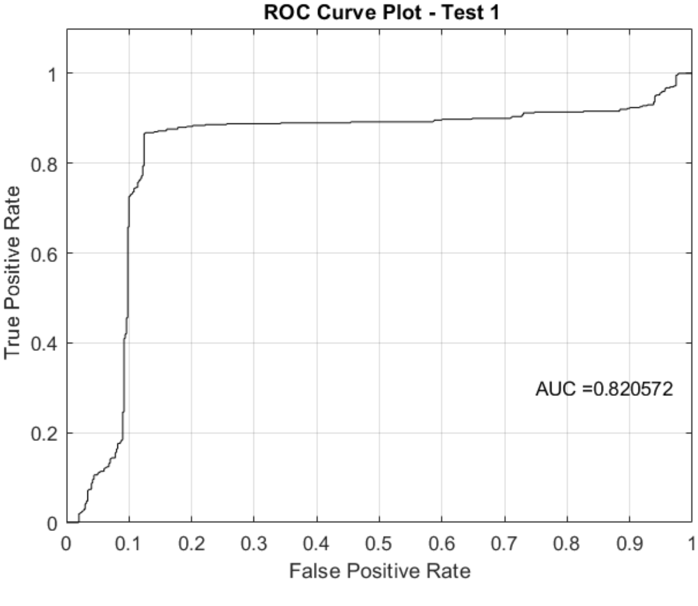

Figure 5.30: Receiver Operating Characteristic Curve of Test 1 for OC-SVM

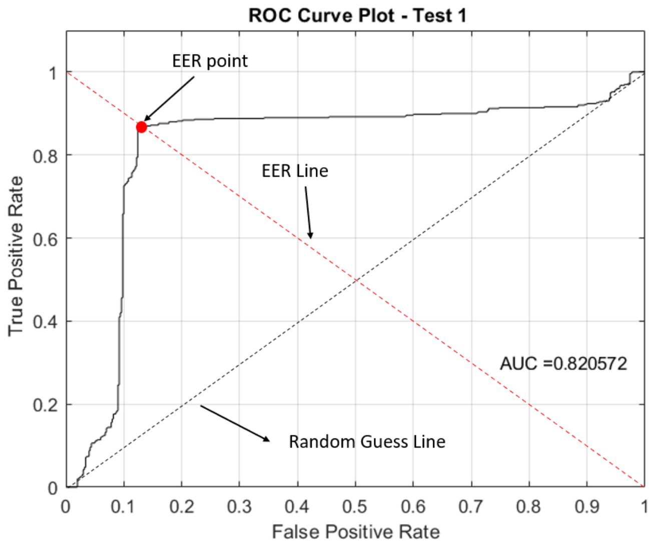

Figure 5.31: Receiver Operating Characteristic Curve of Test 1 for OC-SVM with Equal Error Rate

The detection results from test two by OC-SVM is presented in Figure 5.32 and the ROC Curve for this test is shown in Figure 5.33. The detection accuracy for test two was 61.4% with the AUC of 0.6931. The best threshold for the trained classifier in test two was from the EER point based on the EER line and the ROC curve. The EER point is shown in Figure 5.34, where the TPR was 0.61 and the FPR was 0.38.

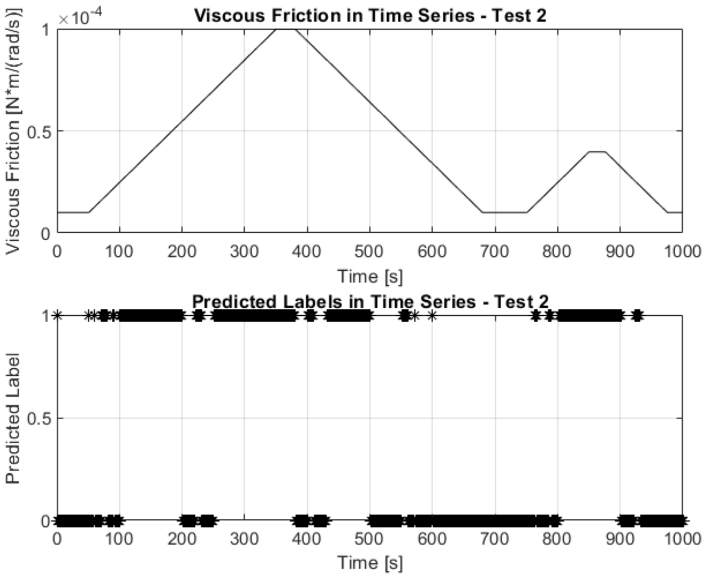

Figure 5.32: Prediction Results for Test 2 for OC-SVM

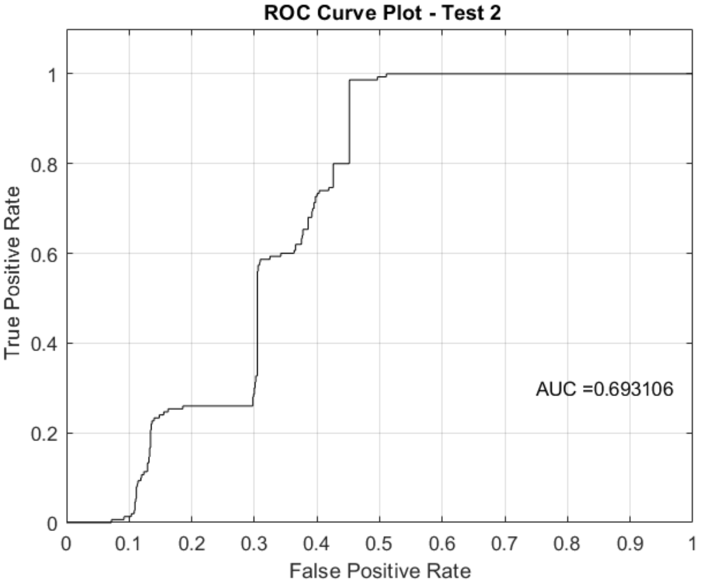

Figure 5.33: Receiver Operating Characteristic Curve of Test 2 for OC-SVM

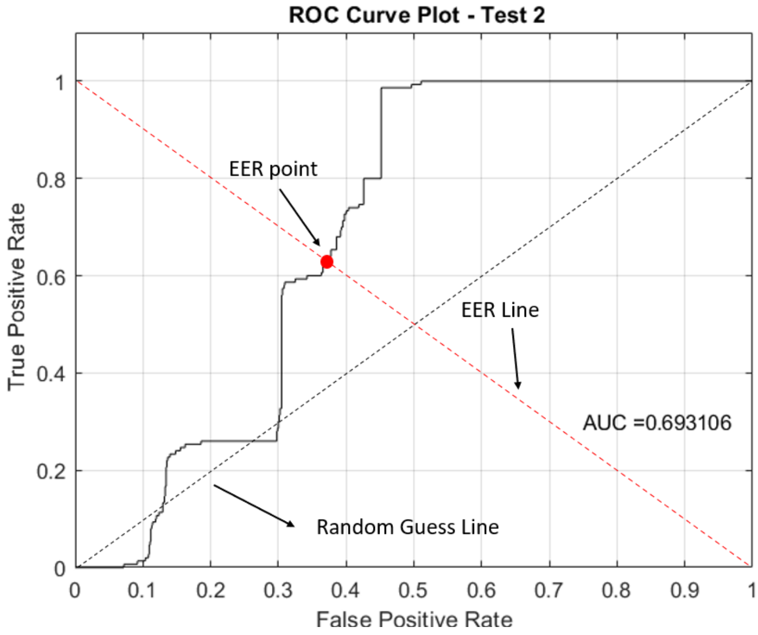

Figure 5.34: Receiver Operating Characteristic Curve of Test 2 for OC-SVM with Equal Error Point

By comparing the results from the two tests for OC-SVM, the anomaly detection system showed good performance for test one with the detection accuracy of 83.0%. The system showed a low detection accuracy on test two where the detection accuracy was around 60%. The detection system also presented great performance on identifying and diagnosing long-term friction failures while showed a relatively poor performance on detecting intermittent friction failures. According to the random guess reference line, the classifier trained by the OC-SVM method showed a good ability to distinguish the normal and abnormal cases for the first simulation test but showed a poor detection ability for the second test.

5.2.2.2 Simulation Results for OC-LR

For the OC-LR method, the coefficients of the assumed polynomial function were first computed and summarized in Table 5.1. Then, the ideal current of the training data was calculated from Eq. 4.40. The plot of predicted and measured reaction wheel currents is shown in Figure 5.35. The difference between the ideal and measured reaction wheel currents is plotted in Figure 5.36.

Table 5.1: Predicted Coefficients of the Polynomial Function for OC-LR

Computed Coefficients	a	b	c	e	f	k	c
	0.0008355	-9.6614 $\times 10^{-8}$	-1.5768 $\times 10^{-8}$	0.08606	-5.6301 $\times 10^{-5}$	7.4941 $\times 10^{-7}$	0.0008520

By combining the equations in Eq. 3.21, Eq. 3.22, Eq. 3.25, and Eq. 3.30, the actual reaction current function can be expressed as:

$$\boldsymbol{i}(t) = \frac{\boldsymbol{I}_{rw}\dot{\boldsymbol{\omega}}_{rw_{real}}(t) + sign(\boldsymbol{\omega}_{rw_{real}}) \cdot (\boldsymbol{f}_{coulomb}) + k_f \boldsymbol{\omega}_{rw_{real}}}{k_t} \tag{5.3}$$

As $\boldsymbol{I}_{rw}$, $\boldsymbol{f}_{coulomb}$, k_f and k_t were all constants, it could be observed that the actual reaction current function was in a linear relationship with the reaction wheel angular velocity and the reaction wheel angular acceleration. From Table 5.1, the predicted coefficients also proved the approximating functional relationship was correct (the initial guess of the relationship was shown in Eq. 4.33) .

The three limits according to the 68-95-99.7 rule were calculated and were also shown in the current difference plot (Figure 5.36), where the calculated limit of 1σ was ±0.004, the calculated limit of 2σ was ±0.008, and the calculated limit of 3σ was ±0.0121 with the mean of the current difference of 0.

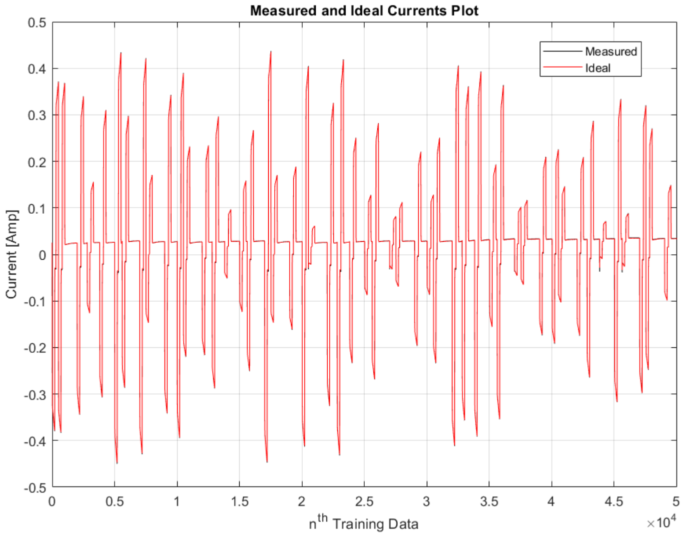

Figure 5.35: Measured and Predicted Reaction Wheel Current using OC-LR

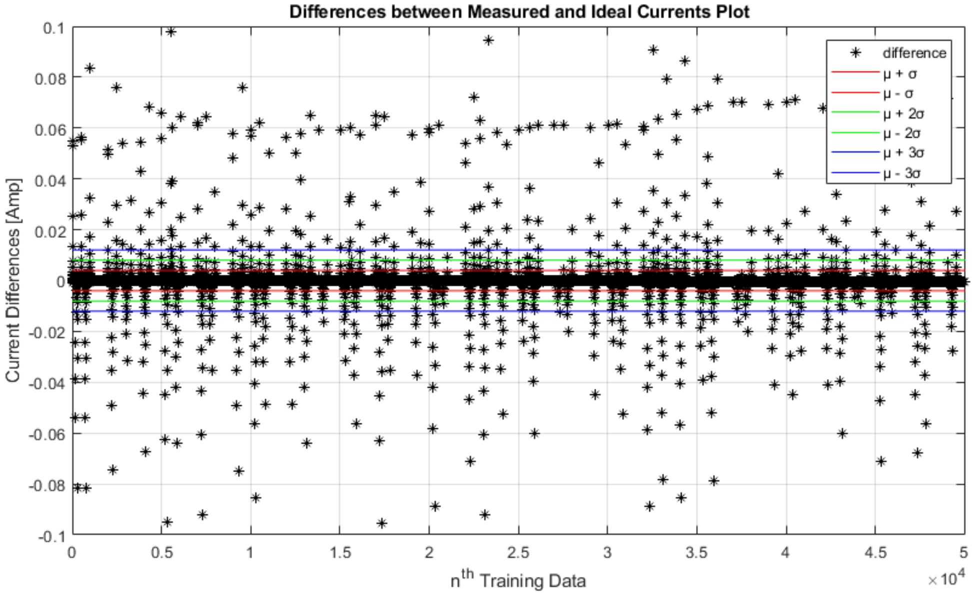

Figure 5.36: Differences between Measured and Predicted Training Current Data using OC-LR

The two simulation tests were then performed by using the OC-LR method and the behavioural labels were predicted. For the first test, the ideal testing currents were computed as shown in Figure 5.37. The difference between the ideal and the measured currents was evaluated as plotted in Figure 5.38. The behavioural labels were predicted after comparing the calculated testing current difference with the simulated measured testing current.

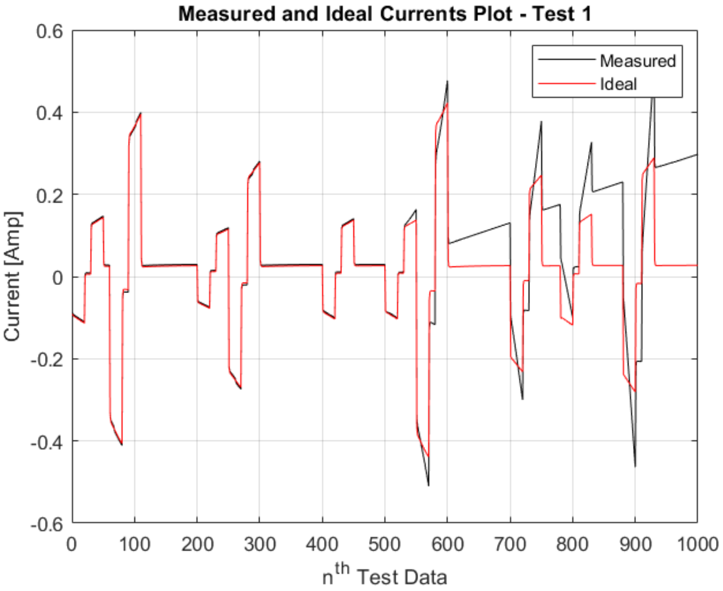

Figure 5.37: Measured and Predicted Reaction Wheel Currents for Test 1 of OC-LR

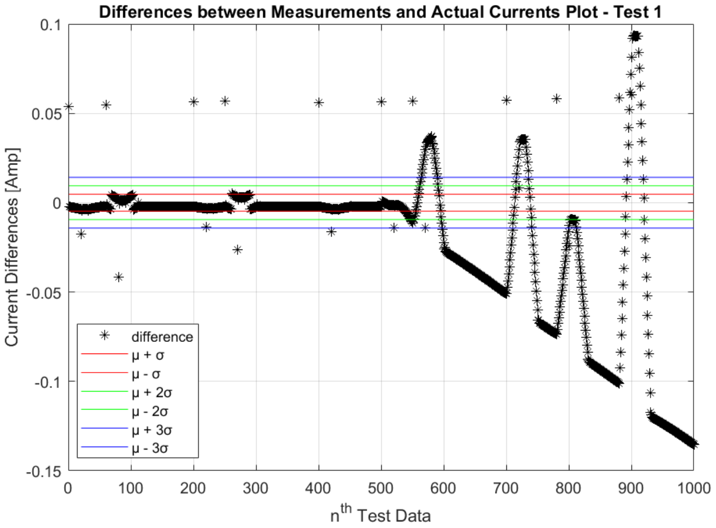

Figure 5.38: Differences between Measured and Predicted Reaction Wheel Currents for Test 1 of OC-LR

Using the 68-95-99.7 rule based on current difference from the training data, three anomaly detection predictions were made (presented in Figure 5.39). If the calculated testing current difference between the ideal and the measured current data was greater than the anomaly detection criteria limits, the behaviour was determined to be abnormal. Otherwise, the behaviour was labelled as normal. With the limit of $\mu \pm \sigma$ from training data, the detection accuracy for test one was 88.90%. With the limit of $\mu \pm 2\sigma$, the detection accuracy for test one was 92.90%. With the limit of $\mu \pm 3\sigma$, the detection accuracy for test one was 90.0%. The ROC Curve for test one using OC-LR is shown in Figure 5.40 with the AUC of 0.7667. Figure 5.41 presents the EER point selected for the trained classifier by using OC-LR where the TPR was 0.78 with the FPR of 0.23.

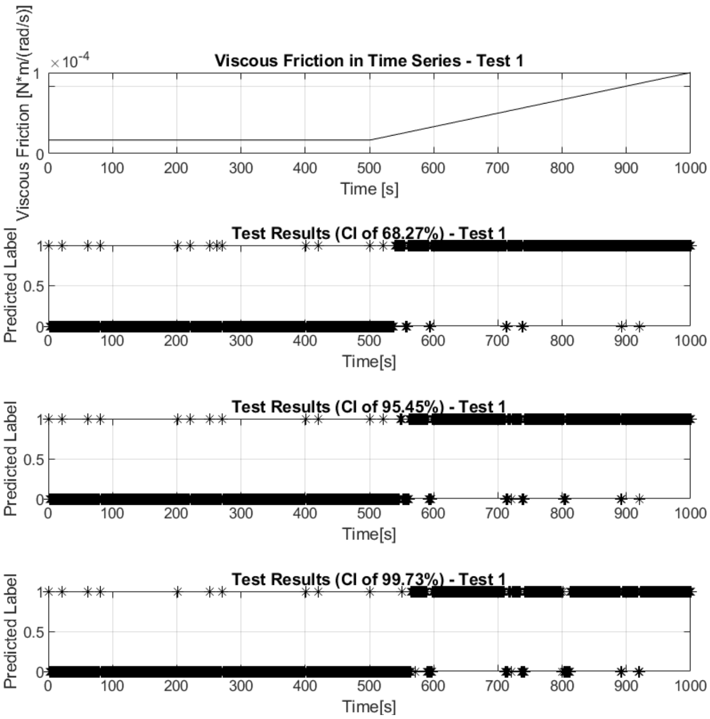

Figure 5.39: Prediction Results for Test 1 for OC-LR

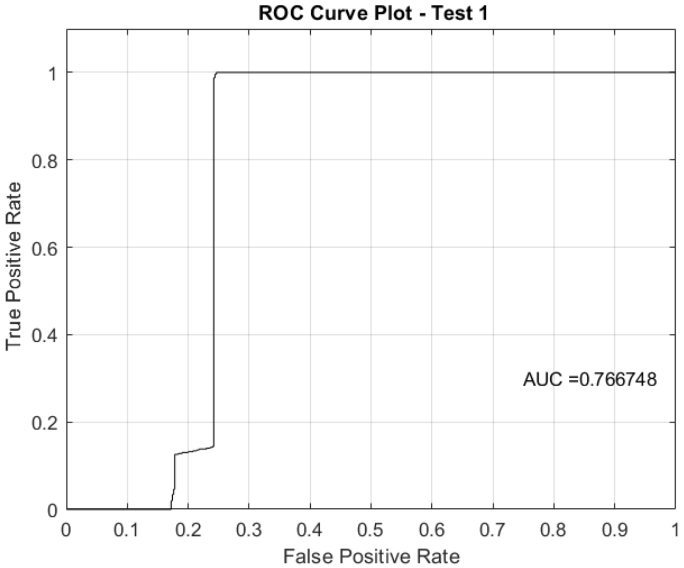

Figure 5.40: Receiver Operating Characteristic Curve of Test 1 for OC-LR

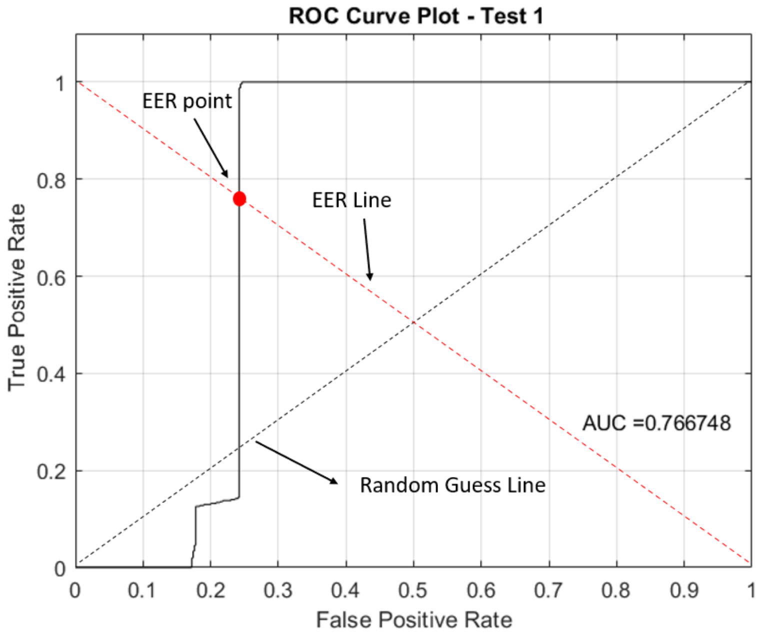

Figure 5.41: Receiver Operating Characteristic Curve of Test 1 for OC-LR with Equal Error Rate Point

The output ideal currents for test two is shown in Figure 5.42, the calculated current difference is presented in Figure 5.43, and the anomaly detection results for test two is shown in Figure 5.44. With the limit of $\mu \pm \sigma$ from training data, the detection accuracy for test two was 88.50%. With the limit of $\mu \pm 2\sigma$, the detection accuracy for test two was 83.60%. With the limit of $\mu \pm 3\sigma$, the detection accuracy for test two was 78.50%. The ROC curve for test two by using the proposed OC-LR method is shown in Figure 5.45 with the AUC of 0.7530. The selected EER point with the best threshold for the classifier in this test in shown in Figure 5.46, where the TPR was 0.78 and the FPR was about 0.25.

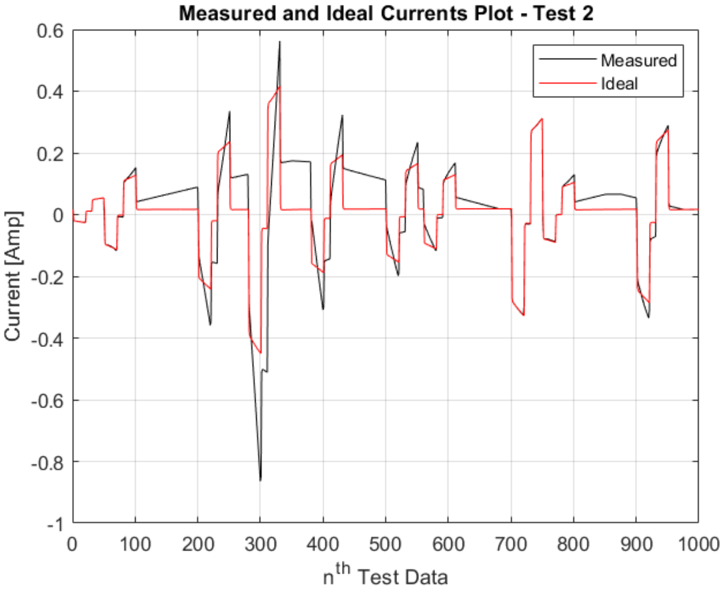

Figure 5.42: Prediction Results for Test 2 using OC-LR

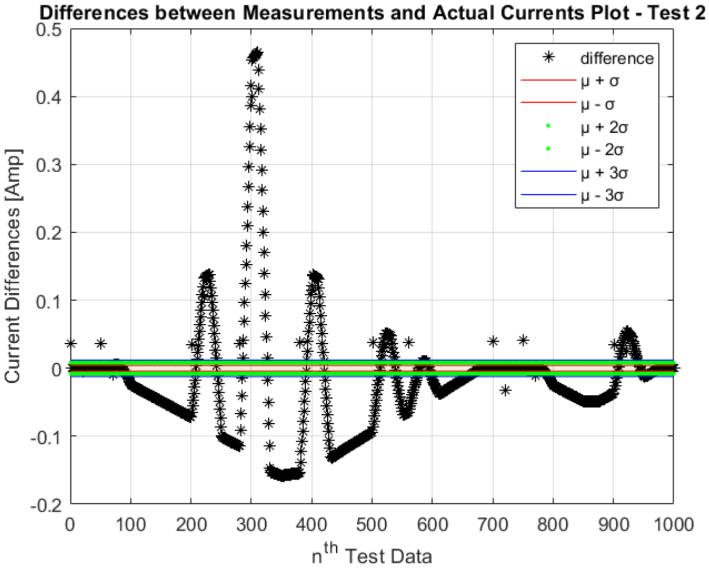

Figure 5.43: Differences between Measured and Predicted Reaction Wheel Currents of Test 2 using OC-LR

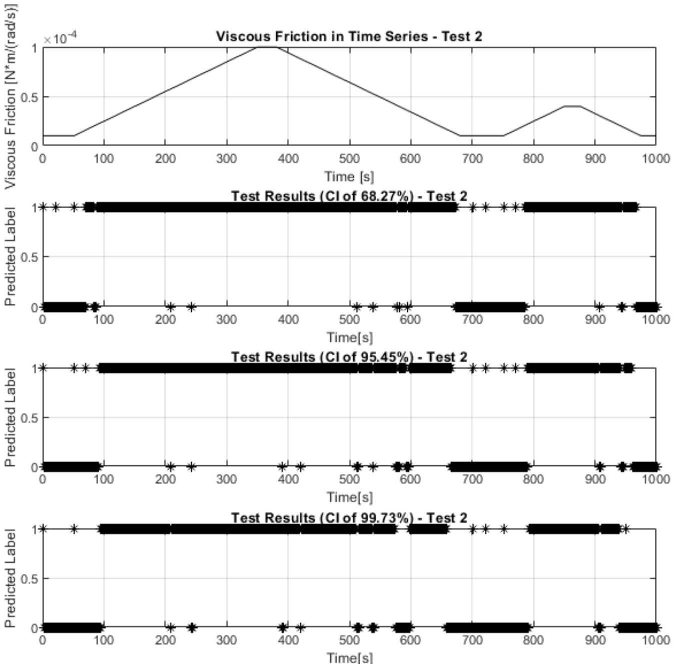

Figure 5.44: Prediction Results of Test 2 using OC-LR

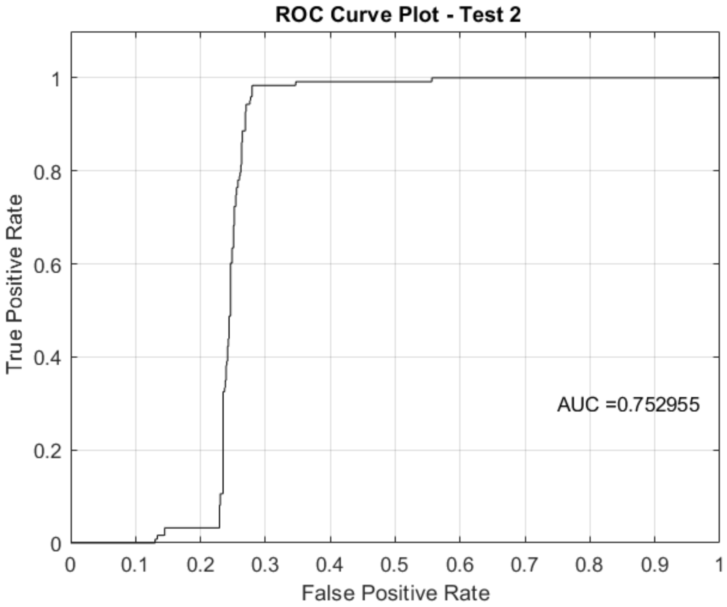

Figure 5.45: Receiver Operating Characteristic Curve of Test 2 for OC-LR

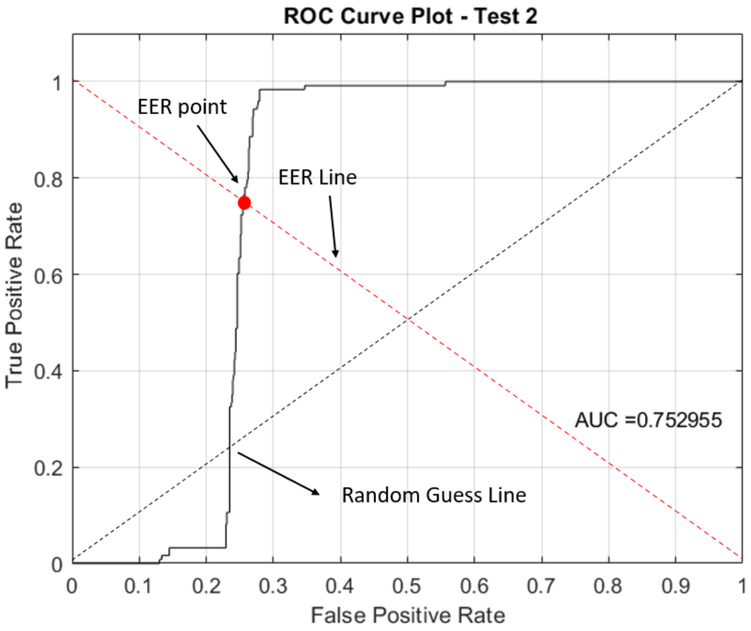

Figure 5.46: Receiver Operating Characteristic Curve of Test 2 for OC-LR with Equal Error Rate Point

By comparing with the detection results from the two tests for OC-LR, this anomaly detection system had the best detection accuracy on both simulated tests over the Logistic Regression and the OC-SVM methods (shown in Table 5.2). With the limit of 95.45%, the detection accuracy for test one was around 93.0%. The detection accuracy was 88.50% for test two with the limit of 68.27%. Although the OC-LR method showed the best detection results on these two simulated tests, the OC-SVM method showed the best trained classifer for the first test - the AUC for test one was 0.7981. A higher AUC value represents better trained classifier. The OC-LR has the best trained classifier for test two with the AUC of 0.7530. The Logistic Regression method has a relatively average trained performance on both simulated tests with the AUC of 0.7981 for the first test and the AUC of 0.7219 for the second test. According to the ROC curves for each test with the random guess line, the classifier trained by the OC-LR method showed a poor ability to distinguish the normal and abnormal cases for the simulation tests before reaching the EER point.

From the perspective of the training process, the Logistic Regression method and the OC-LR method required prior knowledge of the training data. In terms of the Logistic Regression method, both normal and abnormal behavioural data have to be included in the training dataset where the normal and abnormal should have a clear boundary between each other. Such system is more capable of detecting apparent failures in the spacecraft's behaviour. For the OC-LR method, although it did not need abnormal data for training, it required an initial guess for the potential functional relationship between the training features. From that, space engineers and experts may need to stay on-site if the OC-LR detection system is put in use for a mission.

Table 5.2: Simulated Anomaly Detection Tests

	Logistic Regression (Two-Sided)	**One-class Support Vector Machine (One-Sided)**	**Linear Regression (One-Sided)**
Simulated Test 1 Accuracy	85.90%	83.0%	88.90% (CI of 68.27%) 92.90% (CI of 95.45%) 90.00% (CI of 99.73%)
Area under the curve	0.7981	0.8206	0.7667
Simulated Test 2 Accuracy	62.50%	61.40%	88.50% (CI of 68.27%) 83.60% (CI of 95.45%) 78.50% (CI of 99.73%)
Area under the curve	0.7219	0.6931	0.7530

Chapter 6

Physical Test Results and Discussion

In this chapter, physical training data and the training process for the one-sided learning anomaly detection systems are presented. A motor test stand was used to mimic the behaviour of a real motor. The details of the physical motor test, test setup and procedure will be illustrated. The proposed one-sided learning methods discussed in Chapter 4, the OC-SVM and OC-LR methods, are validated by the motor test. The physical anomaly test results for both methods will be compared and discussed.

6.1 Test Setup

The proposed one-sided learning system was tested by a real physical motor test. Similar to the simulated tests performed in the last chapter, the one-sided learning detection system was only trained by the nominal behaviour of the motor and was expected to detect friction failures to distinguish between normal and abnormal motor motion.

The test was performed with a Tyro Robotics Series 1580 TestStand, which includes a brushless motor with drone propeller. I used the brushless motor to mimic the reaction wheel behaviour, and changed the size of the drone propeller to mimic a friction failure on the motor. There were two propellers used in the physical tests (shown in Figure 6.1). The big propeller (with a size of 9.0” × 4.0”) was expected to provide more friction (drag) and would need more current to actuate. The smaller propeller (with a size of 7.0” × 4.0”) was expected to provide less friction and would need less current to actuate. Figure 6.2 shows the TestStand with the brushless motor and Figure 6.3 shows the whole physical test setup with the 9.0” × 4.0” propeller. In this test, the behavioural data collected from the 7.0” × 4.0” propeller was deemed to be normal. The data collected from the 9.0” × 4.0” propeller was assumed to be failure data.

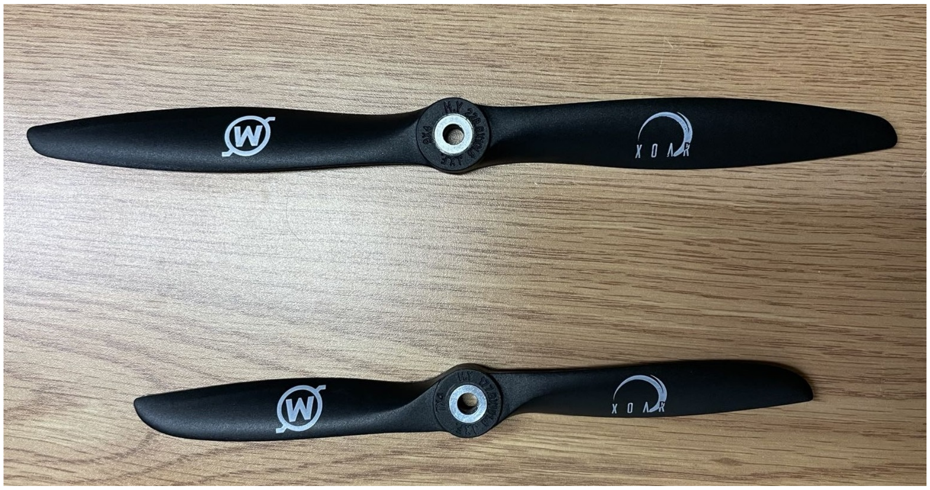

Figure 6.1: Motor Test Propellers

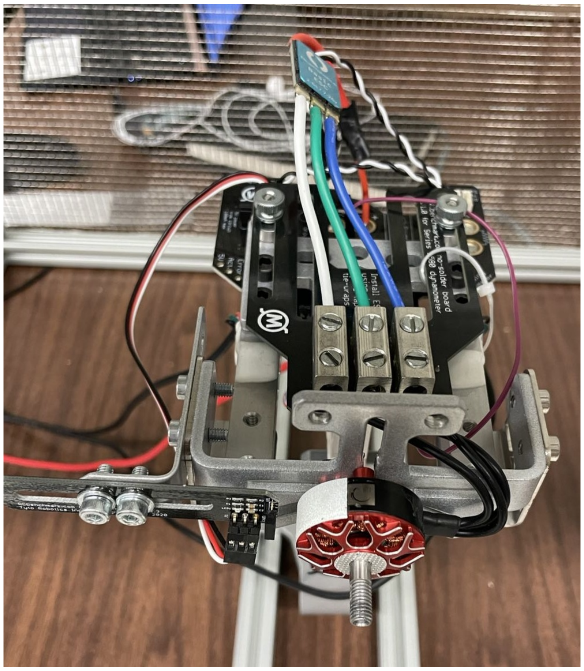

Figure 6.2: Motor Test Stand

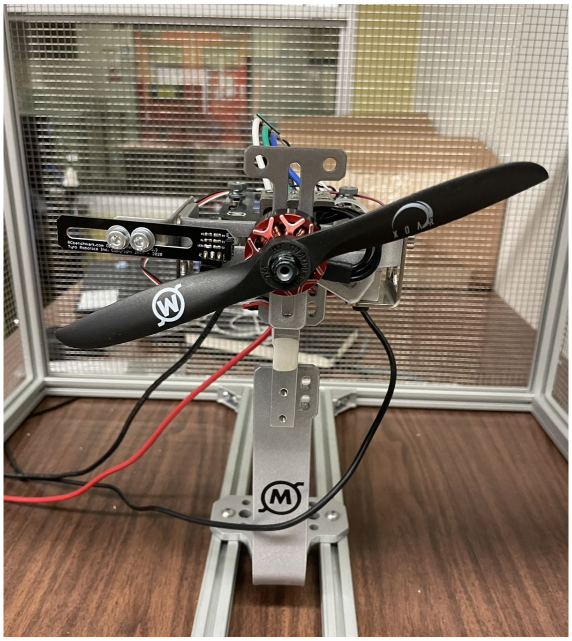

Figure 6.3: Motor Test Stand with Propeller

First, the training data for the motor test was collected. Since the physical test was to examine the one-sided learning anomaly detection system, training data was only collected from the the 7.0” × 4.0” propeller. The training data was collected for 4000 seconds in time series. The motor current, angular velocity and angular acceleration were used as the training features. The collected training data of motor current, angular velocity and angular acceleration are shown in Figure 6.4, Figure 6.5, and Figure 6.6 respectively.

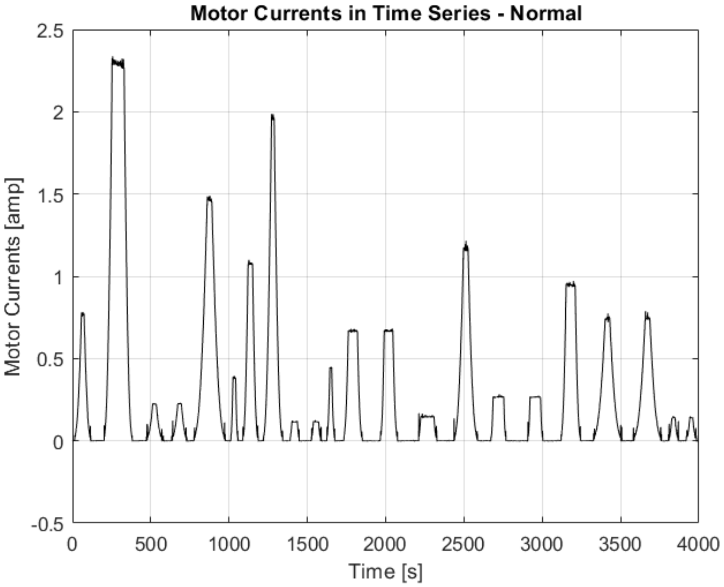

Figure 6.4: Collected Motor Training Current in Time Series (Normal)

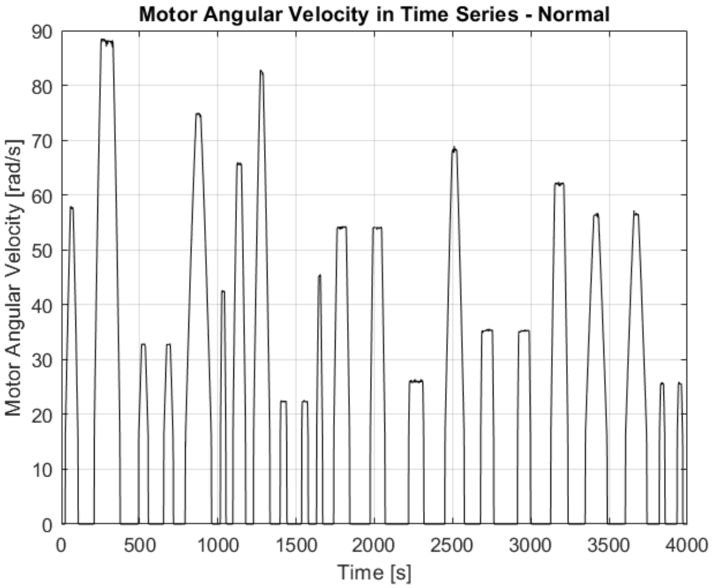

Figure 6.5: Collected Motor Training Angular Velocity in Time Series (Normal)

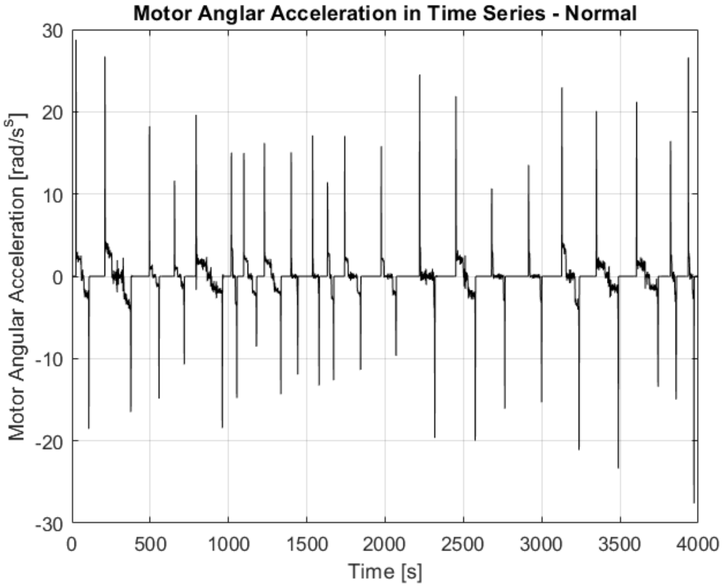

Figure 6.6: Collected Motor Training Angular Acceleration in Time Series (Normal)

There was only one test performed for the physical test. For the test, 2290 seconds data was collected. The first 1065-second data was all collected from the 7.0” × 4.0” propeller as the behaviour from the smaller propeller was assumed to be normal. At the 1065^{th} second, the friction failure was designed to happen. To mimic the friction failure, the 7.0” × 4.0” propeller was replaced by the 9.0” × 4.0” propeller. The collected motor current is in Figure 6.7, the angular velocity is in Figure 6.8, and the angular acceleration is in Figure 6.9.

To test the generality of the proposed one-sided learning anomaly detection system, two types of the motion were included in this test. During the first 1065 seconds, the motor performed intermittent motions where stops were added between slews. At the 1065^{th} second, an initial speed was given to the motor where the continuous motion started.

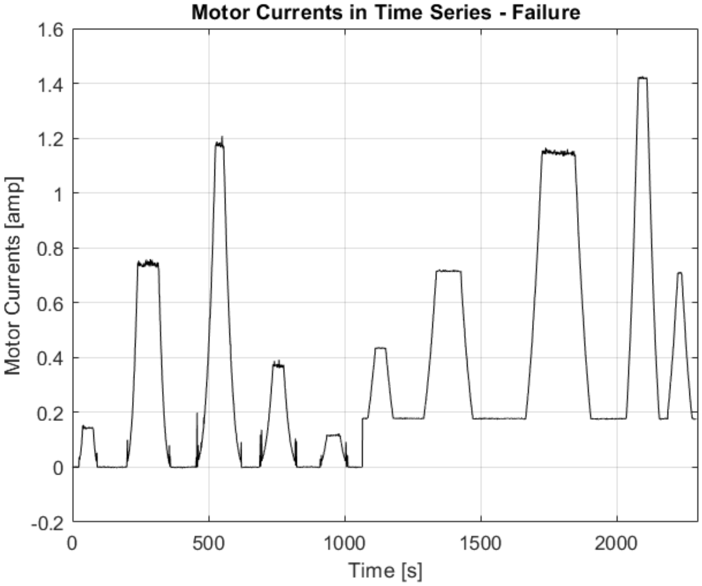

Figure 6.7: Motor Test Current in Time Series

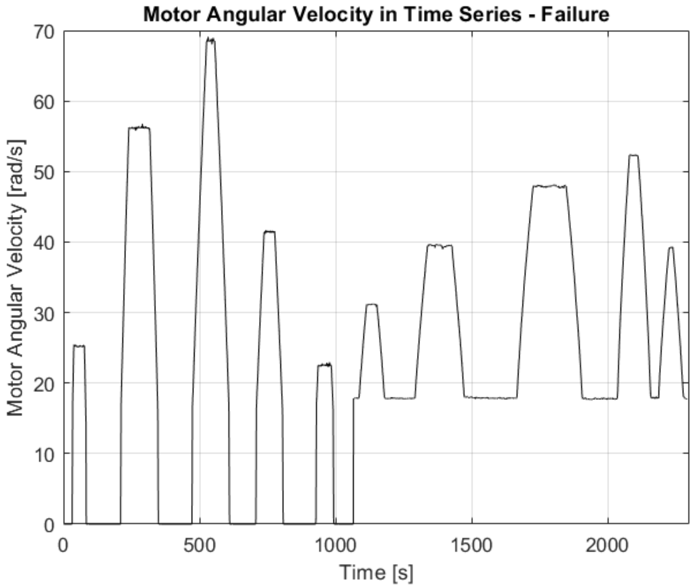

Figure 6.8: Motor Test Angular Velocity in Time Series

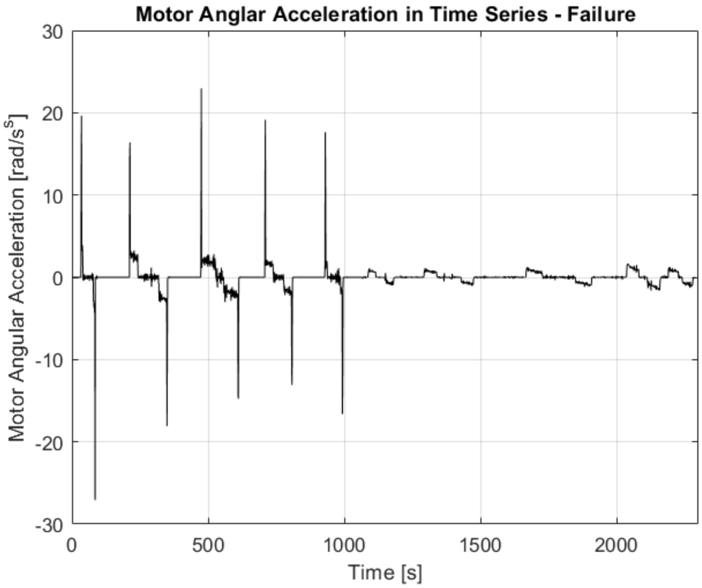

Figure 6.9: Motor Test Angular Acceleration in Time Series

The training data collected from the motor (normal data collected from the 7.0” ×

4.0” propeller) was fed into the proposed one-sided learning basis anomaly detection systems. After training the system using OC-SVM and OC-LR method, the physical motor test was performed. The detection results from the OC-SVM method is shown in Figure 6.10. The ROC curve for motor test for OC-SVM method is presented in Figure 6.11 and the EER point for this test is shown in Figure 6.12. The OC-SVM presented an aproximately perfect detection results on the physical motor test with accuracy of 97.25% and an AUC of 0.9723. The best threshold for the motor test by using OC-SVM was selected from the EER point where the TPR was 0.97 and the FPR was about 0.05.

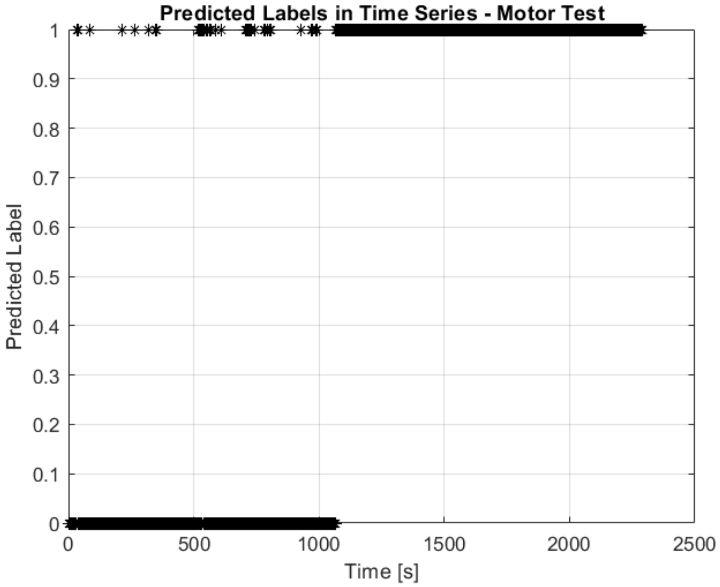

Figure 6.10: Motor Test Results from OC-SVM

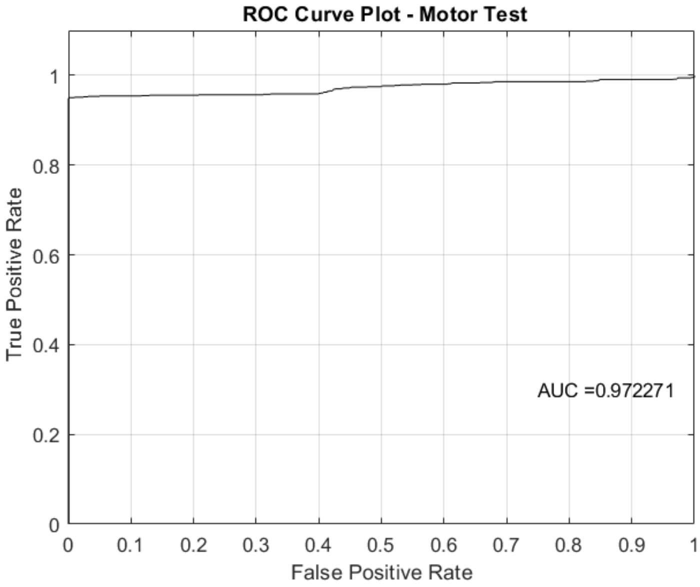

Figure 6.11: Receiver Operating Characteristic Curve for Motor Test by OC-SVM

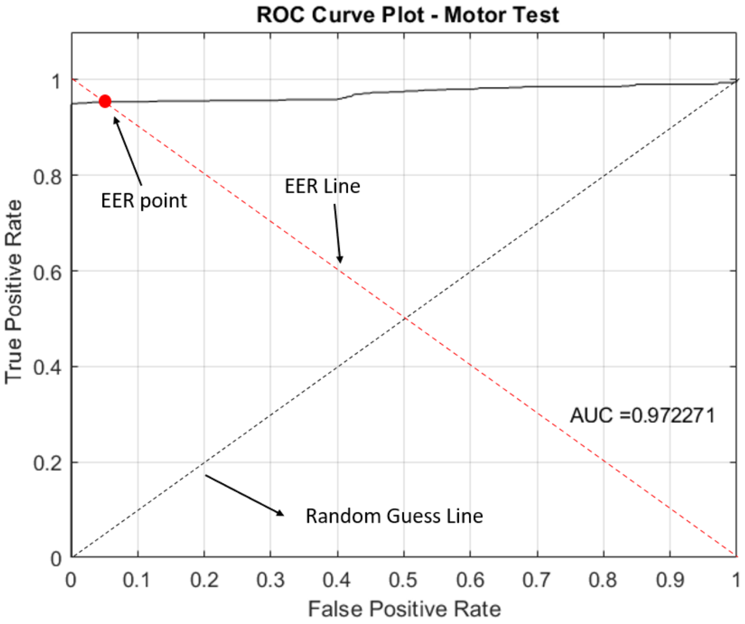

Figure 6.12: Receiver Operating Characteristic Curve for Motor Test by OC-SVM with Equal Error Rate Point

For the OC-LR method, the coefficients for the motor current function was first computed and shown in Table 6.1. Then ,the predicted motor current was computed (shown in Figure 6.13). Similar to the simulation tests, the difference between the predicted current and the measured current was calculated (Figure 6.14). The three limits, $\mu \pm \sigma$, $\mu \pm 2\sigma$ and $\mu \pm 3\sigma$, were evaluated based on the calculated the current difference and presented in Figure 6.14.

Table 6.1: Predicted Coefficients of the Motor Function for OC-LR

Computed Motor Coefficients	a	b	c	e	f	k	c
	0.002821	2.9003 $\times 10^{-5}$	2.6506 $\times 10^{-6}$	0.001482	7.4120 $\times 10^{-5}$	-5.0446 $\times 10^{-5}$	0.009403

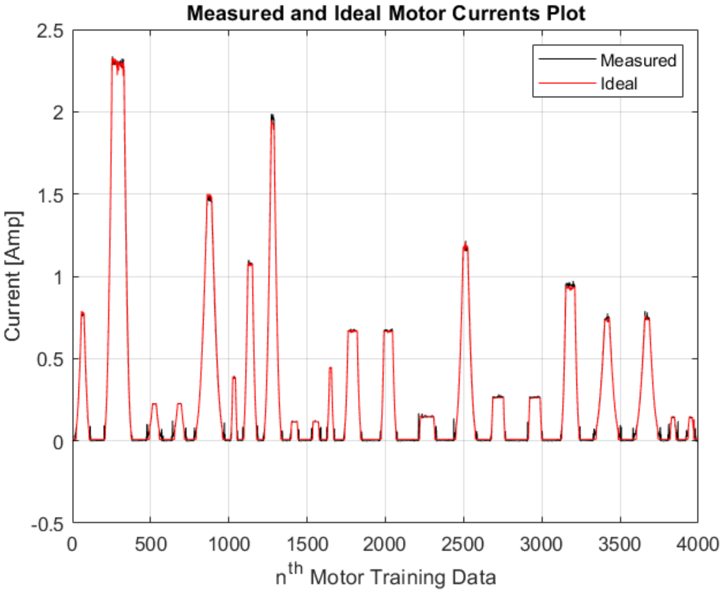

Figure 6.13: Measured and Predicted Motor Currents for OC-LR

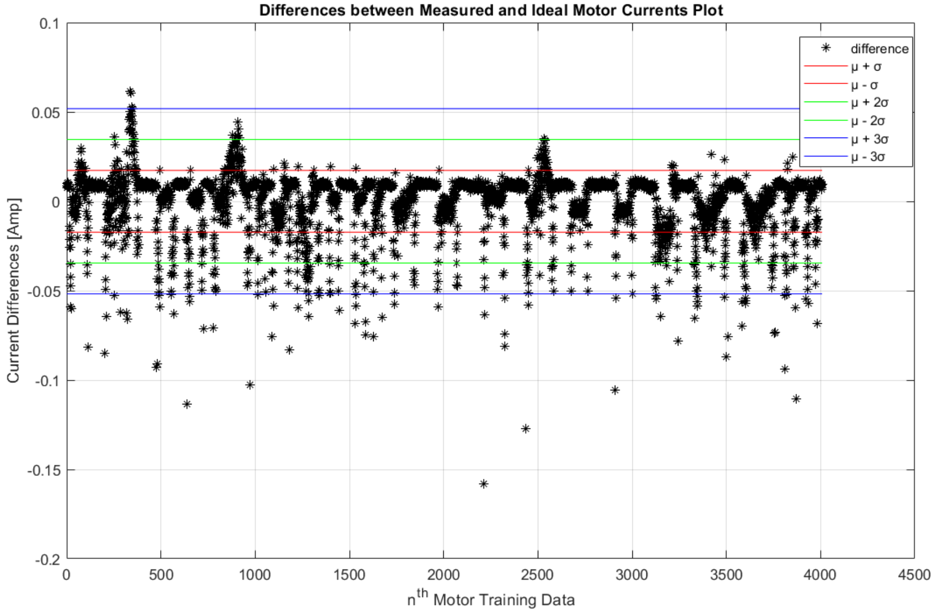

Figure 6.14: Differences between Meausred and Predicted Motor Currents for OC-LR

The motor test was then performed by using this method. The expected current was first predicted with the input motor speed and acceleration (see in Figure 6.15). Then the test current differences were calculated and the three limits from training data were utilized to diagnose the failure. The plot for the test current differences and the calculated three limits are shown in Figure 6.16.

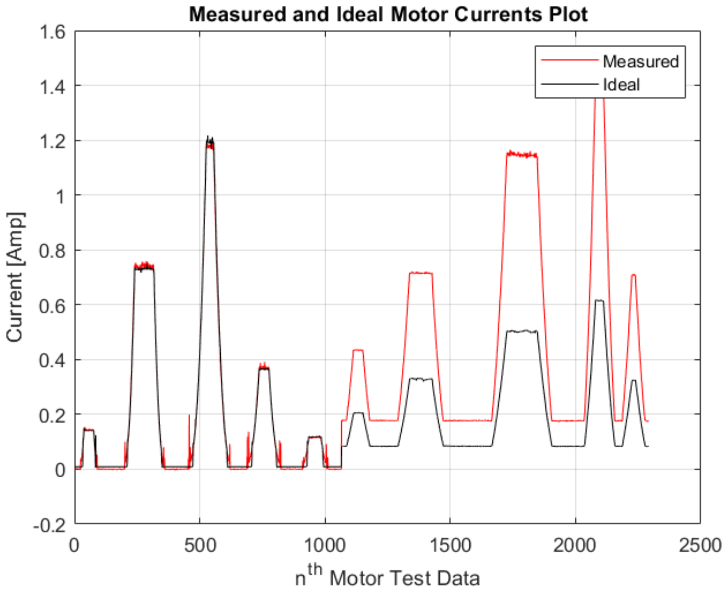

Figure 6.15: Measured and Predicted Currents for Motor Test by OC-LR

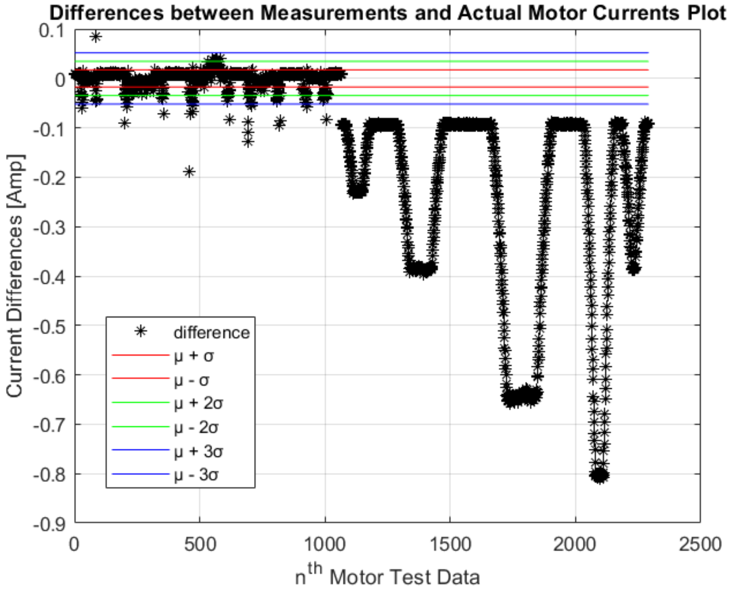

Figure 6.16: Differences between Measured and Predicted Motor Test Data for OC-LR

The detection results are shown in Figure 6.17 and the ROC Curve is in Figure 6.18. With the limit of $\mu \pm \sigma$ from motor training data, the detection accuracy for the physical test was 91.71%. With the limit of $\mu \pm 2\sigma$, the detection accuracy for the motor test was 97.25%. With the limit of $\mu \pm 3\sigma$, the detection accuracy for test two was 99.13%. The ROC curve for test two using the proposed OC-LR method is shown in Figure 6.18 with the AUC of 0.9983. The EER point for the OC-LR trained classifier is shown in Figure 6.19 with the TPR of about 1 and the FPR of 0.

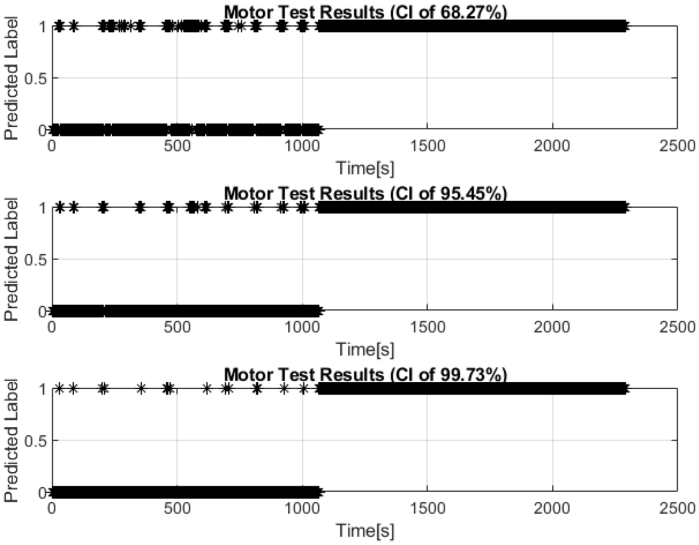

Figure 6.17: Motor Test Results from OC-LR

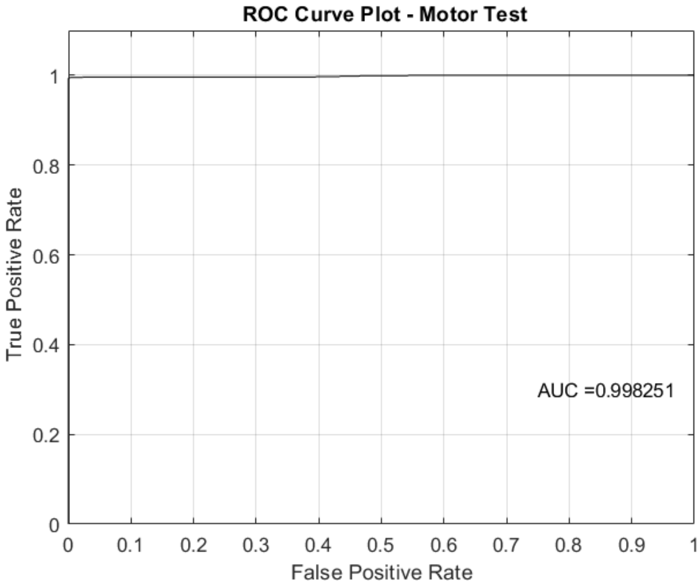

Figure 6.18: Receiver Operating Characteristic Curve for Motor Test

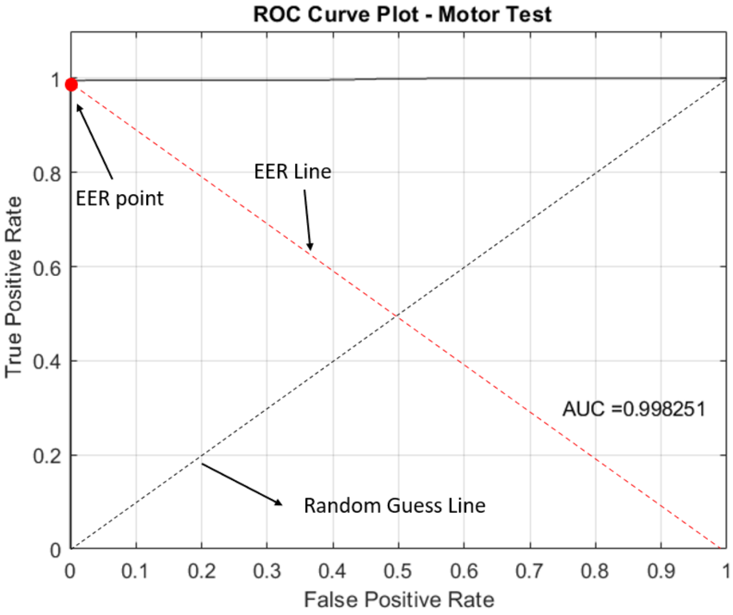

Figure 6.19: Receiver Operating Characteristic Curve for Motor Test with Equal Error Rate

A summary of the physical motor test results for the proposed one-sided learning anomaly detection system is shown in Table 6.2. By applying the OC-SVM method, the detection results showed high accuracy of 97.25%, and the OC-LR method showed a relatively high accuracy of 99.13% with a limit of 99.73%. Both methods, the OC-SVM and OC-LR, had almost perfect AUC of 0.9723 and 0.9983 respectively. Comparatively, the OC-LR system had a better trained classifer than the OC-SVM system. By comparing the ROC curves for each test with the random guess line, the classifier trained by the OC-SVM method showed a good ability to distinguish the normal and abnormal cases for the simulation tests. By comparing the ROC curves for each trained classifier with the random guess line for the motor test, the classifier trained by the OC-LR method showed a better ability to distinguish between the normal and abnormal cases than the OC-SVM method.

Table 6.2: Physical Test Results for OC-SVM and OC-LR

	One-class Support Vector Machine (One-Sided)	**Linear Regression (One-Sided)**
Physical Test Accuracy	97.25%	91.71% (CI of 68.27%) 97.25% (CI of 95.45%) 99.13% (CI of 99.73%)
Area under the curve	0.9723	0.9983

Chapter 7

Conclusion and Future Work

The overall objective of this research was to develop a feasible automated anomaly detection system for a ground station that is capable of learning the normal behaviour of a spacecraft over a period of time without requiring any expert knowledge of the spacecraft and report subtle deviations and anomalies when detected. I proposed two hypotheses to achieve such an autonomous ground station. The hypotheses were:

Hypotheses 1: A ground station can learn the nominal behaviour of a spacecraft attitude control system after observing a specific failure-free period of a spacecraft operations using an artificial neural network or other suitable deep learning method.

Hypotheses 2: A ground station can detect un-trained failures from the performance of the spacecraft attitude control system after a primary learning phase from a specific failure-free period.

To accomplish the proposed objective and investigate hypotheses, in this research, I developed a ground-based automated anomaly detection system for a small satellite attitude dynamics control system by applying both two-sided and one-sided machine learning algorithms. Both two-sided and one-sided learning detection systems were expected to identify and detect anomalies when there was a subtle bahavioural failure in the attitude control system. The proposed fault-detection system was able to identify and diagnose small viscous friction failures by learning from the behaviour of the satellite's attitude control system in a specific period, specifically in the commissioning phase of the satellite mission. Logistic Regression methodology was used to develop a two-sided learning anomaly detection system. One-Class Support Vector Machine (OC-SVM) and One-Class Linear Regression (OC-LR) methods were used to develop the one-sided learning anomaly detection system.

First, the two-sided learning anomaly detection system was developed to provide an insight for the development of the one-sided learning system to achieve my research goals. To implement the two-sided learning anomaly detection system, I used both normal and abnormal behaviour from the attitude dynamics system of the spacecraft for training. The behavioural data was mainly from the reaction wheel outputs in the attitude dynamics system. From the available reaction wheel outputs, reaction wheel current, angular velocity, and angular acceleration were selected as the training features. The normal behaviour of the reaction wheel was simulated from healthy trajectories with a nominal viscous friction. The abnormal behavioural data was created from bearing degradation failures where the friction was greater than the nominal viscous friction. Both normal and abnormal data were labelled as 0 and 1 ("Normal" and "Abnormal") and were used to train the system. The system was expected to explore a logistic relationship between the spacecraft's behavioural data and behavioural labels during operations. A decision boundary was determined to classify the spacecraft behaviour after training.

The one-sided learning anomaly detection system was developed by only learning from the nominal behaviour of the satellite where data only from healthy slews was collected for training. For the OC-SVM method, a one-class classifier was built from a hyperplane decision function during training. The adaptive Sequential Minimal Optimization (SMO) method was utilized to solve the quadratic problem in the application of OC-SVM algorithm to provide an optimal solution for the hyperplane decision function. For the OC-LR method, a polynomial function was computed during training to describe the relationship between reaction wheel current, angular velocity, and angular acceleration. The Empirical Rule (68-95-99.7 Rule) was used in this method where three classifiers were built based on this rule to classify the data.

Two simulated tests and one physical test were developed to validate the feasibility and detection accuracy of the proposed anomaly detection systems. In the simulated tests, untrained reaction wheel bearing failures were added into the attitude control system to examine whether the fault-detection system was capable of detecting and diagnosing the reaction wheel failures. After demonstrating the system in simulation, a motor physical test was performed using the one-sided learning anomaly detection systems where I replicated the tests on real motor hardware, simulating failed bearings by adding friction to the motor. Similar to the simulations, the detection system was only trained by nominal behaviours from the motor and was able to detect friction failures to distinguish between normal and abnormal motor motion. The motor anomaly test illustrated and proved the feasibility and generality of the proposed one-sided learning fault detection system for space systems. The detection performances for each system are analyzed and summarized in Table 7.1. Pros and cons for each developed anomaly detection system are summarized in Table 7.2.

Table 7.1: Anomaly Detection Performance for Proposed Automated Systems

	Logistic Regression (Two-Sided)	**One-class Support Vector Machine (One-Sided)**	**Linear Regression (One-Sided)**
Simulated Test 1 Accuracy	85.90%	83.0%	88.90% (CI of 68.27%) 92.90% (CI of 95.45%) 90.00% (CI of 99.73%)
Area under the curve	0.7981	0.8206	0.7667
Simulated Test 2 Accuracy	62.50%	61.40%	88.50% (CI of 68.27%) 83.60% (CI of 95.45%) 78.50% (CI of 99.73%)
Area under the curve	0.7219	0.6931	0.7530
Physical Test Accuracy	-	97.25%	91.71% (CI of 68.27%) 97.25% (CI of 95.45%) 99.13% (CI of 99.73%)
Area under the curve	-	0.9723	0.9983

Table 7.2: Summary of Pros and Cons for Proposed Automated Systems

Methods	**Pros**	**Cons**
Logistic Regression	- Easy to implement - Less time to compute results - Good performance on detecting long-term failures	- Require binary labelled data for training - Require clear boundary between normal and abnormal data
One-Class Support Vector Machine (OC-SVM)	- Only requires nominal data for training - Good performance on detecting long-term failures	- Require clear boundary between normal and abnormal data - Relatively hard to implement - Poor performance on detecting intermittent failures
One-Class Linear Regression (OC-LR)	- Only require nominal data for training - Easy to implement - Good performance on detecting failures for all tests	- Require an initial guess for potential functional relationship - Affected by noise

Overall, the systems based on the Logistic Regression and One-Class Linear Regression (OC-LR) methods were easier to accomplish than the One-Class Support Vector Machine (OC-SVM) method. Both Logistic Regression and One-Class Support Vector Machine (OC-SVM) algorithms had limitations and restraints on the training data and process. There were more advantages to use the One-Class Linear Regression (OC-LR) method to build the anomaly detection system. However, it required prior knowledge on the potential relationship between the training features.

To diagnose a failure for the One-Class Linear Regression (OC-LR) method, three limits (68.27%, 95.45%, and 99.73%) were tested in this research. From the analysis of the area under curve (AUC), each test showed the same performance on the trained classifer where the AUC was identical when using different limits. Regarding to the detection accuracy, CI of 95.45% present a relatively stable results for all three tests. Although it did not have the best accuracy on all tests, it was less susceptible to be affected by the noise of the data. Therefore, in this research, the One-Class Linear Regression (OC-LR) was considered as the most suitable method to put in use for a future space mission.

The main contributions of my research were:

1. A mathematical model of a spacecraft attitude control system, including a cluster of reaction wheels that enabled custom bearing failures to be injected.

2. An evaluation of the performance of a two-sided logistic-function-based machine learning algorithm for detecting subtle changes in reaction wheel bearing friction based only on simulated downlinked satellite data.

3. An analysis and a validation of the performances of a one-sided logic-regression-based and a one-sided-classification-based machine learning algorithms for detecting subtle changes in reaction wheel bearing friction based on both simulated downlinked satellite data and physical motor experiments data.

Future work on this research can be improving the detection accuracy and performance on the simulation tests for all methods. To improve the detection accuracy and performance, an analysis of the training data can be first conducted to discover the impacts from the training data and process on the different anomaly detection tests. From the training process in this research, I found out that different sets of training data had different effects on the testing results. By adding more training data or involving more training features for the learning process, it may improve the current detection accuracy.

Further, more effort can be put to develop different fault-detection systems by implementing other suitable one-sided learning algorithms as well as with different types of failure tests. One suitable one-sided learning algorithm can be One-Class Neural Networks (OC-NN) method. The OC-NN system is also expected to only learn from the nominal behaviour of the satellite using neural network layers. Same simulation tests can be used to test the system. If an input test datapoint is accepted by the trained OC-NN system, the test data will be labelled as normal. If an input test datapoint is rejected by the trained OC-NN system, the test data will be labelled as abnormal.

Different types of failure in the spacecraft's attitude dynamics control system can be considered as some common failures that may happen during a space mission, such as errors in reaction wheel speed feedback, increase in reaction wheel electrical resistance, etc. Furthermore, the proposed anomaly detection system can also be used in other subsystems of the satellite (e.g., power subsystem, thermal subsystem) to detect over or undervoltage failures and temperature failures. And with the rapid development in the AI field, novelty anomaly detection systems using multi-class classification method has become more popular. The system will not be limited to only detecting one kind of failure but is capable of detecting multiple failures at the same time. The multi-class anomaly detection system can be further developed on the basis of the current simulations that I have performed in this research.

References

[1] Y. Huang and P. A. Ferguson, "Attitude dynamics and control anomaly detection using an autonomous ground station," presented at the 2021 AIAA Scitech Conference, Jan. 2021, ISBN: 978-1-62410-609-5. DOI: 10.2514/6.2021-0665.

[2] Y. Huang and P. A. Ferguson, "Automated fault-detection for small satellite pointing control systems using one-sided learning," presented at the 35th Annual Small Satellite Conference, Jul. 2021. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2021/all2021/36/.

[3] P. Fortescue, S. Graham, and S. John, *Spacecraft Systems Engineering.* A John Wiley & Sons, Ltd., Aug. 2011, pp. 467–494. DOI: 10.1002/9781119971009.ch14.

[4] "Soa 2021 - ground data systems and mission operations," 2021. [Online]. Available: https://www.nasa.gov/smallsat-institute/sst-soa/ground-data-systems-and-mission-operations.

[5] V. Abbasi, N. Jackson, M. Doyon, *et al.*, "Neossat recovery following magnetometer and torque rod failure," presented at the 15th International Conference on Space Operations, 2018, ISBN: 9781624105623. DOI: 10.2514/6.2018-2664.

[6] F. Piatti, M. Walker, F. Figueroa, and L. Underwood, "A case study on the challenges and opportunities for the deployment of phm capabilities in existing engineering systems," vol. 2021-March, in Proceedings of IEEE Aerospace

Conference, Mar. 2021, ISBN: 9781728174365. DOI: 10.1109/AERO50100.2021.9438426.

[7] Q. Li and X. Peng, "Application of large-data-driven phm technology in satellite test and on-orbit management," presented at the 2017 Prognostics and System Health Management Conference (PHM-Harbin), Jul. 2017, pp. 1–5, ISBN: 978-1-5386-0370-3. DOI: 10.1109/PHM.2017.8079214.

[8] A. D. Lewis and K. M. Groth, "Metrics for evaluating the performance of complex engineering system health monitoring models," *Reliability Engineering System Safety*, vol. 223, p. 108 473, Mar. 2022, ISSN: 0951-8320. DOI: 10.1016/j.ress.2022.108473.

[9] J. Gal-Edd, J. Steck, C. Fatig, and R. Zepp, "Low cost ground systems - fantasy or reality," in Proceedings of IEEE Aerospace Conference, Mar. 2002, p. 7, ISBN: 0-7803-7231-X. DOI: 10.1109/AERO.2002.1035296.

[10] M. Schmidt, J. Bolvansky, and K. Schilling, "Satellite operation improvement through efficient data combination in ground station networks," *IFAC Proceedings Volumes*, vol. 43, pp. 524–530, 2010, ISSN: 14746670. DOI: 10.3182/20100906-5-JP-2022.00089.

[11] T. S. Tuli, N. G. Orr, and R. E. Zee, "Low cost ground station design for nanosatellite missions," in Proceedings of AMSAT Symposium, 2006. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.564.3644&rep=rep1&type=pdf.

[12] F. Shamutally, Z. Soreefan, A. Suddhoo, and J. M. Momple, "Practical "low cost" leo receiving ground station," in Proceedings of 2018 IEEE Radio and Antenna Days of the Indian Ocean (RADIO), Oct. 2018, pp. 1–2, ISBN: 978-99949-0-470-9. DOI: 10.23919/RADIO.2018.8572330.

[13] V. Parthasarathy, "Virtual ground station for automated spacecraft operations," University of Manitoba, 2021. [Online]. Available: `http://hdl.handle.net/1993/35224`.

[14] T. Yang, B. Chen, Y. Gao, J. Feng, H. Zhang, and X. Wang, "Data mining-based fault detection and prediction methods for in-orbit satellite," in Proceedings of 2nd International Conference on Measurement, Information and Control, Aug. 2013, pp. 805–808, ISBN: 978-1-4799-1392-3. DOI: `10.1109/MIC.2013.6758085`.

[15] Q. Li, X. Zhou, P. Lin, and S. Li, "Anomaly detection and fault diagnosis technology of spacecraft based on telemetry-mining," in Proceedings of 3rd International Symposium on Systems, Control in Aeronautics, and Astronautics, Jun. 2010, pp. 233–236, ISBN: 978-1-4244-6043-4. DOI: `10.1109/ISSCAA.2010.5633180`.

[16] Y. K. T. Yairi, "Telemetry-mining: A machine learning approach to anomaly detection and fault diagnosis for space systems," in Proceedings of 2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06), 2006, pp. 466–476, ISBN: 0-7695-2644-6. DOI: `10.1109/SMC-IT.2006.79`.

[17] Y. Gao, T. Yang, M. Xu, and N. Xing, "An unsupervised anomaly detection approach for spacecraft based on normal behavior clustering," in Proceedings of 5th International Conference on Intelligent Computation Technology and Automation, Jan. 2012, pp. 478–481, ISBN: 978-1-4673-0470-2. DOI: `10.1109/ICICTA.2012.126`.

[18] S. Abdelghafar, A. Darwish, A. E. Hassanien, M. Yahia, and A. Zaghrout, "Anomaly detection of satellite telemetry based on optimized extreme learning machine," *Journal of Space Safety Engineering*, vol. 6, pp. 291–298, 4 Dec. 2019, ISSN: 2468-8967. DOI: `10.1016/J.JSSE.2019.10.005`.

[19] P. Vaughan, *Reducing the Cost of Spacecraft Ground Systems and Operations*. Springer, 2000, pp. 87–95. DOI: 10.1007/978-94-015-9395-3_12.

[20] D Firestone, R Atkin, C Hooks, *et al.*, "Low-cost, automated ground station for leo mission support," *IEEE Aerospace and Electronic Systems Magazine*, vol. 26, pp. 12–18, 3 Mar. 2011, ISSN: 0885-8985. DOI: 10.1109/MAES.2011.5746180.

[21] S. K. Ibrahim, A. Ahmed, M. A. E. Zeidan, and I. E. Ziedan, "Machine learning techniques for satellite fault diagnosis," *Ain Shams Engineering Journal*, vol. 11, pp. 45–56, 1 Mar. 2020, ISSN: 20904479. DOI: 10.1016/j.asej.2019.08.006.

[22] E. A. Omran and W. A. Murtada, "Efficient anomaly classification for spacecraft reaction wheels," *Neural Computing and Applications*, vol. 31, pp. 2741–2747, 7 Jul. 2019, ISSN: 0941-0643. DOI: 10.1007/s00521-017-3226-y.

[23] W. Bialke and E. Hansell, "A newly discovered branch of the fault tree explaining systemic reaction wheel failures and anomalies," in Proceedings of the European Space Mechanisms and Tribology Symposium, 2017, pp. 20–22.

[24] D. Schor, W. Kinsner, and A. Thoren, "Satellite ground station emulator: An architecture and implementation proposal," presented at the Canadian Conference on Electrical and Computer Engineering, May 2009, pp. 868–873, ISBN: 978-1-4244-3509-8. DOI: 10.1109/CCECE.2009.5090253.

[25] Y. Liu, Y. Chen, Y. Jiao, H. Ma, and T. Wu, "A shared satellite ground station using user-oriented virtualization technology," *IEEE Access*, vol. 8, pp. 63 923–63 934, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2984485.

[26] J. Cutler and C. Kitts, "Mercury: A satellite ground station control system," in Proceedings of IEEE Aerospace Conference, 1999, 51–58 vol.2, ISBN: 0-7803-5425-7. DOI: 10.1109/AERO.1999.793142.

[27] G. Minelli, M. Karpenko, I Ross, *et al.*, "Autonomous operations of large-scale satellite constellations and ground station networks," 2017. [Online]. Available: `http://hdl.handle.net/10945/65798`.

[28] G. Curzi, D. Modenini, and P. Tortora, "Large constellations of small satellites: A survey of near future challenges and missions," *Aerospace*, vol. 7, p. 133, 9 Sep. 2020, ISSN: 2226-4310. DOI: `10.3390/aerospace7090133`.

[29] J. Howard, D. Oza, and D. S. Smith, "Best practices for operations of satellite constellations," Jun. 2006. [Online]. Available: `https://ntrs.nasa.gov/api/citations/20080039173/downloads/20080039173.pdf`.

[30] A. Done, C.-E. Lesanu, A.-M. Cailean, A. Graur, and M. Dimian, "Implementation of an on-line remote control ground station for leo satellites," presented at the 21st International Conference on System Theory, Control and Computing (ICSTCC), Oct. 2017, pp. 855–859, ISBN: 978-1-5386-3842-2. DOI: `10.1109/ICSTCC.2017.8107144`.

[31] L. Dudas, L. Papay, and R. Seller, "Automated and remote controlled ground station of masat-1, the first hungarian satellite," presented at the 24th International Conference Radioelektronika, Apr. 2014, pp. 1–4, ISBN: 978-1-4799-3715-8. DOI: `10.1109/Radioelek.2014.6828410`.

[32] A. Freimann, A. Kleinschrodt, M. Schmidt, and K. Schilling, "Advanced autonomy for low cost ground stations," *IFAC Proceedings Volumes*, vol. 46, pp. 388–392, 2013, ISSN: 14746670. DOI: `10.3182/20130902-5-DE-2040.00054`.

[33] A. H. Ramadan, A. E. Hassanien, H. A. Hefny, and L. F. Ibrahim, *Machine Learning and Data Mining in Aerospace Technology*. Springer, 2020, vol. 836, pp. 3–16, ISBN: 978-3-030-20212-5. DOI: `10.1007/978-3-030-20212-5_1`.

[34] F. Fisher, D. Mutz, T. Estlin, *et al.*, “The past, present, and future of ground station automation within the dsn,” in Proceedings of IEEE Aerospace Conference, 1999, 315–324 vol.1, ISBN: 0-7803-5425-7. DOI: `10.1109/AERO.1999.794304`.

[35] D. Decoste, “Automated learning and monitoring of limit functions,” in Proceedings of International Symposium on Artificial Intelligence, Robotics, and Automation in Space, 1997. [Online]. Available: `https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.2940&rep=rep1&type=pdf`.

[36] J. D. Kleer and B. C. Williams, “Diagnosis with behavioral modes,” vol. 89, IJCAI, 1989, pp. 1324–1330. [Online]. Available: `https://www.vehicular.isy.liu.se/Edu/Courses/DocDiagnos/CourseMaterial/sherlock.pdf`.

[37] F. Hutter and R. Dearden, “The gaussian particle filter for diagnosis of non-linear systems,” *IFAC Proceedings Volumes*, vol. 36, pp. 909–914, 5 Jun. 2003, ISSN: 14746670. DOI: `10.1016/S1474-6670(17)36608-9`.

[38] S. H. Chung, J. M. V. Eepoel, and B. C. Williams, “Improving model-based mode estimation through offline compilation,” In Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space, Jun. 2001. [Online]. Available: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.68.5363&rep=rep1&type=pdf`.

[39] T. Bi, Y. Liu, P. Li, and S. Li, “Telecom customer churn prediction method based on cluster stratified sampling logistic regression,” presented at the International Conference on Software Intelligence Technologies and Applications & International Conference on Frontiers of Internet of Things, 2014, pp. 282–287, ISBN: 978-1-84919-970-4. DOI: `10.1049/cp.2014.1576`.

[40] Y. han Wang, Y. Ou, X. dong Deng, L. ran Zhao, and C. yu Zhang, "The ship collision accidents based on logistic regression and big data," IEEE, Jun. 2019, pp. 4438–4440, ISBN: 978-1-7281-0106-4. DOI: `10.1109/CCDC.2019.8832686`.

[41] T. Kim and J. Park, "Logistic regression for ldpc decoding failure prediction," presented at the IEEE 18th Annual Consumer Communications Networking Conference (CCNC), Jan. 2021, pp. 1–6, ISBN: 978-1-7281-9794-4. DOI: `10.1109/CCNC49032.2021.9369532`.

[42] R. Serban, A. Kupraszewicz, and G. Hu, "Predicting the characteristics of people living in the south usa using logistic regression and decision tree," presented at the 9th IEEE International Conference on Industrial Informatics, Jul. 2011, pp. 688–693, ISBN: 978-1-4577-0435-2. DOI: `10.1109/INDIN.2011.6034974`.

[43] M. P. LaValley, "Logistic regression," *Circulation*, vol. 117, pp. 2395–2399, 18 May 2008, ISSN: 0009-7322. DOI: `10.1161/CIRCULATIONAHA.106.682658`.

[44] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, pp. 1443–1471, 7 Jul. 2001, ISSN: 0899-7667. DOI: `10.1162/089976601750264965`.

[45] L. M. Manevitz, M. Yousef, N. Cristianini, J. Shawe-Taylor, and B. Williamson, "One-class svms for document classification," *Journal of Machine Learning Research*, vol. 2, pp. 139–154, 2001.

[46] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, pp. 1171–1220, Jun. 2008, ISSN: 00905364. DOI: `10.1214/009053607000000677`.

[47] B. Scholkopf, R. Williamson, A. Smola, J. Shawe-Taylor, J. Platt, and R. Holloway, "Support vector method for novelty detection," vol. 12, in Proceedings of Advances in neural information processing systems, 1999.

[48] J. Zhou, K. Chan, V. Chong, and S. Krishnan, "Extraction of brain tumor from mr images using one-class support vector machine," in Proceedings of IEEE Engineering in Medicine and Biology 27th Annual Conference, 2005, pp. 6411–6414, ISBN: 0-7803-8741-4. DOI: 10.1109/IEMBS.2005.1615965.

[49] J. Santiago-Paz, D. Torres-Roman, A. Figueroa-Ypiña, and J. Argaez-Xool, "Using generalized entropies and oc-svm with mahalanobis kernel for detection and classification of anomalies in network traffic," *Entropy*, vol. 17, pp. 6239–6257, 12 Sep. 2015, ISSN: 1099-4300. DOI: 10.3390/e17096239.

[50] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in Proceedings of the International Joint Conference on Neural Networks, 2003, pp. 1741–1745, ISBN: 0-7803-7898-9. DOI: 10.1109/IJCNN.2003.1223670.

[51] Z. Wang, "Unsupervised structural damage detection and localization using deep learning and machine learning," University of Manitoba, 2021. [Online]. Available: http://hdl.handle.net/1993/35666.

[52] S. Weisberg, *Applied linear regression*. Wiley-Interscience, 2005, vol. 528, p. 310, ISBN: 0-471-66379-4.

[53] A. S. G.A. Lee, *Linear regression analysis*. John Wiley & Sons, 2012.

[54] P. C. Hughes, *Spacecraft Attitude Dynamics*. Courier Corporation, May 2012, pp. 1–129, ISBN: 9780486140131.

[55] F. Philip, "Mech 7780 - spacecraft dynamics and control," 2020, pp. 113–121.

[56] A. O and Dwyer, *PI AND PID CONTROLLER TUNING RULES 3rd Edition*. Imperial College Press, 2009, pp. 4–17.

[57] MSCI, *Msci microwheel 1000*, 2013. [Online]. Available: http://www.reactionwheel.ca/products/MicroWheel-1000.pdf.

[58] F. A. Leve, B. Hamilton, and M. Peck, *Spacecraft Momentum Control Systems.* Springer, Oct. 2015, vol. 1010, pp. 193–200. [Online]. Available: `http://www.springer.com/series/6575`.

[59] MATLAB, *version 9.10.0.1739362 (R2021a) Update 5.* The MathWorks Inc.

[60] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning Data Mining, Inference, and Prediction.* Springer, Aug. 2009, vol. 2, pp. 1–200.

[61] S. Yin, X. Zhu, and C. Jing, "Fault detection based on a robust one class support vector machine," *Neurocomputing*, vol. 145, pp. 263–268, Dec. 2014, ISSN: 09252312. DOI: `10.1016/j.neucom.2014.05.035`.

[62] P. Perera, S. Member, V. M. Patel, and S. Member, "Learning deep features for one-class classification," in Proceedings of IEEE Transactions on Image Processing, 2019, pp. 5450–5463.

[63] V. N. Vapnik, "An overview of statistical learning theory," vol. 10, in Proceedings of IEEE Transactions on Neural Networks, 1999, pp. 988–999.

[64] B Schölkopf, A. Smola, and F Bach, *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002, ISBN: 9780262256933.

[65] Y. Xiao, H. Wang, and W. Xu, "Parameter selection of gaussian kernel for one-class svm," *IEEE Transactions on Cybernetics*, vol. 45, pp. 941–953, 5 May 2015, ISSN: 2168-2267. DOI: `10.1109/TCYB.2014.2340433`.

[66] A. J. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, pp. 637–649, 4 Jun. 1998, ISSN: 08936080. DOI: `10.1016/S0893-6080(98)00032-X`.

[67] C. C. Aggarwal, *Outlier Analysis.* Springer, 2017, pp. 88–95.

[68] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," *Microsoft*, Jul. 1998. [Online]. Available: `https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/`.

[69] J. Jiong and Z. Hao-Ran, "A fast learning algorithm for one-class support vector machine," in Proceedings of the third International Conference on Natural Computation (ICNC 2007), 2007, pp. 19–23. DOI: `10.1109/ICNC.2007.25`.

[70] H. Wang, G. Zhao, and H. Gu, "A fast training method for oc-svm based on the random sampling lemma," in Proceedings of the sixth International Conference on Natural Computation, Aug. 2010, pp. 824–827, ISBN: 978-1-4244-5958-2. DOI: `10.1109/ICNC.2010.5583242`.

[71] P. Dimitri, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, 1997.

[72] A. Dresden, "The fourteenth western meeting of the american mathematical society," *Bulletin of the American Mathematical Society*, vol. 26, pp. 385–397, 9 Jun. 1920, ISSN: 0002-9904. DOI: `10.1090/S0002-9904-1920-03322-7`.

[73] M. Galarnyk, *Explaining the 68-95-99.7 rule for a normal distribution*, Jun. 2018. [Online]. Available: `https://towardsdatascience.com/understanding-the-68-95-99-7-rule-for-a-normal-distribution-b7b7cbf760c2`.

[74] J. N. Mandrekar, "Receiver operating characteristic curve in diagnostic test assessment," *Journal of Thoracic Oncology*, vol. 5, pp. 1315–1316, 9 Sep. 2010, ISSN: 15560864. DOI: `10.1097/JTO.0b013e3181ec173d`.

Appendix A

Author Copyright Permission

This appendix gives the evidence that the permission to re-use the contents and results from previously published papers.

A.0.1 Copy Right from AIAA

```
-------------------------------------------------------
From: Yujia Huang <huangy25@myumanitoba.ca>
Sent: Wednesday, February 17, 2021 1:02 PM
To: Heather Brennan <HeatherB@aiaa.org>
Subject: Re: Questions about re-use published paper
```

Hi Heather,

Thank you so much for the reply.

I will add the acknowledge agreement at my thesis footnote then.

Thank you again!

Sincerely, Yujia

From: Heather Brennan <HeatherB@aiaa.org>

Sent: Wednesday, February 17, 2021 12:59 PM

To: Yujia Huang <huangy25@myumanitoba.ca>

Subject: RE: Questions about re-use published paper

If you retained the copyright, then you do not need to seek permission to republish the work. (You would essentially be asking for permission from yourself.)

Although permission to reuse the paper in your thesis isn't required, you do need to cite and acknowledge the prior publication as part of your thesis. In a footnote or directly within the text, acknowledge the section/chapter of your thesis that is based on the previously published work and fully cite your conference paper in your reference work.

Heather

From: Yujia Huang <huangy25@myumanitoba.ca>

Sent: Wednesday, February 17, 2021 1:49 PM

To: Heather Brennan <HeatherB@aiaa.org>

Subject: Re: Questions about re-use published paper

Hi Heather,

It is me again. Sorry to bother you again.

I tried to search my paper through the website while my paper was not able to be found. Also, I noticed that the copyright of the paper is under my name (I have attached the capture), is this still applicable for the permission?

I am wondering if I could get some help on these? The DOI link of my paper is: https://arc.aiaa.org/doi/abs/10.2514/6.2021-0665

Thank you very much for your time.

Sincerely,

Yujia

From: Heather Brennan <HeatherB@aiaa.org>

Sent: Wednesday, February 17, 2021 7:56 AM

To: Yujia Huang <huangy25@myumanitoba.ca>

Subject: FW: Questions about re-use published paper

Dear Yujia,

Thank you for your inquiry. If you transferred copyright of your paper to AIAA, you should submit your reprint request to Copyright Clearance Center

www.copyright.com. Enter the conference name to begin your transaction, not the title of your paper. Publication titles and copyright status for all AIAA publications can be confirmed by visiting our electronic library, Aerospace Research Central, at https://arc.aiaa.org.

Once you retrieve the publication record, you will have the opportunity to provide the specific details of your request. When responding to, "Who will republish the content?" be sure to indicate that you are the author of the requested content, and also indicate that the Type of Us is for a thesis or dissertation; this will ensure that no fee is assessed.

Additional guidance can be found on AIAA's Rights and Permissions page:

http://www.aiaa.org/rightsandpermissions/

If you have trouble locating the appropriate record in the CCC database, please let me know. The data for SciTech 2021 has been submitted to CCC, but this is a very recent publication and sometimes updates are slow to show up in the database.

Sincerely,

Heather A. Brennan Managing Director, Publications

American Institute of Aeronautics and Astronautics

www.aiaa.org

12700 Sunrise Valley Drive, Suite 200

Reston, VA 20191-5807

800-639-AIAA (2422)

heatherb@aiaa.org 703.264.7568 (direct)

```
-------------------------------------------------------
From: Ann Ames <AnnA@aiaa.org>
Sent: Wednesday, February 17, 2021 8:05 AM
To: Yujia Huang <huangy25@myumanitoba.ca>
Cc: Heather Brennan <HeatherB@aiaa.org>
Subject: RE: Questions about re-use published paper
```

Heather Brennan, copied, will assist you.

Ann Ames Manager, Technical Program

American Institute of Aeronautics and Astronautics www.aiaa.org 12700 Sunrise Valley Drive, Suite 200 Reston, VA 20191-5807 800-639-AIAA (2422) AnnA@aiaa.org 703-264-7549 (direct)

```
-------------------------------------------------------
From: Yujia Huang <huangy25@myumanitoba.ca>
Sent: Tuesday, February 16, 2021 6:05 PM
To: Ann Ames <AnnA@aiaa.org>
Subject: Questions about re-use published paper
```

Hi Ann,

Hope you are doing well.

I was a former presenter in 2021 AIAA forum and I also published my paper as well.

I am wondering if I could ask for a permission to re-use my published content in my MSc thesis?

Thank you very much.

Sincerely, Yujia

A.0.2 Copy Right from Small Satellite

```
-------------------------------------------------------
From: Yujia Huang
Sent: Thursday, October 28, 2021 8:33 PM
To: Michelle Howa <Michelle.Howa@sdl.usu.edu>
Subject: RE: Questions about re-use published paper
```

Thank you so much Michelle!

Hope you are all doing well!

Sincerely,

Yujia

From: Michelle Howa <Michelle.Howa@sdl.usu.edu>

Sent: October 28, 2021 8:32 PM

To: Yujia Huang <huangy25@myumanitoba.ca>

Subject: Re: Questions about re-use published paper

Hello Yujia,

We do not hold copyright for your work. You are welcome to do whatever you want with it. I wish you well with your thesis.

Sincerely,

Michelle Howa

From: Yujia Huang

Sent: Wednesday, October 20, 2021 11:20 AM

To: Michelle Howa <Michelle.Howa@sdl.usu.edu>

Subject: Questions about re-use published paper

Hi Michelle,

Hope you are doing well.

I was a former presenter in 2021 SmallSat poster session with a published technical paper.

I am wondering if I could ask for a permission to re-use my published content in my MSc thesis?

Thank you!

Sincerely,

Yujia