University of Windsor

Scholarship at UWindsor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

4-6-2021

A Data-driven Fault Isolation and Identification Scheme for Multiple In-Phase Faults in Satellite Control Moment Gyros

Hossein Varvani Farahani
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation

Varvani Farahani, Hossein, "A Data-driven Fault Isolation and Identification Scheme for Multiple In-Phase Faults in Satellite Control Moment Gyros" (2021). *Electronic Theses and Dissertations*. 8586.
https://scholar.uwindsor.ca/etd/8586

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

A Data-driven Fault Isolation and Identification Scheme for Multiple In-Phase Faults in Satellite Control Moment Gyros

By

Hossein Varvani Farahani

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Mechanical, Automotive, and Materials Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada

2021

©2021 Hossein Varvani Farahani

A Data-driven Fault Isolation and Identification Scheme for Multiple In-Phase Faults in Satellite Control Moment Gyros

By

Hossein Varvani Farahani

APPROVED BY:

R. Razavi-Far
Department of Electrical and Computer Engineering

J. Ahamed
Department of Mechanical, Automotive, and Materials Engineering

---

A. Rahimi, Advisor
Department of Mechanical, Automotive, and Materials Engineering

March 3, 2021

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATIONS

I. Co-Authorship

I hereby declare that this thesis incorporates material that is the outcome of my research under the supervision of Dr. Afshin Rahimi who is also the co-author of the papers published through this work. In all cases including the thesis and the papers, the problem definition and the satellite simulator used for generating the raw data come from the previous works done by Dr. Rahimi. The primary contributions/novelties, data analysis, interpretation, and writing were performed by the author and the contribution of the co-author by providing supervision, feedback, and comments.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

II. Previous Publication

This thesis partly includes the original papers that have been previously submitted or published in peer-reviewed journals and conferences, as provided in the following table:

Thesis Chapter	Publication Title/Full Citation	Publication Status
1-6	Varvani Farahani, H., and Rahimi, A. “Fault Diagnosis of Control Moment Gyroscope Using Optimized Support Vector Machine.” 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2020, pp. 3111–3116. https://doi.org/10.1109/SMC42975.2020.9283402	Published
1-6	Varvani Farahani, H., and Rahimi, A. “An Optimized Support Vector Regression for Identification of In-phase Faults in Control Moment Gyroscope Assembly.” 2021 IEEE International Conference on Prognostics and Health Management (ICPHM 2021)	Submitted
1-6	Varvani Farahani, H., and Rahimi, A. “Data-driven Fault Diagnosis for Satellite Control Moment Gyro Assembly with Multiple In-Phase Faults.” Journal Paper	Submitted

I certify that I have obtained a written permission from the copyright owner(s) to include the above-published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

III. General

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

A satellite can only complete its mission successfully when all its subsystems, including the attitude control subsystem, are in healthy condition and work properly. Control moment gyroscope is a type of actuator used in the attitude control subsystems of satellites. Any fault in the control moment gyroscope can cause the satellite mission failure if it is not detected, isolated and resolved in-time. Fault isolation provides an opportunity to detect and isolate the occurring faults and, if accompanied by proactive remedial actions, can avoid failure and improve the satellite reliability. It is also necessary to know the fault severity for better maintenance planning and prioritize the corrective actions. This way, the more severe faults can be corrected first.

In this work, an enhanced data-driven fault diagnosis scheme is introduced for fault isolation and identification of multiple in-phase faults of satellite control moment gyroscopes that is not addressed in the literature before with high accuracy. The proposed method is based on an optimized support vector machine and an optimized support vector regressor. The results yield fault predictions with up to 95.6% accuracy for isolation and 94.9% accuracy for identification, on average. In addition, a sensitivity analysis with regards to noise, missing values, and missing sensors is done where the results show that the proposed model is robust enough to be used in real applications.

DEDICATION

I would like to dedicate my thesis to my dear parents for their unconditional love and kindness, which makes me more powerful and confident every day. To my lovely wife, Zahra, whom I can never be successful without her support, patience, encouragement. Finally, to our dear daughter, Delaram, who makes our life meaningful.

ACKNOWLEDGEMENTS

This work is part of the research conducted at the Intelligent Control, Analytics, and Modeling (iCAM) lab, in the Department of Mechanical, Automotive, and Materials Engineering at the University of Windsor. I would like to express my sincere gratitude to my supervisor, Dr. Afshin Rahimi, for his patience, motivation, immense knowledge, and financial support throughout my master's study. He supported me in conducting research, collaborating and improving in the academic environment, and gave me the area to be creative and work toward different approaches to reach the goal. I want to thank him for all of the opportunities I was given to conduct my research and further my thesis. It was such an honour for me to know you and be part of your research team.

I would also like to thank my thesis committee members, Dr. Roozbeh Razavi-Far, from the Electrical and Computer Engineering Department, and Dr. Jalal Ahamed, from the Department of Mechanical, Automotive, and Materials Engineering for taking the time out of their busy schedule to participate in reviewing my proposal and thesis and providing insightful comments to improve this work. Without their participation and support, this thesis could not have been successfully conducted.

TABLE OF CONTENTS

Declaration of Co-authorship / Previous Publications iii

Abstract vi

Dedication vii

Acknowledgements viii

List of Tables xi

List of Figures xii

List of Appendices xiii

List of Abbreviations/Symbols xiv

Nomenclature xv

1 Chapter 1 Introduction 1

1.1 Introduction 1

1.2 Literature Review 1

1.3 Contributions of This Work 3

1.4 Problem Definition 4

1.5 Outline 5

2 Chapter 2 Theoretical Background 7

2.1 Feature Engineering 7

2.2 Machine Learning 11

2.3 Algorithm Complexity 15

3 Chapter 3 Methodology 17

3.1 The Proposed Scheme Outline 17

3.2 Acquiring the Data 18

3.3 Data Preprocessing 18

3.4 Machine Learning Model Selection 19

3.5 Training, Testing, and Tuning the Models 19

3.6 Predicting Outcomes 19

4 Chapter 4 Case Study: A Satellite with Four CMGs 20

4.1 Introduction 20

4.2 Satellite Dynamics and Kinematics 20

4.3 Controller and Steering Logic 21

4.4 Actuators 24

4.5 Raw Data 24

4.6 Feature Engineering 27

4.7 Machine Learning Models 30

5 Chapter 5 Results and Discussion 32

5.1 Fault Isolation Results 32

5.2 Fault Identification Results 41

6 Chapter 6 Conclusions and Future Works 46

6.1 Conclusions 46

6.2 Future Works 46

References 47

Appendices 54

Appendix A: Copyright Permissions 54

Vita Auctoris 55

LIST OF TABLES

Table 2.1 - Features calculated from time domain and time-frequency domain 8
Table 2.2 - Features calculated from the frequency domain....................................8
Table 2.2 - Time complexity for different methods...16
Table 3.1 - Different scenarios for faults in the CMG assembly 18
Table 4.1 - Required Inputs for the Satellite Simulator..25
Table 5.1 - Comparison of different feature extraction methods............................32
Table 5.2 - Comparison of different feature reduction/selection methods 33
Table 5.3 - Comparison of machine learning models..34
Table 5.4 - Comparison of different classification approaches 34
Table 5.5 - The SVM optimization grid search 35
Table 5.6 - The results for five-folds cross-validation..36
Table 5.7 - The case study confusion matrix 37
Table 5.8 - Model’s sensitivity to the number of active faults 38
Table 5.9 - Model’s sensitivity to noise...39
Table 5.10 - Model’s sensitivity to missing sensors..40
Table 5.11 - Model’s sensitivity to missing values...41
Table 5.12 - The SVR Model Performance for Different Feature Sets 41
Table 5.13 - The SVR optimization grid search 42
Table 5.14 - The regression analysis results for five-folds cross-validation 43
Table 5.15 - MSE of regression analysis results..43

LIST OF FIGURES

Figure 2.1 - Classification by support vector machines..12

Figure 2.2 - Regression analysis by support vector regressors..............................13

Figure 2.3 - Architecture of BPNN with two hidden layers 14

Figure 3.1 - The proposed fault isolation and identification platform outline........17

Figure 4.1 - CMG assembly in pyramid configuration (a) isometric (b) top views20

Figure 4.2 - Satellite simulation setup 20

Figure 4.3 - Sample raw data 26

Figure 4.4 - Sample residual data for the satellite quaternions, (a) the whole simulation period (b) zoomed around the fault period...27

Figure 4.5 - Sample residual data for the satellite angular speed, (a) the whole simulation period (b) zoomed around the fault period...28

Figure 4.6 - Sample residual data for the CMGs gimbal angle, (a) the whole simulation period (b) zoomed around the fault period...29

Figure 5.1 - Features Explained Variance...33

Figure 5.2 - SVM validation curves (a) score vs. gamma (b) score vs. C 35

Figure 5.3 - The learning curve for the SVM model 36

Figure 5.4 - Features explained variance for the SVR model................................41

Figure 5.5 - Validation curve of the SVR model for gamma.................................42

Figure 5.6 - Validation curve of the SVR model for C..42

Figure 5.7 - The SVR model score versus fault scenarios for f1............................44

Figure 5.8 - The SVR model score versus fault scenarios for f2............................44

Figure 5.9 - The SVR model score versus fault scenarios for f3............................44

Figure 5.10 - The SVR model score versus fault scenarios for f4..........................45

LIST OF APPENDICES

Appendix A: Copyright Permissions..54

LIST OF ABBREVIATIONS/SYMBOLS

CMG	Control Moment Gyroscope
RW	Reaction Wheel
ACS	Attitude Control System
FF	Frequency-domain Feature
VMD	Variational Mode Decomposition
EEMD	Ensemble Empirical Mode Decomposition
LSTM	Long Short-Term Memory
WPT	Wavelet Packet Transform
PCA	Principal Component Analysis
NPCA	Non-linear Principal Component Analysis
ANN	Artificial Neural Networks
BPNN	Backpropagation Neural Network
SVM	Support Vector Machines
PSO	Particle Swarm Optimization
FDI	Fault Detection and Isolation

NOMENCLATURE

X	state vector
u	control input
θ	system parameter
y	measurement
ω^X	additive process noise for states
ω^θ	additive process noise for parameters
v	additive measurement noise
f, g	generic non-linear functions
θ_0	nominal parameter value
α	fault parameter value
O	time complexity function
n	number of training samples
p	number of features
n_{sv}	support vector numbers
n_{trees}	number of trees
n_{epoch}	number of epoches
n_l	number of neurons
ω	satellite angular speed
τ_e	external force
H	angular momentum
J	moment of inertia
J_w	axial moment of inertia
A	mapping matrix
h	angular moment
q	quaternion
$\bar{e}$	principal axis
Φ	principal angle
I	identity matrix
C	rotating matrix
σ	sliding manifold
λ	sliding gain
sgn	sign function
p_0	a positive constant
δ	CMG gimbal angle
Ω	flywheels angular speed
c	cosine
s	sine
r	residual

ρ	Pearson correlation coefficient
COV	covariance
VAR	variance
N	number of instances
R	multi correlation coefficient
z	FFT spectrum
M	number of spectrum lines
ρ_{rank}	Spearman rank correlation coefficient
d	difference between ranks
$\boldsymbol{n}$	number of observations
τ	Kendall rank correlation coefficient
P	number of concordant pairs
Q	number of discordant pairs
H	support vector
w	weight
b	bias
γ	margin
$\boldsymbol{f}, \boldsymbol{g}$	hyperplane functions
ε	maximum deviation
σ	activation function
E	error
η	learning rate

Subscripts

k	time step
i	layer number
s	satellite
CMG	control moment gyro
d	desired
e	error
r	required
n	number of CMGs
q	fault scenario
m	spectrum line
j	data set instance
h	layer number
p, o	neuron number

CHAPTER 1

INTRODUCTION

1.1 Introduction

Satellites are essential assets for space exploration and data collection. Their fault-free operation is critical, which relies on the health of their subsystems and components. One of the major systems of any satellite is the attitude control subsystem (ACS) that uses different actuators such as reaction wheels, momentum wheels, control moment gyros (CMGs), among others. If the ACS fails, the satellite cannot complete its mission. Hence, if a fault occurs in any part of the CMGs, it may lead to a failure if unattended. Establishing and running an asset health management platform can help prevent failure and improving satellite reliability.

An asset health management platform has two major elements: Fault isolation and fault severity identification. Fault isolation determines the required corrective actions, while fault severity identification helps optimize maintenance planning through prioritizing maintenance actions. Hence, fault isolation and identification of CMG are promising methods for preventing failures that can affect the satellite's reliability.

There are different fault isolation approaches, including the model-based and data-driven categories [1–3]. The application of different data-driven methods in fault isolation is a trend these days. Specifically, different machine learning methods, such as support vector machines (SVM), neural networks, and gradient boosting machines, along with different deep learning methods, are widely used for this application [2,4]. These methods are used to establish a fault isolation scheme by classifying given data to distinguish between different possible faults.

Fault severity identification can include different tasks such as fault degradation estimation and fault severity classification, and different methods can be used for data-driven fault severity assessment, including classification and regression analysis [5].

1.2 Literature Review

Several research pieces are done on the fault diagnosis of a satellite's ACS using a SVM, out of other data-driven approaches [6–9]. The SVM is a supervised learning method with reasonable flexibility and can adapt to any application. As each fault scenario can be considered as a class, the SVM can be used for fault diagnosis. In [6], a multi-classifier model is formed based on the combination of the Dempster-Shafer theory and SVM, while nonlinear principal component analysis (NPCA) is adopted for reducing the feature size. In [8], SVM and neural networks are used to build a model for the satellite power supply system's health monitoring. In [7], the combination of random forest, partial least square, SVM, and Naïve Bayes, is used to form a framework for detecting and isolating faults. In [9], Telemetry data is used as input to extract the features. Then, PCA is used for feature reduction, followed by an optimized SVM model using the particle swarm optimization (PSO) method adopted for FDI.

Neural networks and deep learning methods are also employed for satellite ACS FDI [10–13]. Prony analysis is used for feature extraction, and a feed-forward neural network is developed for anomaly classification [10]. A model is established to find the characteristics that express the faults using a deep neural network. Next, the fault-to-noise ratio and characteristics differences are amplified using a sliding window. The proposed method is then used for fault identification of a satellite ACS [12]. A feed-forward wavelet-based neural network is adopted to form an adaptive observer for fault detection. Adopting a feed-forward wavelet-based neural network with single hidden layer, the proposed method can be applied to non-linear systems [13].

Various other machine learning approaches such as minimum error minimax probability machine [14], gradient boosting machines [15], and kernel principal component analysis [16] are used for fault detection and isolation in aerospace applications.

Different methods can be used for data-driven fault severity assessment, including classification and regression analysis [5]. In [17], a combination of ensemble empirical mode decomposition (EEMD) and correlation analysis is used for data pre-processing. Multi-domain features are extracted, and an optimized support vector regression model is adopted to identify fault sizes of an aviation pump. In [18], a combination of variational mode decomposition (VMD) for feature extraction and long short-term memory (LSTM) neural network for regression analysis is adopted for predicting the severity of faults in rolling element bearings. SVM and artificial neural network (ANN) are used to develop models to classify multi-severity faults in rolling element bearings [19]. Multiple regression models are used to study the relationships between the test points, condition indicators and fault severities in [20]. The optimal condition indicators are selected for identifying the fault severity for rolling element bearings. In [21], time series segmentation is used to determine the fault severity for each input data instance. An SVM model is developed for fault-severity classification and used for point machine sliding-chair health assessment. A deep convolutional neural network pre-trained by a stacked convolutional autoencoder is developed in [22]. The developed model is then used to automatically extract the features for a helical gearbox's fault severity assessment. In [23], different machine learning algorithms, including random forest and gradient boosting machines, are used for severity identification of the satellite CMGs through regression analysis.

While several research pieces exist on ACS fault isolation, most focus on systems that only have one active fault. The proposed models cannot handle cases with multiple in-phase faults, while these cases are likely to occur during a real-life satellite operation. When there is more than one fault present simultaneously, the cross-effect of each other's faults makes the isolation more challenging. The only works that have evaluated the multiple in-phase faults [17] reported a maximum accuracy of 66.6% for fault isolation and 79.9% for fault identification. These scores are not sufficient for use in real applications. Thus, there is a need for a specific approach to handle this problem while achieving a reasonable accuracy for both tasks. In this work, a new data-driven scheme is developed for fault isolation and identification that can handle multiple in-phase faults in satellite CMGs to address the above-mentioned shortcomings in the literature.

1.3 Contributions of This Work

Contributions of this work are listed and described below:

[CONTRIBUTION 1] New feature set is found that best adapts the problem while having the minimum possible size. Using appropriate features is one of the most critical steps of any data-driven approach. This project aims at covering multiple in-phase faults. Based on the assessments done, using common feature sets such as time series statistical measures is not the right approach and does not provide satisfactory results. There was a need to find or develop some new features that could meaningfully represent the system with multiple in-phase faults for effective fault diagnosis.

[CONTRIBUTION 2] The best machine learning method or strategy is found to overcome the inherent complexity of the problem under study. This will be a properly engineered and structured learning platform that best suits the defined problem and could achieve acceptable accuracy.

[CONTRIBUTION 3] The best-reported accuracies for a similar problem are 66.6% for fault isolation and 79.9% for fault identification [17]. This work has improved these scores to 95.9% and 94.9%, respectively.

1.4 Problem Definition

Any non-linear dynamical system can, in general, be modelled in state-space as:

$$\Omega: \begin{cases} X_{k+1} = f(X_k, u_k, \theta_k, \omega_k^X) \\ \theta_{k+1} = \theta_k + \omega_k^\theta \\ y_k = g(X_k, \theta_k) + v_k \end{cases} \tag{1}$$

where $X_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^m$ is the control input, $\theta_k \in \mathbb{R}^l$ is the system parameter, $y_k \in \mathbb{R}^m$ is the measurement, $\omega_k^X,\ \omega_k^\theta \in \mathbb{R}^n$ are the additive process noise for states and parameters, respectively. $v_k \in \mathbb{R}^m$ is the additive measurement noise, k is the time step and finally, the process and measurement models are represented by $f(\cdot)$ and $g(\cdot)$, respectively, which are non-linear functions.

Assuming that any change in the physical parameters of the satellite is accompanied by a change in one of the parameters of the system [25], a fault isolation problem can be expressed as:

$$\theta_k = \theta_0 + \alpha_k \tag{2}$$

where $\theta_0 \in \mathbb{R}^L$ is a vector demonstrating the nominal parameter values, $\alpha_k \in \mathbb{R}^L$ is a vector representing the parameter values in the presence of fault, and L is the number of possible scenarios for faults that can be considered for the satellite. Eq. (2) is a demonstration of a multi-parameter model and can be split into L single parameter models as [25]:

$$\Omega_i: \{\theta_k^i = \theta_0^i + \alpha_k^i \qquad i = 1, \ldots, L \tag{3}$$

Eq. (3) is an expression of a classification problem with L classes for which a data-driven approach can be used to find a solution for.

The data-driven algorithm first aims to classify the current system state while the number of classes is L. This classification problem is solved when a fault is detected by each single-parameter fault model defined in Eq. (3), where the ith model has system parameter θ_k^i and fault severity α_k^i. Then, another data-driven algorithm is employed to identify the fault severity, α_k^i , through regression analysis. It is clear that using Eq. (2), one would be able to extract the severity of the fault by rearranging the terms in Eq. (2) as:

$$\alpha_k = \theta_k - \theta_0 \tag{4}$$

Hence, the problem of fault isolation for a given non-linear system in Eq. (1) with fault formulation in Eqs. (2) and (3) can be formally expressed.

The assumptions made in this work are:

1. The induced faults are in-phase. Each data instance has assigned fault inception and duration times, which are the same for all of the CMG units that are faulty.
2. The assigned fault severity for each instance is from 0 to 1 to cover all of the possible fault severities.
3. All of the state measurements are available.
4. There is no source of noise nor missing values in the raw input data.

This work aims to design and develop a data-driven fault isolation and identification scheme that can use the system outputs and predicts the presence of any possible fault, isolate the fault location, and identify the fault severity under the assumptions mentioned above.

1.5 Outline

The subsequent chapters of this thesis are structured as follows:

In Chapter 2, theoretical backgrounds are explored and includes explanations about the techniques and methods used in this thesis. In Chapter3, the proposed fault isolation and identification scheme is introduced and described. In Chapter 4, a case study is presented to assess the proposed scheme’s performance. Results are presented and discussed in Chapter 5 and Chapter 6 concludes the thesis with final remarks and recommendations for future work.

CHAPTER 2

THEORETICAL BACKGROUND

In this chapter, some theoretical backgrounds are provided that are necessary to support the proposed scheme for fault isolation and identification.

2.1 Feature Engineering

One of the major steps in any data-driven approach is feature engineering. Feature engineering covers a vast list of actions that includes extraction, selection and reduction of features from the pre-processed raw data [26]. Without meaningful and right chosen features, it will be unlikely to achieve reasonable results in machine learning applications. This is because the machine learning models see the data through the provided features, and it will not work well when the features cannot appropriately represent the data and the desired hidden pattern in it. There is a wide range of methods available for use in feature engineering. The best ones can be chosen based on the characteristics of the problem and the data at hand. Another issue in data-driven approaches is time complexity, which depends on the machine learning algorithms and the feature set size. Dealing with large feature sets is time-consuming.

Moreover, large feature sets may lead to less accurate results as the irrelevant portions of the features can mislead the machine learning algorithm. Different methods are proposed for feature selection and reduction to reduce the time complexity and provide the most relevant and meaningful features. In this section, some of the methods that are used in this work are introduced and discussed.

2.1.1 Feature Extraction based on the Multi-domain Analysis

One of the most holistic and comprehensive approaches for obtaining features from time-series data is multi-domain feature extraction [27,28]. In which features are extracted in time, frequency and time-frequency domains. Each of these domains represents the data from a different point of view. So, collecting them together helps to get a comprehensive insight into the data while dealing with only one unique feature set.

Time-domain features are mainly statistical features directly obtained from time series. Table 2.1 shows some of the time domain features used in this work.

Table 2.1 - Features calculated from time domain and time-frequency domain

Statistical Features		
Root mean square Standard deviation Variance Mean Peak to peak	Energy Root sum of squares Kurtosis Crest factor Skewness Peak	Median Sum 3rd moment 4th moment 5th moment

For obtaining frequency domain features, one should perform Fast Fourier Transform (FFT) on the data and then calculate the desired features, which are statistical representatives of the data in the frequency domain. Table 2.2 shows some of the frequency domain features used in this work [28]. In this table, $z(m)$ is the FFT spectrum of the given time series, M is the number of spectrum lines in FFT spectrum, and f_m is the frequency value corresponding to the m_{th} spectrum line.

Table 2.2 - Features calculated from the frequency domain

$FF_1 = \frac{\sum_{m=1}^{M} z(m)}{M}$	$FF_6 = \sqrt{\frac{\sum_{m=1}^{M}[(f_m - FF_5)^2 z(m)]}{M}}$	$FF_{10} = \frac{FF_6}{FF_5}$
$FF_2 = \frac{\sum_{m=1}^{M}[z(m) - FF_1]^2}{M-1}$	$FF_7 = \sqrt{\frac{\sum_{m=1}^{M}\left[{f_m}^2 z(m)\right]}{\sum_{m=1}^{M} z(m)}}$	$FF_{11} = \frac{\sum_{m=1}^{M}[(f_m - FF_5)^3 z(m)]}{M(FF_6)^3}$
$FF_3 = \frac{\sum_{m=1}^{M}[z(m) - FF_1]^3}{M-1}$	$FF_8 = \sqrt{\frac{\sum_{m=1}^{M}\left[{f_m}^4 z(m)\right]}{\sum_{m=1}^{M}\left[{f_m}^2 z(m)\right]}}$	$FF_{12} = \frac{\sum_{m=1}^{M}[(f_m - FF_5)^4 z(m)]}{M(FF_6)^4}$
$FF_4 = \frac{\sum_{m=1}^{M}[z(m) - FF_1]^4}{M(FF_2)^2}$	$FF_9 = \frac{\sum_{m=1}^{M}({f_m}^2 z(m))}{\sqrt{\left[\sum_{m=1}^{M}({f_m}^4 z(m))\right]\left[\sum_{m=1}^{M} z(m)\right]}}$	$FF_{13} = \frac{\sum_{m=1}^{M}\left[\sqrt{\lvert f_m - FF_5}\rvert z(m)\right]}{M\sqrt{FF_6}}$
$FF_5 = \frac{\sum_{m=1}^{M}(f_k z(m))}{\sum_{m=1}^{M} z(m)}$		

Time-frequency analysis methods such as variational mode decomposition (VMD) [28–30] or wavelet packet transform (WPT) [31,32] should be performed on the data, followed by some statistical calculations for extracting features in the time-frequency domain. One of the unique advantages of time-frequency analysis is looking for small transient events that can easily be lost in time or frequency domain analysis. Table 2.1 shows the time-frequency domain features used in this work. These features are calculated from the WPT coefficients of the raw data.

2.1.2 Feature Extraction based on the Correlation Analysis

Wherever multiple parameters interact with each other or are affected by each other, there is room for correlation analysis. For example, for a satellite ACS with four CMGs, there are complicated interactions between the system outputs, which are satellite attitude parameters resulting from multiple faults in the CMGs. In this case, correlation analysis can be used to discover these interactions and provide insights for diagnosing the faults that have resulted in them. In this work, correlation analysis is considered a method for analyzing data and extracting features. Some of the approaches that can be used for correlation analysis in this ground are correlation analysis [33], multi-correlation analysis [34], and cross-correlation analysis [35–38].

Different methods are used for the correlation analysis to discover the relations between each pair of data [33]. The Pearson correlation coefficient is calculated between each pair of the residual data using [39]:

$$\rho_{ij} = \frac{COV_{ij}}{\sqrt{VAR_{ii}VAR_{jj}}} \tag{5}$$

where COV_{ij} is the covariance of r_i and r_j , and r stands for the residual calculated using Eq. (21). COV_{ij} can be calculated using [39]:

$$COV_{ij} = \frac{\sum(r_i - \overline{r_i})\left(r_j - \overline{r_j}\right)}{N} \tag{6}$$

In which, $\overline{r_i}$ and $\overline{r_j}$ are the mean values for r_i and r_j, respectively, and N is the number of instances. VAR_{ii} and VAR_{jj} are the variance of r_i and r_j, respectively. Variacne can be calculated using [39]:

$$VAR_{ii} = \frac{\sum(r_i - \overline{r_i})^2}{N} \tag{7}$$

The Spearman rank correlation coefficient is calculated using [39]:

$$\rho_{rank} = 1 - \frac{6\sum {d_k}^2}{\boldsymbol{n}^3 - \boldsymbol{n}} \tag{8}$$

where $\boldsymbol{n}$ is the number of observations (i.e., length of the residual time series) and d_k is the difference between the ranks of the residuals ${r_i}_k$ and ${r_j}_k$. The Kendall rank correlation coefficient is calculated using [39]:

$$\tau = \frac{2(P - Q)}{\boldsymbol{n}(\boldsymbol{n} - 1)} \tag{9}$$

where P is the number of concordant pairs and Q is the number of discordant pairs. A pair of observations, for example $\left({r_i}_1, {r_j}_1\right)$ and $\left({r_i}_2, {r_j}_2\right)$, is concordant if the point that has a higher ${r_i}_{index}$ also has a higher ${r_j}_{index}$ and vice versa. Therefore, the two datapoints $\left({r_i}_1, {r_j}_1\right)$ and $\left({r_i}_2, {r_j}_2\right)$ are concordant if and only if:

$$sign\left({r_i}_2 - {r_i}_1\right) = sign\left({r_j}_2 - {r_j}_1\right) \tag{10}$$

And are discordant if and only if:

$$sign\left({r_i}_2 - {r_i}_1\right) = -\,sign({r_j}_2 - {r_j}_1) \tag{11}$$

Multi correlation analysis is the same as correlation coefficient calculation, except for it is calculated for each set of three residuals and represents the correlation between the three parameters. It can be calculated using [34]:

$$R_{ijk} = \sqrt{{\rho_{ij}}^2 + {\rho_{jk}}^2 + {\rho_{ik}}^2 - 2\rho_{ij}\rho_{jk}\rho_{ik}} \tag{12}$$

in which ρ is the correlation coefficient as shown in Eq. (5).

Cross-correlation analysis and feature extraction are done based on the method used in [38], and the details are not repeated here. Correlation analysis can also be used for selecting the desired features from a feature pool that is the subject of the next section.

2.1.3 Feature Reduction based on the Principal Component Analysis

Principal component analysis (PCA) is a common method for reducing the number of features while capturing the desired amount of its variance. It extracts orthogonal vectors in sets, known as loading vectors and then calculates the amount of variance known to orthogonal vectors [40].

2.1.4 Feature Selection based on the Importance of the Features

Feature extraction methods usually result in a big pool of features that are not necessarily useful for the problem at hand. So, it is wise to use some methods for finding the most relevant and meaningful features from this pool. One of the approaches that can be used is based on calculating an importance index for each feature and finally choosing the desired number of features with the higher importance index. Implementing this approach, also known as the wrapper model, needs to build a classifier and run it sequentially with different features. This can be done, for example, by starting with only one feature and run the classifier. Then add another feature and rerun it and continue this until all of the features are used. This will be a forward selection approach. Finally, each feature will have an importance index based on whether it has improved or reduced the model performance. Another approach is to start with all of the features and run the model. Then remove one feature and rerun the model and continue this until no features remain. This is known as the backward selection or elimination approach. Assigning the importance index is done based on the negative or positive effect of each feature on the model performance.

2.2 Machine Learning

Machine learning methods are used for classification and regression analysis. In this chapter, some of the methods used in this work are introduced briefly.

2.2.1 Support Vector Machines

Support vector machine (SVM) is a machine learning method based on statistical learning theory. It maps the input data into a high dimensional space called the feature space. Then it finds an optimal hyperplane to separate the two classes while maximizing generalization. The hyperplane is selected to maximize the margin between the plane and the nearest data points of the two classes called the support vectors.

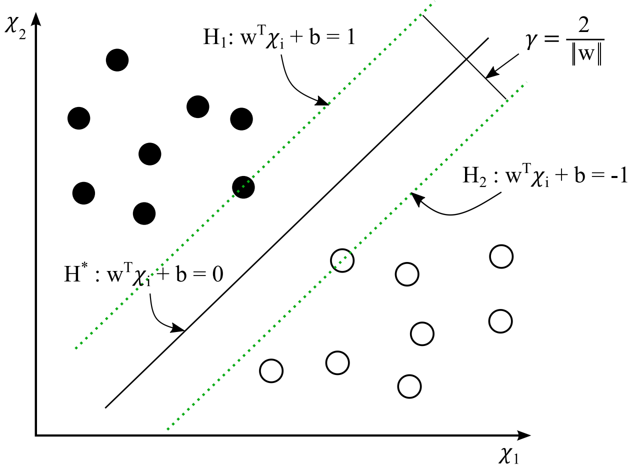


Figure 2.1 - Classification by support vector machines

Given the dataset $\{\chi_j, \mu_j\}_{j=1}^{m}$ with m instances and $\mu_i \in \{-1, 1\}$, a hyperplane $\boldsymbol{f}(\chi) = 0$ is expected to be found to separate the given datasets into two classes, and the hyperplane is shown as:

$$\boldsymbol{f}(\chi) = w^T\chi + b = \sum_{j=1}^{m} w^T\chi_i + b = 0 \tag{13}$$

where the hyperplane is determined by w and b. In order to divide the instances into the positive class and the negative class, the determined hyperplane is subject to:

$$\mu_j\boldsymbol{f}(\chi_j) = \mu_j(w^T\chi_j + b) \geq 1, \quad j = 1,2,\dots,m \tag{14}$$

Support vectors H_1 and H_2 can satisfy the condition given in Eq. (14) as shown in Figure 2.1. The linear SVM is expected to place a hyperplane H^* between the positive and negative datasets, which is orientated by maximizing the margin $\gamma = \frac{2}{\|w\|}$. Therefore, the optimization objective of the linear SVM is shown as follows [2]:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad s.t. \ \ \mu_j(w^T\chi_j + b) \geq 1, \quad j = 1,2,\dots,m \tag{15}$$

2.2.2 Support Vector Regressor

SVM can also be used as a regression analysis method, maintaining all the main features that are the algorithm characteristics. The support vector regressor (SVR) is established on the same principles as the SVM for classification, with a little modification. As the output is a real number, it is difficult to predict the information at hand, which has infinite possibilities. For regression analysis, a margin of tolerance, ε, is set in approximation to the SVM, which would have already requested from the problem. Suppose that the given dataset is $\{\chi_j, \mu_j\}_{j=1}^{m}$ with m instances, the regression analysis aims at finding a function $\boldsymbol{g}(\chi)$ that has the maximum deviation of ε from the targets μ_j for all of the training instances, and at the same time is as flat as possible [41]. This problem can be written as a convex optimization problem:

$$\begin{gathered} \min_{w,b} \frac{1}{2} \|w\|^2 \\ s.t. \;\; \mu_j - (w^T \chi_j + b) \;\leq \varepsilon \;\; and \;\; (w^T \chi_j + b) - \mu_j \;\leq \varepsilon, \;\; j = 1,2, \dots, m \end{gathered} \tag{16}$$

Eq. (16) is based on the assumption that such a function $\boldsymbol{g}(\chi)$ actually exists that approximates all pairs (χ_j, μ_j) with ε precision, or in simple terms, it is possible to solve the convex optimization problem. This optimization problem is also illustrated in Figure 2.2.

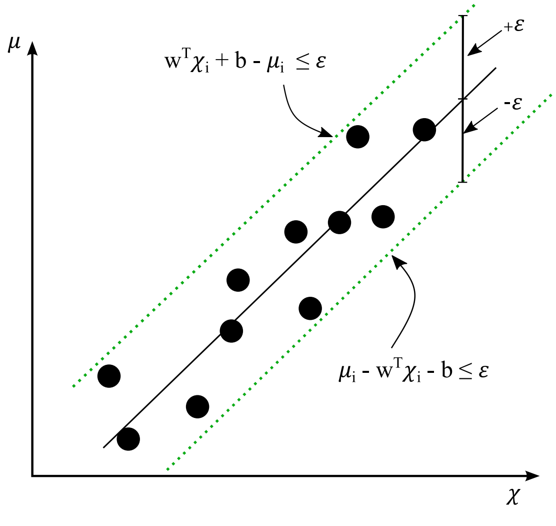

μ
$w^T\chi_i + b - \mu_i \leq \varepsilon$
$+\varepsilon$
$-\varepsilon$
$\mu_i - w^T\chi_i - b \leq \varepsilon$
χ

Figure 2.2 - Regression analysis by support vector regressors

2.2.3 *Artificial Neural Networks*

Artificial neural networks (ANN) imitate the human brain activities in information processing and is a promising approach for building machine learning models. ANN has different implementations, including the backpropagation neural network (BPNN). The BPNN is a multilayer perceptron, which comprises the forward propagation and the backpropagation. In the forward propagation, as shown in Figure 2.3, the multi-hidden layers process the input samples, and finally, the output layer maps them into the target class.

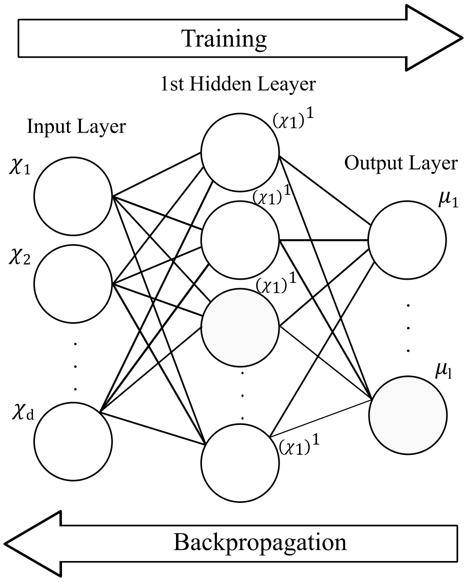


Figure 2.3 - Architecture of BPNN with two hidden layers

Given the training dataset $\{\chi_j, \mu_j\}_{j=1}^{m}$ in which m is the number of samples, $\chi_j \in \mathbb{R}^d$ is the feature set with d features, and $\mu_j \in \mathbb{R}^l$ is the label set with l classes, the output of the h_{th} hidden layer is demonstrated as [2]:

$$(\chi_j^h)_p = \sigma^h \left(\sum_{j=1}^{n_{h-1}} w_p^h . \chi_j^{h-1} + b_p^h \right), \qquad p = 1,2, \dots, n_h, \quad h = 1,2, \dots, H \tag{17}$$

where $(\chi_j^h)_p$ is the output of the p_{th} neuron in the h_{th} hidden layer, and $\chi_j^0 = \chi_j$, n_h is the number of neurons in the h_{th} hidden layer, σ^h expresses the h_{th} hidden layer activation function, n_{h-1} is the number of neurons in the $(h-1)_{th}$ hidden layer, w_k^h is the weights between the neurons in the previous layer and the p_{th} neuron in the h_{th} hidden layer, and b_p^h is the bias of the h_{th} hidden layer. The predicted output of BPNN is [2]:

$$(\hat{\mu}_j)_o = \sigma^{out}\left(\sum_{j=1}^{n_H} w_o^{out}.\chi_j^H + b_o^{out}\right), \qquad o = 1,2, \dots , l \tag{18}$$

where $(\hat{\mu}_j)_o$ is the predicted output of the o_{th} neuron in the output layer, σ^{out} demonstrates the output layer activation function, w_o^{out} and b_o^{out} are the output layer weights and bias, respectively. When given a certain training sample (χ_j, μ_j), the objective of the BPNN optimization process is to minimize the error between the predicted and the target output. This can be expressed as [2]:

$$\min_{w,b} E_j = \frac{1}{2}\sum_{o=1}^{l}\left[\left(\mu_j\right)_o - (\hat{\mu}_j)_o\right]^2 \tag{19}$$

For solving this optimization problem, the training parameters w and b are updated by the gradient descent as follows [2]:

$$w \longleftarrow w - \eta.\frac{\partial E_j}{\partial w}, \qquad b \longleftarrow b - \eta.\frac{\partial E_j}{\partial b} \tag{20}$$

where η expresses the learning rate. The error gradient propagates backward from the output layer to the input layer, and the training parameters are updated layer by layer.

2.3 Algorithm Complexity

Table 2.3 shows the time complexity analysis for the machine learning models used in this work. In this table, n is the number of training samples, p is the feature numbers, n_{sv}, demonstrates support vector numbers, n_{trees} is number of trees, d is the maximum depth of trees, n_{epoch} is the number of epoches, and n_{l_i} is the number of neurons of layer i.

Table 2.3 - Time complexity for different methods

Algorithm	Training Phase	Prediction Phase
Neural Network	$O(npn_{epoch}(n_{l_1}n_{l_2} + n_{l_2}n_{l_3} + n_{l_3}n_{l_4} + \cdots))$	$O(pn_{l_1}(n_{l_1}n_{l_2} + n_{l_2}n_{l_3} + n_{l_3}n_{l_4} + \cdots))$
Support Vector Machine (Kernel)	$O(n^3)$	$O(n_{sv}p)$
Gradient Boosting Machine	$O(ndn_{trees}logn)$	$O(pn_{trees})$
Random Forest	$O(\mathrm{n}pn_{trees}logn)$	$O(pn_{trees})$

Complexity analysis of neural networks is not straightforward. References [42,43] have some details about this. The SVM algorithms include solving the constrained quadratic equation that is equivalent to the calculation of the inversion of an n size square matrix, which has the complexity of $O(n^3)$. In [44], a comprehensive time complexity analysis is done for different steps of implementing an SVM classifier. The time complexity of training with a gradient boosting machine is $O(ndn_{trees}logn)$ and prediction for a new sample takes $O(pn_{trees})$ [45]. Assuming trees are free to grow to maximum height $O(logn)$, training time complexity for random forest is $O(\mathrm{n}pn_{trees}logn)$, and prediction of a new sample takes $O(pn_{trees})$ [46].

CHAPTER 3

METHODOLOGY

In this chapter, the proposed scheme for fault isolation is introduced and described.

3.1 The Proposed Scheme Outline

A data-driven fault diagnosis platform is introduced that is comprised of an optimized machine learning model for isolating and identifying multiple in-phase faults of the satellite CMG. The features are calculated using the CMG data, and then feature reduction is made through the principal component analysis. The chosen features are fed to the machine learning models as inputs for the training and testing steps. For improving the performance, the machine learning models are tuned by finding the optimal values of their parameters. At the last step, the optimized machine learning models can be used to isolate and identify the faults of the CMG. Figure 3.1 shows the flow diagram of the proposed fault diagnosis scheme, and its elements are described in sections 3.2 to 3.6.

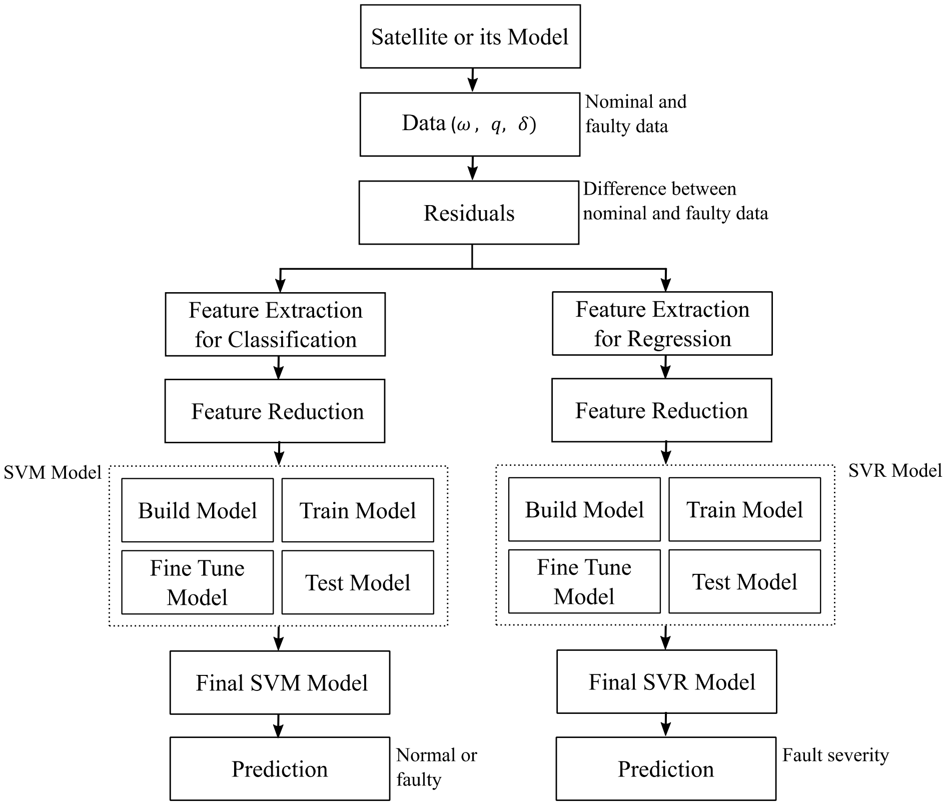


Figure 3.1 - The proposed fault isolation and identification platform outline

3.2 Acquiring the Data

The raw data is acquired from a satellite telemetry system or a satellite mathematical model. In this study, a high-fidelity satellite model with four CMG units is used to generate the required data described in Chapter 4. The raw data comprises satellite attitude quaternions, angular speeds and the CMGs' gimbal angles. The data is stored in a time-series format, with each set representing one of the fault scenarios shown in Table 3.1. There is a total of 16 scenarios. Scenario 0 represents the system without any fault. Scenarios 1 to 15 represent the system with 1, 2, 3 or 4 faulty units.

Table 3.1 - Different scenarios for faults in the CMG assembly

Scenario No.	Faulty CMG(s)	Scenario No.	Faulty CMG(s)
0	—	8	2, 3
1	1	9	2, 4
2	2	10	3, 4
3	3	11	1, 2, 3
4	4	12	1, 2, 4
5	1, 2	13	1, 3, 4
6	1, 3	14	2, 3, 4
7	1, 4	15	1, 2, 3, 4

3.3 Data Preprocessing

The raw data is used to calculate the residuals. Residuals represent the difference between the system outputs in a nominal and faulty condition. The residuals can be calculated using:

$$r_k^q = \theta_k^q - \theta_k^0 \qquad q = 0, \dots, 15 \tag{21}$$

where, θ^q represents the system parameters for faulty state q, q depicts the desired fault scenario, θ^0, is the system parameters for a healthy state, and k is the measurement time step. The features are extracted from the residual time-series. Feature selection/reduction methods are used to reduce the number of the extracted features while looking for the most representative features. There are various methods for feature extraction/reduction/selection that are described in Chapter 2. The chosen feature set is then split into train and test subsets fed into the machine learning model.

3.4 Machine Learning Model Selection

Two different machine learning models are developed to be used for the classification of data and then for regression analysis. There are a variety of methods suitable for machine learning that are described in Chapter 2. Fault scenarios are used as labels, and as each instance of the input feature sets belongs to a specific fault scenario, the developed classification machine learning model aims to predict the true label for every instance of the input feature set and the developed regression machine learning model is used to predict the fault severity. Both are achieved by training the models with the available feature sets with known labels and fault severity and testing and tuning the model.

3.5 Training, Testing, and Tuning the Models

The train portion of the feature sets is used to train the machine learning models. The models are then tested by the test portion of the feature sets, and finally, the optimum values for the models' hyperparameters are obtained through an optimization process to avoid over- or under-fitting.

3.6 Predicting Outcomes

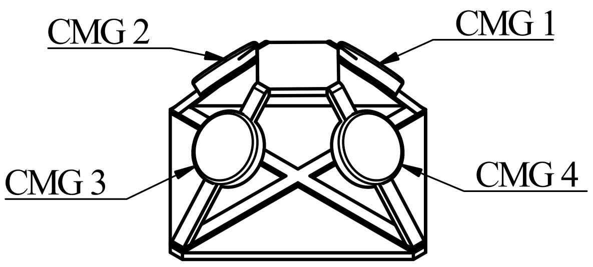
The optimized classification machine learning model can predict the fault scenario label, and the optimized regression machine learning model can predict the fault severity for every new and unseen instance of the data. A case study is presented in the next chapter to evaluate the performance of the proposed scheme.

CHAPTER 4

CASE STUDY: A SATELLITE WITH FOUR CMGS

4.1 Introduction

In this chapter, a satellite with four CMGs is used to evaluate the performance of the proposed fault diagnosis scheme. Figure 4.1 shows the CMGs assembly in a pyramid configuration. A high-fidelity satellite mathematical model and simulator [47] is used in this work. The simulation setup is shown in Figure 4.2. The components of this simulator are described in sections 4.2 to 4.4.



(a) **(b)**

Figure 4.1 - CMG assembly in pyramid configuration (a) isometric (b) top views

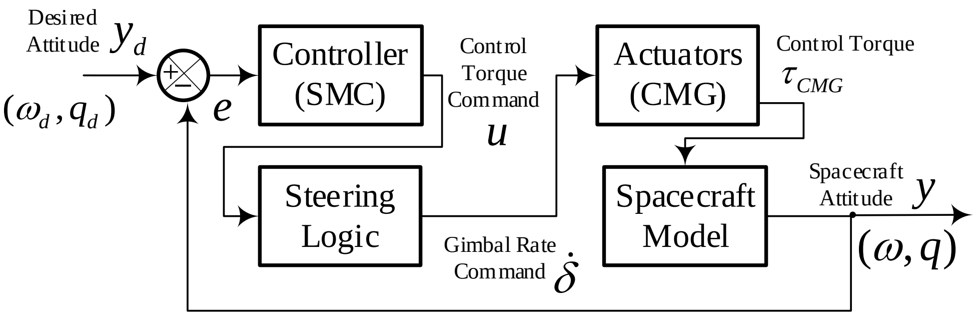


Figure 4.2 - Satellite simulation setup

4.2 Satellite Dynamics and Kinematics

Dynamics and kinematics for the satellite are used to calculate the required outputs from the input control torque. The dynamics equation of a satellite with reaction wheels onboard can be expressed as [47]:

$$\dot{H}_{BI}^{B} + \omega_{BI}^{B} \times H_{BI}^{B} = \tau_e \tag{22}$$

where ω_{BI}^{B} is the satellite's angular speed relative to the inertial frame demonstrated in the body frame, $\tau_e \in \mathbb{R}^{3\times1}$ is the external force, and H_{BI}^{B} is the total angular momentum of the satellite. H_{BI}^{B} can be expressed as:

$$H_{BI}^{B} = J\omega_{BI}^{B} + h \tag{23}$$

where J expressed as $J = J_s - AJ_wA^T$ in which $J_s \in \mathbb{R}^{3\times3}$ is the satellite's inertia moment including the CMGs. $J_w \in \mathbb{R}^{4\times4} = diag([J_{w1}, J_{w2}, J_{w3}, J_{w4}])$ denotes the inertia moment of the CMGs in the axial direction. The torques provided by the CMGs are transformed into the axes of the satellite body by A, the transformation matrix. Substituting Eq. (23) into (22), and expressing h for CMG results in:

$$J\dot{\omega}_{BI}^{B} = -\omega_{BI}^{B} \times \left(J_s\omega_{BI}^{B} + h_{CMG}\right) - \dot{h}_{CMG} + \tau_e \tag{24}$$

where h_{CMG} is the CMGs' moment, and $\dot{h}_{CMG}$ is its derivative. The kinematic equations of the satellite can be expressed as:

$$\begin{bmatrix} \dot{q}_v \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 I + q_v^{\times} \\ -q_v^{T} \end{bmatrix} \omega_{BL}^{B} \tag{25}$$

where $\bar{q} = \left[\bar{e}\sin\left(\frac{\Phi}{2}\right), \cos\left(\frac{\Phi}{2}\right)\right]^T = \begin{bmatrix} q_v \\ q_4 \end{bmatrix}$ is the unit quaternion, Φ expresses the principal angle, $\bar{e} = [e_1, e_2, e_3]^T$ is the principal axis that conforms with the 'Euler's theorem $(e_1^2 + e_2^2 + e_3^2 = 1)$', $q_4 \in \mathbb{R}$ and $q_v \in \mathbb{R}^{3\times1} = [q_1, q_2, q_3]^T$ denotes the Euler parameters expressing the satellite body frame orientation with regard to the orbital frame where $q_v^4 q_v + q_4 = 1$. $I \in \mathbb{R}^{3\times3}$ is the unity matrix and $q_v^{\times}$ is the skew-symmetric matrix representation of the quaternion vector.

4.3 Controller and Steering Logic

The desired attitude of $q_d \in \mathbb{R}^{4\times1}$ and $\omega_d \in \mathbb{R}^{3\times1}$ are attained by a non-linear sliding mode controller in a simplified version [47]. the error terms for the quaternion tracking are expressed as:

$$q_e = q_{d4}q_v - q_4 q_{dv} + q_v^{\times} q_{dv}$$
$$q_{e4} = q_{d4}q_4 + q_{dv}^T q_v \tag{26}$$

where $q_e^T q_e + q_{4e}^2 = 1$. The rotating matrix, $C_e = C(q_e, q_{4e})$ is obtained using:

$$C_e = (q_{4e}^2 - q_e^T q_e)I + 2q_e q_e^T - 2q_{4e} q_e^{\times} \tag{27}$$

The relative angular speed $\omega_e \in \mathbb{R}^{3\times1}$ is expressed as:

$$\omega_e = \omega_{BL}^B - C_e \omega_d \tag{28}$$

Considering the error definitions shown in Eq. (26) and (28), the sliding manifold can be obtained from:

$$\sigma = \omega_e + \lambda sgn(q_{4e}) q_e \tag{29}$$

where $\lambda > 0$ expresses the gain for the sliding manifold and $sgn(q_{4e})$ represents the sign function for q_{4e}. Finally, the control command that is fed to the system can be expressed as:

$$u_r = -p_0 \sigma \tag{30}$$

where p_0 is a positive constant. In this work, all the parameters for the controller are set as [47], $\lambda = 1$ with regards to the values shown in [47], and $p_0 = 0.1$ based on the simulation outcomes.

As the CMGs have gimballing action, an extra component is needed for the controller that is known as the steering logic. The steering logic is responsible for converting the required torque from the controller to the required gimbal angle rates to generate that torque by the CMGs. The CMG angular momentum is, in general, a function of CMG gimbal angles, $\delta = (\delta_1, \dots, \delta_n)$, and flywheels angular speed, $\Omega = (\Omega_1, \dots, \Omega_n)$ given by:

$$H_{CMG} = H(\delta, \Omega) \tag{31}$$

where n is the number of CMGs. One of the CMG steering logic approaches is to use the differential relationship between gimbal angles and the CMG momentum vector. For such a method, the derivation of h is obtained as:

$$\dot{h}_{CMG} = A_{CMG}\dot{\delta} \tag{32}$$

where $A_{CMG} = A_{CMG}(\delta) \in \mathbb{R}^{3\times n}$ as the Jacobin matrix is

$$A_{CMG} = \frac{\partial h}{\partial \delta} = \left[\frac{\partial h_i}{\partial \delta_i}\right] \tag{33}$$

The gimbal rate can be calculated using Eq. (31). At first, h_{CMG} can be calculated based on the CMGs' configuration. For the pyramid configuration [47]:

$$\begin{aligned} h_{CMG} &= \sum_{i=1}^{4} h_i\,(\delta_i, \Omega_i) \\ &= \begin{bmatrix} -c\beta s\delta_1 & -c\delta_2 & c\beta s\delta_3 & c\delta_4 \\ c\delta_1 & -c\beta s\delta_2 & -c\delta_3 & c\beta s\delta_4 \\ c\beta s\delta_1 & c\beta s\delta_2 & c\beta s\delta_3 & c\beta s\delta_4 \end{bmatrix} \\ &\times \left[h_{0_1}(\Omega_1)h_{0_2}(\Omega_2)h_{0_3}(\Omega_3)h_{0_4}(\Omega_4)\right]^T \end{aligned} \tag{34}$$

where h_i is the angular momentum of each CMG expressed in the reference frame of the satellite. δ_i are the gimbal angles, Ω_i are the flywheel angular speed, and h_{0i} is the momentum magnitude for the ith CMG. The derivative of the CMG angular momentum versus time can be calculated as:

$$\dot{h}_{CMG} = \sum_{i=1}^{4} \dot{h}_i\,(\delta_i, \Omega_i) = \left[h_{0_1}(\Omega_1)h_{0_2}(\Omega_2)h_{0_3}(\Omega_3)h_{0_4}(\Omega_4)\right]A_{CMG}\dot{\delta} \tag{35}$$

where δ is the gimbal angle vector and:

$$A_{CMG} = \begin{bmatrix} -c\beta s\delta_1 & -c\delta_2 & c\beta s\delta_3 & c\delta_4 \\ c\delta_1 & -c\beta s\delta_2 & -c\delta_3 & c\beta s\delta_4 \\ c\beta s\delta_1 & c\beta s\delta_2 & c\beta s\delta_3 & c\beta s\delta_4 \end{bmatrix} \tag{36}$$

For a given control torque τ_c, the torque command of the CMG, $\dot{h}$, is selected as:

$$\dot{h}_{CMG} = u = -\tau_c - \omega_{BI}^{B} \times h_{CMG} \tag{37}$$

And the gimbal rate command $\dot{\delta}$, given $h_0 = h_{0_1} = h_{0_2} = h_{0_3} = h_{0_4}$ is calculated as [47]:

$$\dot{\delta} = \left(\frac{1}{h_0}\right) A_{CMG}^{+} \dot{h}_{CMG} \tag{38}$$

where $A_{CMG}^{+} = A_{CMG}^{T}(A_{CMG}A_{CMG}^{T})^{-1}$, is the pseudoinverse steering logic and most CMG steering logics determine the gimbal rate commands with variations of it.

4.4 Actuators

As the critical components of any satellite's ACS, the actuators provide the torque required for controlling the satellite attitude. In this model, four CMGs are used as actuators. CMG is a reaction wheel capable of changing its angular momentum direction by gimballing the spinning rotor. The CMGs receive the gimbal rate command as input to provide the required control torque for the satellite.

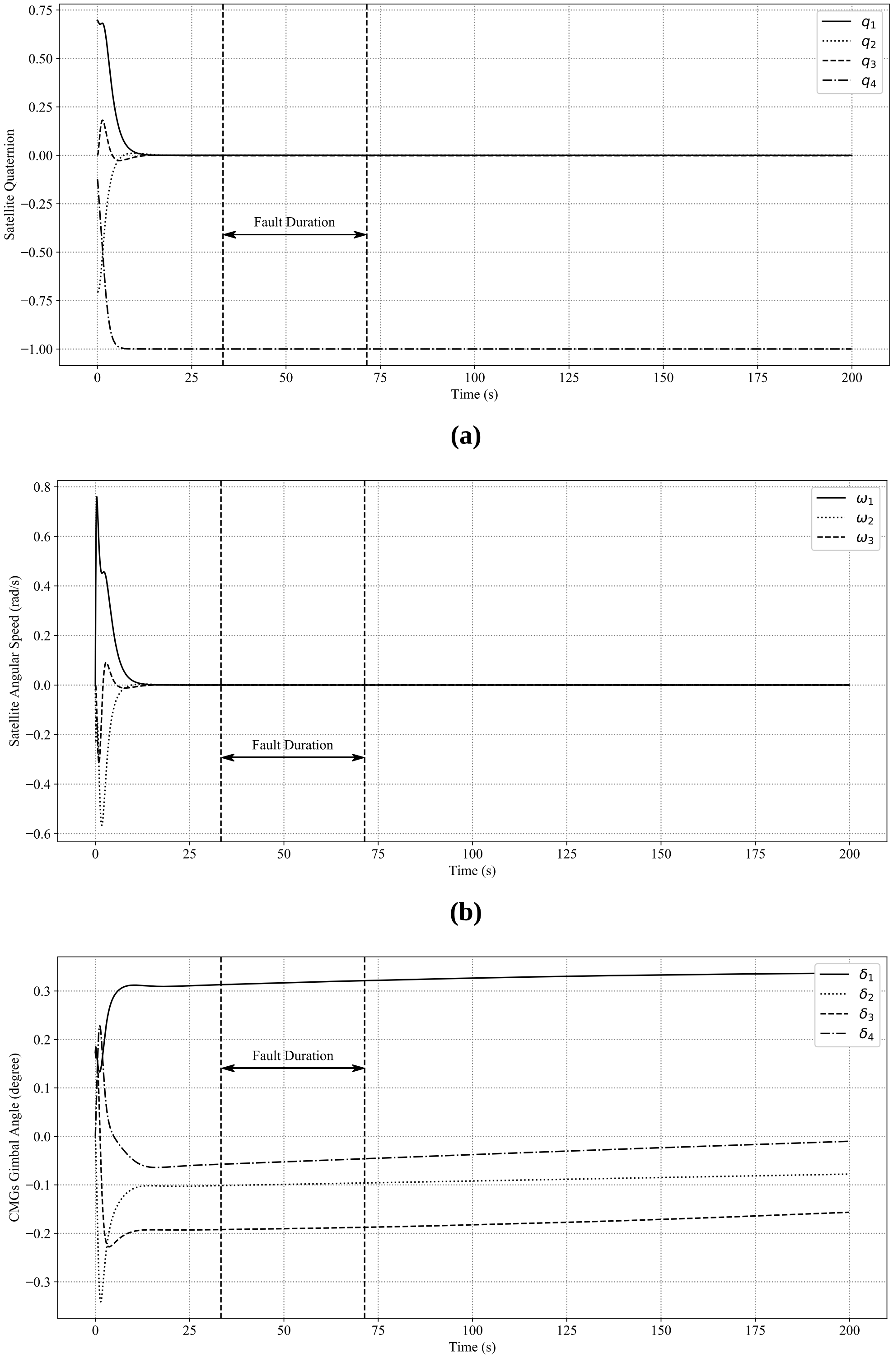
4.5 Raw Data

The simulator, which its components are described in sections 4.2 to 4.4, is used to generate the raw data. The first step is to define the required parameters for the desired fault scenarios that are shown in Table 3.1. These input parameters are shown in Table 4.1.

Table 4.1 - Required Inputs for the Satellite Simulator

Index	Description	Range	Unit
1	Fault Scenario	0-15	integer
2	f_1 fault parameter	0-1	float
3	f_2 fault parameter	0-1	float
4	f_3, fault parameter	0-1	float
5	f_4 fault parameter	0-1	float
6	fault inception	0-200	second
7	fault duration	0-50	second

Then, the parameters are fed to the satellite with CMGs in-board simulator. For the case study in this work, a CMG assembly with four CMGs is used. Each CMG unit can have effectiveness between 0 and 1 for a completely failed and a completely healthy CMG, respectively. As there are four CMG units in this work, four fault effectiveness are chosen randomly for each data instance. Each fault effectiveness is assigned to one of the CMG units and is multiplied by its output torque. The net output torque of the four CMG units is then fed into the satellite, and it is assumed that the faults injected into the CMG units can be detected and isolated by analyzing the satellite output parameters as raw data. So, the required data are extracted as time-series from the output of the satellite simulator for different fault scenarios shown in Table 3.1, and they include the satellite attitude parameters, quaternion, $q_{1-}\ q_4$, and angular speed, $\omega_1 - \omega_3$, along with CMGs gimbal angle, $\delta_1 - \delta_4$. The raw data were extracted from the simulator's outputs by running it, 40,000 times for each fault scenario. As there are 16 scenarios, in total, there were 640,000 data sets, each of them stored in a comma-delimited value (CSV) file. As the data related to q_4 are not independent of $q_1,\ q_2,\ q_3$, they were discarded in this work. Each time series has a time length of 200 seconds. There is a nominal and a faulty value for each parameter. A total of 22 columns and 2,000 rows were stored in each CSV file. As the simulation time step is 0.1 seconds and has 200 seconds length, 2,000 rows were generated. The number of columns is calculated as $11 \times 2 = 22$ for two sets of 11 parameters; one set expresses the healthy, and the other expresses the faulty situation.



(a)

(b)

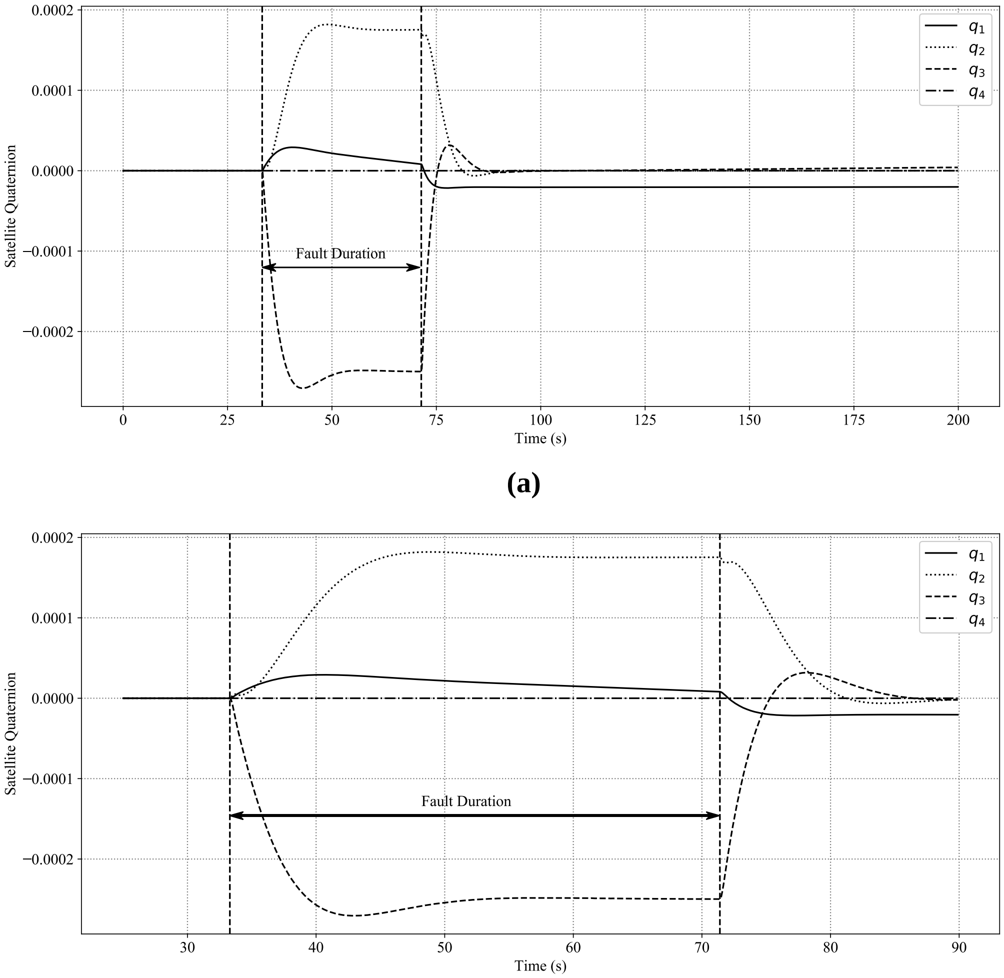
(c)

Figure 4.3 - Sample raw data, (a) the satellite quaternions (b) the satellite angular speed (c) the CMGs gimbal angle

Figure 4.3 shows a sample of the raw data used in this work. The raw data are used to calculate the residuals and then do the feature engineering, including feature extraction and selection/reduction.

4.6 Feature Engineering

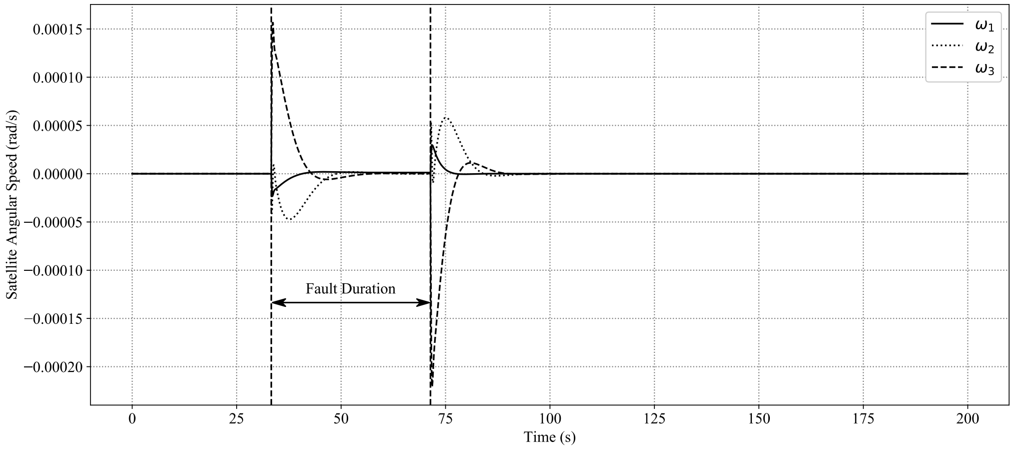
The residuals are the difference between the system's raw data in a healthy and faulty condition. Residuals are calculated for each instance of the raw data related to each fault scenario using Eq. (21). Figure 4.4 to Figure 4.6 show samples of the residual data used in this work. As these figures depict, the residuals have different behaviour during the fault period and this confirms their suitability to be used for fault detection and isolation.



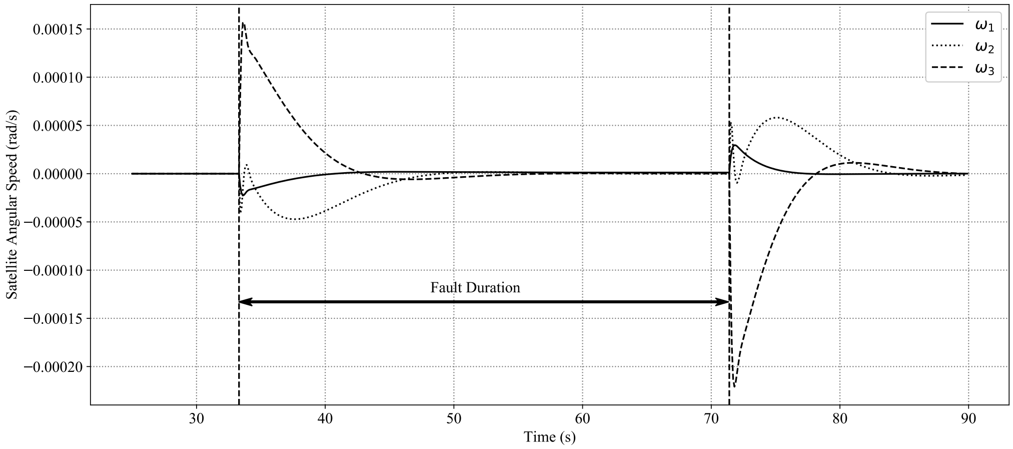
(a)

(b)

Figure 4.4 - Sample residual data for the satellite quaternions, (a) the whole simulation period (b) zoomed around the fault period

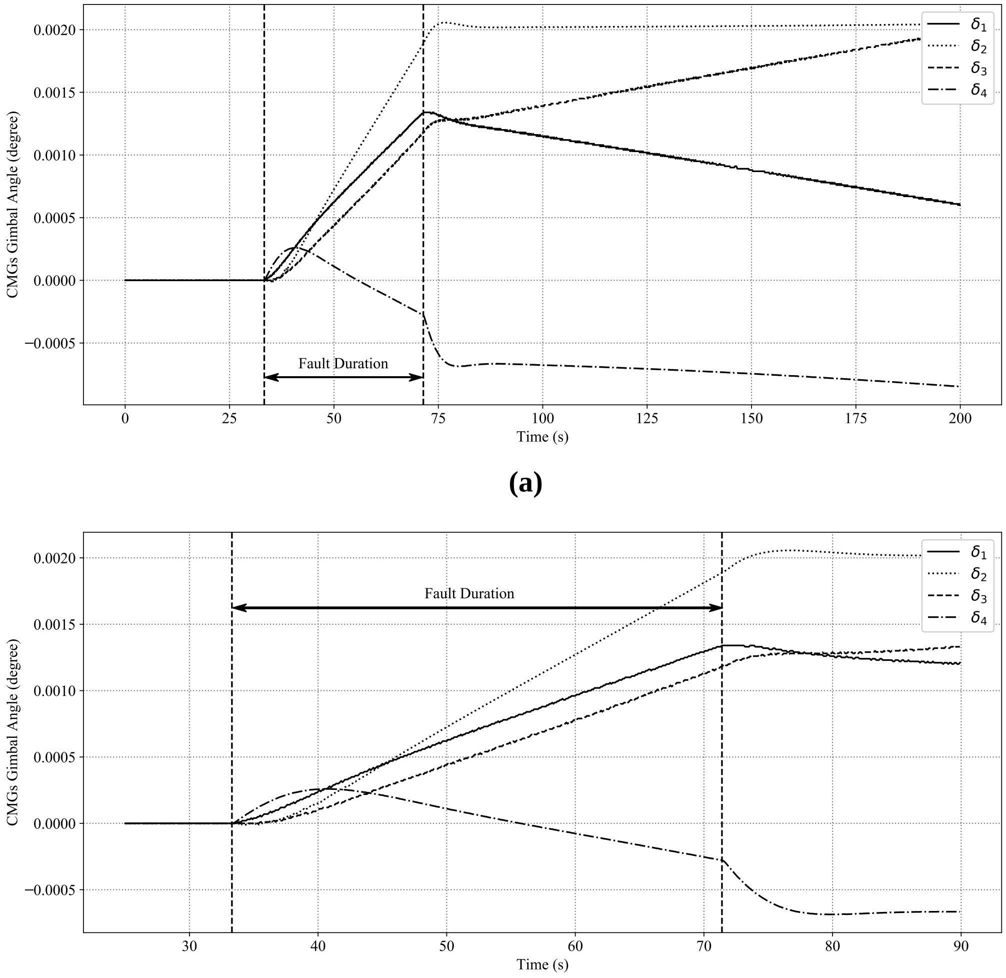


(a)



(b)

Figure 4.5 - Sample residual data for the satellite angular speed, (a) the whole simulation period (b) zoomed around the fault period



(a)

(b)

Figure 4.6 - Sample residual data for the CMGs gimbal angle, (a) the whole simulation period (b) zoomed around the fault period

The residuals are used to extract the features. To find a feature set that best represents the desired fault scenarios, different methods are used for feature extraction in this work described in sections 2.1.1 and 2.1.2. They include wavelet packet transform (WPT) [32], multi-domain analysis [28], correlation analysis [48], cross-correlation analysis [38], and multi correlation analysis [34]. The WPT and multi-domain analysis features are used to discover almost any pattern that can be present in time series data. This includes variations in the shape of the data, amplitude changes over a short period, and changes in data frequency. These two methods are considered as univariate analyses as they extract the features from each time series individually and do not consider any possible relation between any two sets of the time series. So, as there is more than one fault simultaneously active in the CMG units in this study, the residuals calculated from the satellite attitude parameters can have complicated relations with each other. For example, they can be correlated with each other differently for each possible scenario. So, there is a need to use multi-variable analysis techniques to handle this issue. Based on this assumption, correlation, cross-correlation and multi correlation analysis are also chosen for feature extraction in this study. Different methods are used for the correlation analysis to discover the different aspects of the relations between each pair of data [33]. In this work, the Pearson correlation coefficient, the Spearman rank correlation coefficient, and the Kendall rank correlation coefficient are used. These coefficients are introduced and described in section 2.1.2.

Feature reduction/selection aims at finding the most representative features to improve the model performance while reducing its time complexity. Different feature reduction/selection methods have been used in the literature [2] and described in sections 2.1.4 and 2.1.3. In this work, PCA [40], recursive feature elimination [49], and feature importance method [50] are used for this purpose. The chosen features are used for training and testing the machine learning models.

4.7 Machine Learning Models

Various machine learning approaches have been used for classification and regression analysis purposes in the literature [2] [5]. In this work, SVM [28], neural networks [12], random forest [51], and gradient boosting machines [52,53] are used for classification and support vector regressor (SVR) is used for regression analysis. The model with the best performance will then be selected for moving forward with.

In addition to the models mentioned above, different classification approaches, including multi-label classification, multi-step classification, and ensemble learning, are used in this work to improve the performance of the proposed scheme. The rationale for using the multi-label approach is that the fault scenarios include cases with more than one active fault, and the faulty units' number can be used as labels instead of the scenario number. For example, for scenario number 11, Table 3.1, it is possible to use the array [1, 2, 3] as the label instead of merely using 11. The multi-label method is implemented using a scikit-learn package called LabelPowerset [50]. This package transforms a multi-label problem into a multi-class problem with 1 multi-class classifier trained on all unique combinations of labels. The method maps each combination to a unique combination id number and conducts multi-class classification using the classifier as a multi-class classifier and combination ids as classes [50]. Multi-step classification is implemented by dividing the problem into finding the number of active faults and then using different classifiers for cases belonging to a different number of faults. Figure 4.7 shows the proposed method for multi-step classification. In step 1, the label set is [1, 2, 3], which are the possible number of active faults. In step 2, three classifiers are trained. Each classifier only deals with the cases with the same number of active faults.

In the next chapter, the proposed fault diagnosis scheme is applied to a case study to evaluate its performance.

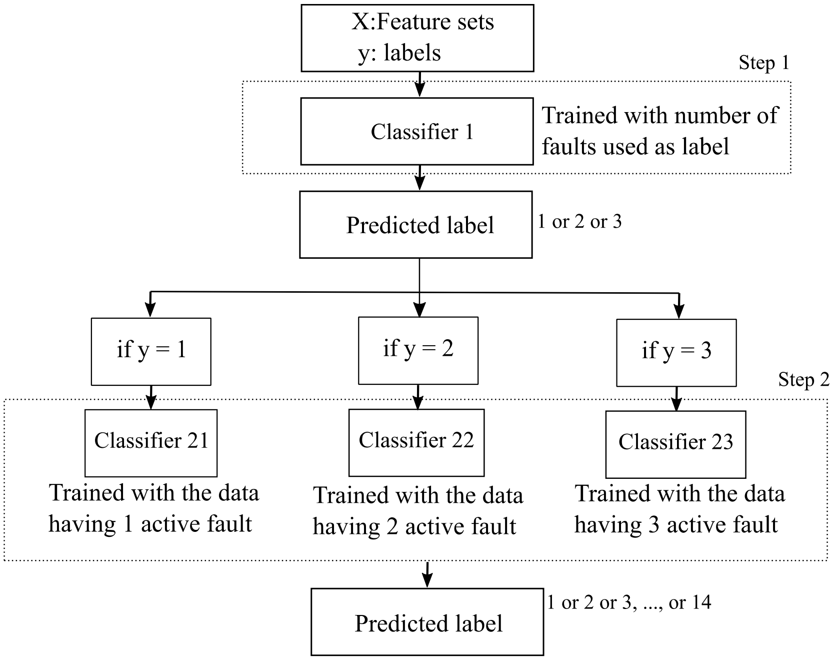


Figure 4.7 - The proposed method for multi-step classification

CHAPTER 5

RESULTS AND DISCUSSION

In this chapter, the proposed fault diagnosis scheme is used on the satellite with four CMGs to find the optimum choices for each step's method and evaluate the performance of the optimized scheme. The proposed scheme was run using a PC comprised of an Intel® Core™ i7-4790 CPU with a processing power of 3.6 GHz, 8 MB cache and 8 GB of RAM. The evaluation includes using different feature extraction methods, feature reduction/selection and machine learning to find the optimum method for each step. It also includes evaluating the performance of the optimized scheme for the test data and performing the sensitivity analysis to ensure that it is suitable for real applications.

5.1 Fault Isolation Results

In this section, the fault isolation results are presented.

5.1.1 Feature Extraction

By discarding q_4, as explained in section 4.5, 10 satellite attitude parameters are chosen to extract the features. Feature extraction is done using different methods to find the best suitable one. Table 5.1 shows the performance of the proposed scheme when different methods are used for feature extraction, in which the score are the averaged values obtained through a 5-folds cross validation analysis.

Table 5.1 - Comparison of different feature extraction methods using cross validation

Feature Extraction Method	Score (%)
Wavelet Packet Transform	63.5
Multi-Domain Analysis	70.2
Correlation Analysis	86.2
Cross-Correlation Analysis	75.6
Multi Correlation Analysis	85.4

The results show that the correlation analysis features provide the best performance for the proposed method among the other feature extraction methods. This set of features is chosen to be used in the next step, which is feature reduction/selection.

5.1.2 *Feature Reduction/Selection*

Table 5.2 shows the results for applying the proposed scheme with different feature reduction/selection methods.

Table 5.2 - Comparison of different feature reduction/selection methods

Feature Reduction/Selection Method	Input		Output	
	No of features	Score (%)	No of features	Score (%)
PCA	45	92.9	25	92.2
Recursive Feature Elimination	45	92.9	25	91.5
Feature Importance	45	92.9	25	91.6

The results depict that the PCA provides the highest score after feature reduction. So, the PCA is selected for feature reduction. The reduction aims to keep the features that represent 99% of all features variance. Figure 5.1 shows the features explained variance. As this figure depicts, the features reach 99% of the variance with 25 out of 45 components.

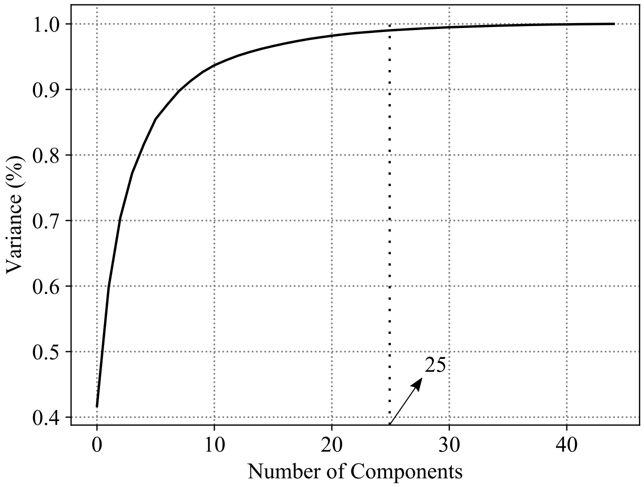


Figure 5.1 - Features Explained Variance

5.1.3 *Comparing the Performance of Different Machine Learning Methods*

Table 5.3 shows the results of using different machine learning methods. As the results show, the SVM model has the best performance than the neural networks, the gradient boosting machines, and the random forest.

Table 5.3 - Comparison of machine learning models

Model	Score (%)
SVM	92.2
Neural Networks	87.1
Gradient Boosting Machines	86.5
Random Forest	85.3

Table 5.4 shows the results of applying different classification approaches. The SVM method is used as the classification algorithm in both the multi-label and the multi-step approaches. As this table shows, neither of these two approaches has a better performance than the traditional machine learning methods, shown in Table 5.4.

Table 5.4 - Comparison of different classification approaches

Approach	Score (%)
Multi-Label Classification	82.1
Multi-Step Classification	77.1

Based on the results shown in Table 5.3 and Table 5.4, the SVM is selected as a machine learning model, and the complementary results are presented in sections 5.1.4 to 5.1.6.

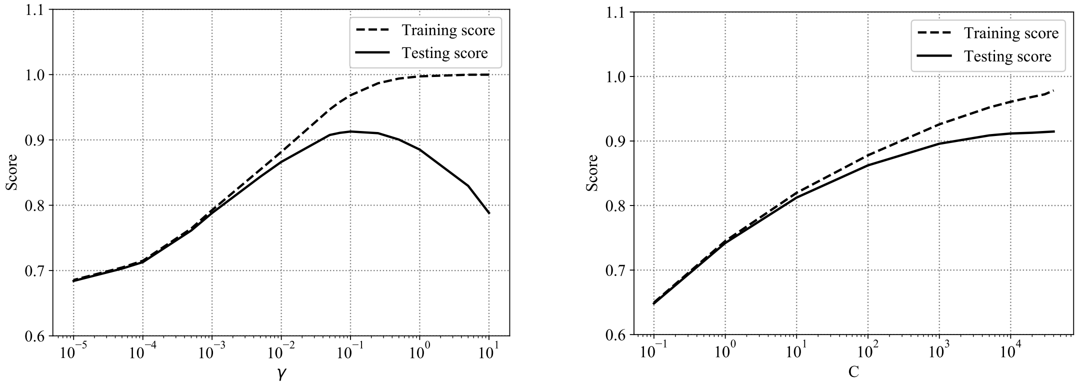
5.1.4 Validation/Learning Curves for the SVM Model

The optimum values and choices for the SVM model hyper-parameters are found using the grid search. Table 5.5 shows the search domain and the optimum value/choice for each hyper-parameter from the grid search. The coefficient C is the penalty factor that is used for the regularization of the model. This parameter makes a balance between the training accuracy and simplicity of the model. A small C makes the decision surface smooth, while a large C aims at classifying all of the training samples accurately [50]. The gamma coefficient applicable for "poly", "rbf", and "sigmoid" kernels, where rbf is an abbreviation for “radial basis function”. Gamma takes over the effect that each training example has on the model. By increasing gamma, only the closer samples are being affected [50]. The degree is only applicable for the polynomial kernel, and as "rbf" is selected as kernel through the grid search for this study, the degree does not apply here.

Table 5.5 - The SVM optimization grid search

Parameter	Search Domain	Optimum
C	0.1, 1, 10, 100, 1000, 5000, 10000, 20000, 30000, 100000	100000
gamma	1, 0.1, 0.01, 0.001, 0.0001	0.1
Kernel	linear, rbf, polynomial	rbf
Degree	2, 3, 4, 5, 6	—

Figure 5.2 shows the SVM model's validation curve for gamma and C. These figures confirm that the parameters' selected values are optimum.



(a) **(b)**

Figure 5.2 - SVM validation curves (a) score vs. gamma (b) score vs. C

Figure 5.3 shows the learning curve for the SVM model. In this figure, the model training score and testing score, calculated by the cross-validation method, are shown versus the number of training samples. By increasing the number of samples, the training score has decreased while the testing score is increasing until a specific point, 350’000 training samples, and then enters a plateau region. Based on this, it seems that having 640’000 total samples is enough when the ratio of train/test split is chosen as 0.7/0.3.

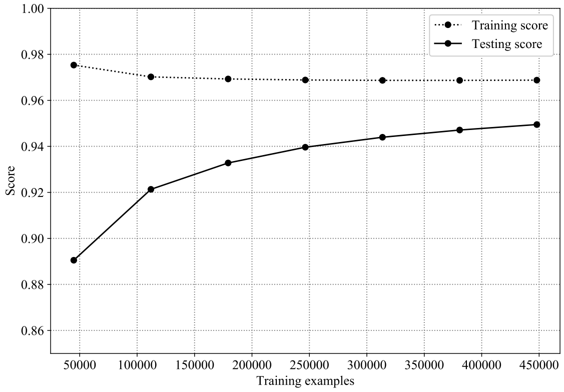


Figure 5.3 - The learning curve for the SVM model

The model was evaluated with the test data using five-fold cross-validation. Table 5.6 lists the results. It can be observed from Table 5.6 that the chosen model achieves a high score in every fold while the standard deviation is low. This means that no over-fitting has occurred, and the scores are very close to each other for different folds.

Table 5.6 - The results for five-folds cross-validation

Scores (%)	Mean (%)	Standard Deviation (%)
95.6, 95.7, 95.6, 95.5, 95.4	95.6	0.1

5.1.5 Confusion Matrix for the SVM Model

The confusion matrix is obtained for the test data to evaluate the model performance with more details per each scenario and is demonstrated in Table 5.7. The number of instances tested per class is used for normalizing the results.

Table 5.7 - The case study confusion matrix

Expected

0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	99	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	99	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	99	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	99	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	99	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	98	0	0	0	0	0	0	0	0	0
7	0	0	0	0	1	0	0	96	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	99	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	97	0	0	0	0	0	0
10	0	0	0	0	1	0	0	0	0	0	95	0	0	0	0	0
11	0	0	0	0	0	1	1	0	1	0	0	94	0	0	0	0
12	0	0	0	0	0	0	0	2	0	2	0	0	91	1	0	2
13	0	0	0	0	0	0	0	3	0	0	3	0	1	87	0	2
14	0	0	0	0	0	0	0	0	0	3	2	0	0	0	90	2
15	0	0	0	0	0	0	0	0	0	0	0	1	7	5	6	77
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Predicted

The values in the diagonal depict the percentage of the instances predicted correctly. The first scenario that is related to all healthy units has 100 percent accuracy. The next four scenarios are related to the cases with only one faulty unit and have 99 percent accuracy. However, as the number of faulty units increases, the model performance degrades for the next scenarios. For the scenarios with only one faulty unit, the accuracy is, on average, 99%. In the cases that have two concurrently faulty units, the average accuracy reduces to 97.8%. This pattern continues with an average accuracy of 91.4% and 77%, with three and four faulty units, respectively. Therefore, the model performance degrades as more faults are present simultaneously. This behaviour can be explained due to the overlap that the cases with more than 1 active faults have with the other cases which have the same faulty

CMG units. For example, scenario 13 has three active faults in CMG unit numbers 1, 3, and 4, as shown in .Figure 4.1 This scenario has overlap with scenarios number 7, 10, and 15 in which the faulty units are (1, 4), (3, 4), and (1, 2, 3, 4), respectively. As Table 5.7 depicts, for scenario 13, the mentioned scenarios have the largest percentage of mispredicted labels, same as for other scenarios.

5.1.6 Sensitivity Analysis of the SVM Model

In this section, a comprehensive sensitivity analysis for the proposed model is presented. The model's sensitivity is evaluated for noise, missing sensors (due to sensor failure), and missing measurements (due to sensor fault) to ensure the model's robustness.

5.1.6.1. Number of Scenarios

A total of 16 different scenarios are considered in this study, as shown in Table 3.1. Table 5.8 shows the results for subsets of all 16 scenarios, including one active fault, two active faults, three active faults, four active faults and a combination of these. As the results show, the model has 100% accuracy for the scenarios where there is only one active fault. The accuracy drops gradually as the number of simultaneous faults increases to four.

Table 5.8 - Model’s sensitivity to the number of active faults

Scenarios	1 to 4	1 to 10	1 to 14	1 to 15
Maximum Number of Active Faults	1	2	3	4
Score (%)	100	98.4	95.5	93.2

5.1.6.1 The Effect of Noise

Noise has been added to the raw data with different signal-to-noise ratio (SNR) levels to study the effect of noisy raw data on the model performance. The added noise in this study is Gaussian with a zero mean. Table 5.9 shows the results for different levels of SNR. The results show that the model performance degrades as the SNR decreases. It should be noted that the model maintains a reasonable score when the SNR is above 50dB, which is the case in most practical applications.

Table 5.9 - Model's sensitivity to noise

SNR (dB)	No Noise	60	50	40	30	20	10
Score (%)	86	78	70	56	37	22	13

5.1.6.2 Missing Sensors

The satellite attitude parameters and the CMGs gimbal angles, used as raw data in this work, represent sensor readings from the satellite. In practical applications, there may be situations where some of these sensors malfunction or fail. In this section, a study is done on the cases where one or more sensors have failed, and the data is not available from these sensors. As the feature set is different for each case of available sensors, the machine learning model should be trained and tested separately for each case, and hence, there will be multiple developed models to be used for each case [54]. Another approach for dealing with a reduced number of measurements is transfer learning which is addressed in [55]. In this work, the SVM model is trained and tested for each of the possible missing sensor scenarios. Table 5.10 shows the results for different possible failed sensor combinations. As the results show in Table 5.10, the model accuracy degrades when one or more sensors fail. However, in cases where 6 or more out of 10 sensors are properly functioning, the model performance is reasonable for real applications.

Table 5.10 - Model's sensitivity to missing sensors

Functioning Sensors	Score (%)
$q_1, q_2, q_3, \omega_1, \omega_2, \omega_3, \delta_1, \delta_2, \delta_3, \delta_4$	86.4
$q_1, q_2, q_3, \omega_1, \omega_2, \omega_3, \delta_1, \delta_2, \delta_3$	85.4
$q_1, q_2, q_3, \omega_1, \omega_3, \delta_1, \delta_2, \delta_3, \delta_4$	85.8
$q_2, q_3, \omega_1, \omega_2, \omega_3, \delta_1, \delta_2, \delta_3, \delta_4$	86.0
$q_1, q_2, \omega_1, \omega_3, \delta_1, \delta_2, \delta_3, \delta_4$	85.1
$q_2, q_3, \omega_1, \omega_2, \omega_3, \delta_1, \delta_3, \delta_4$	86.0
$\omega_1, \omega_2, \omega_3, \delta_1, \delta_2, \delta_3, \delta_4$	84.5
$q_1, q_2, q_3, \delta_1, \delta_2, \delta_3$	84.6
$q_1, q_2, q_3, \omega_1, \omega_2, \omega_3$	83.4
$\omega_1, \omega_2, \omega_3, \delta_1, \delta_2, \delta_3$	81.7
$q_1, q_2, \omega_1, \omega_2, \delta_1, \delta_2$	79.3
$\delta_1, \delta_2, \delta_3, \delta_4$	73.7
q_1, q_2, q_3	55.7
$\omega_1, \omega_2, \omega_3$	61.8
q_1, ω_1, δ_1	41.3

5.1.6.3 Missing Values

It is common for sensory data to contain missing values due to faults in communication channels or sensor components. In this section, an analysis is done to evaluate the model performance for missing data. The original raw data used in this study does not have any missing values. Hence, missing values are added to the original dataset (generated through simulations) manually at different percentages to conduct this analysis. There are different methods available for imputing the missing values in sensory data [56]. In this work, the linear interpolation imputes the missing values before moving forward for calculating the residuals and extracting the features. Table 5.11 shows the results for possible missing data percentages. The model score drops as the percentage of the missing values increases. However, the model score is still reasonable for 10% or less missing values.

Table 5.11 - Model's sensitivity to missing values

Missing values (%)	0	1	3	5	7	10	20	35	50
Score (%)	86.4	79.6	75.2	73.4	70.8	69.3	64.3	55.8	47.8

5.2 Fault Identification Results

In this section, the fault Identification results are presented. Fault identification is done using an optimized SVR model. The correlation analysis is used for feature extraction, as is described in section 2.1.2. As shown in Table 5.12, the best score is obtained using all of the different correlation coefficients.

Table 5.12 - The SVR Model Performance for Different Feature Sets

Feature set	Score (%)
Pearson, Spearman, and Kendall Coefficients	94.9
Pearson Coefficients	89.4

The most representative features are selected using the principal component analysis for the SVR model. Figure 5.4 demonstrates how the feature space captures the data variance after reduction for the whole feature set. It shows that 60 features out of the total of 135 can capture 0.99 of the data variance. At this step, the chosen features were provided for use in the proposed scheme for regression analysis.

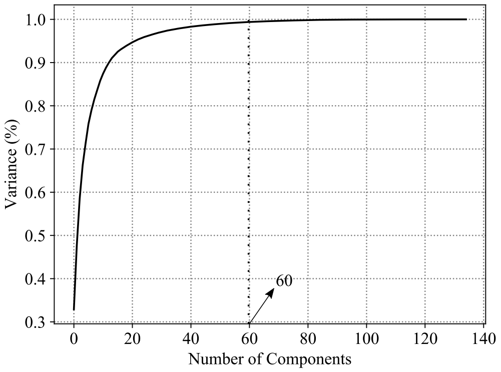


Figure 5.4 - Features explained variance for the SVR model

The parameters search domain and the final optimum choices for each model are shown in Table 5.13.

Table 5.13 - The SVR optimization grid search

Parameter	Search Domain	Optimum
C	0.1, 1, 10, 100, 1000, 5000, 10000, 20000, 30000, 100000	10
gamma	1, 0.1, 0.01, 0.001, 0.0001	0.1
Kernel	linear, rbf, polynomial	rbf
Degree	2, 3, 4, 5, 6	—

Figure 5.5 and Figure 5.6 show the SVR model's validation curve for gamma and C, respectively. These figures depict that the selected parameter values are optimum as the maximum score has occurred at the points selected as optimum values.

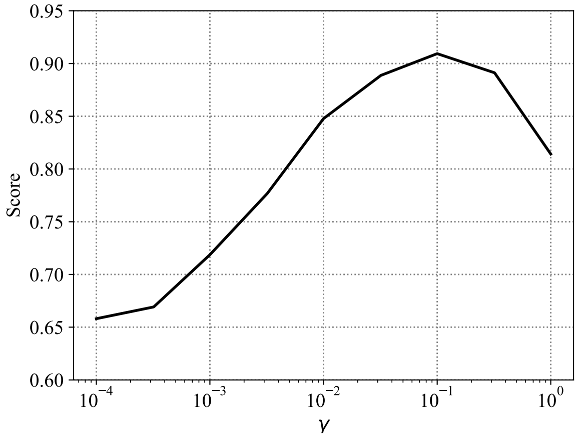


Figure 5.5 - Validation curve of the SVR model for gamma

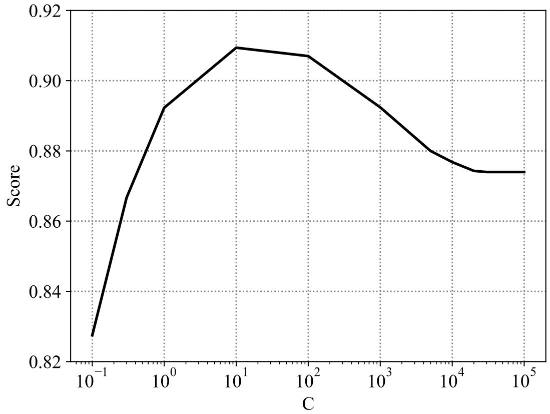


Figure 5.6 - Validation curve of the SVR model for C

The SVR models were tested for all inputs using a cross-validation approach with five folds. The results are shown in Table 5.14. The model has a high score in each fold with a low standard deviation based on the results. This implies that over-fitting had not occurred.

Table 5.14 - The regression analysis results for five-folds cross-validation

Variable	Scores (%)	Mean (%)	Standard Deviation (%)
f_1	94.9, 93.0, 96.1, 92.6, 88.3	93.0	3.0
f_2	94.6, 91.9, 94.9, 93.8, 87.6	92.6	3.0
f_3	94.5, 91.8, 95.4, 94.5, 89.7	93.2	2.4
f_4	96.2, 95.2, 96.0, 97.0, 95.2	95.9	0.7

The regression accuracy of the proposed SVR model is measured against the test set. Table 5.15 shows the mean squared error (MSE) of the regression analysis for each fault parameter, which is very low for all fault parameters.

Table 5.15 - MSE of regression analysis results

Variable	Mean error
f_1	0.0059
f_2	0.0066
f_3	0.0060
f_4	0.0034

The results are also shown in Figure 5.7 to Figure 5.10. Each of these figures is devoted to each one out of 4 CMG units. The score expresses the performance of the model to predict the value for the fault parameter. These figures can be used to evaluate the model performance for each CMG unit and each fault scenario in detail. The scenarios in which the CMG unit is healthy are not shown in these figures. For example, in scenario 2, CMG unit 2 is healthy. So, no value for scenario 2 is reported in Figure 5.7, which is devoted to CMG unit 1 fault parameter or f_1. Fault scenarios are shown in Table 3.1 and fault parameters are described in section 4.5. Figure 5.7 to Figure 5.10 also include the results from [23] for comparison. As these figures depict, this work has improved the scores considerably, especially for the scenarios with 2 and more active faults. This is achieved mainly by using more representative features.

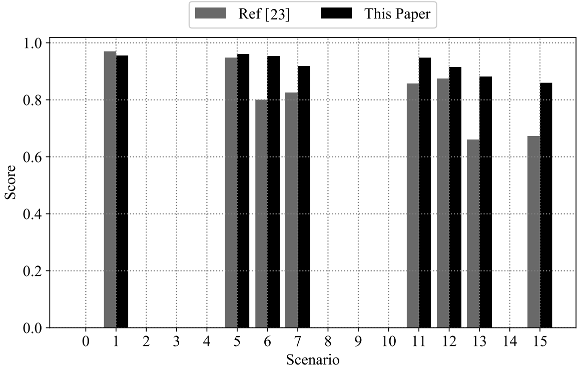


Figure 5.7 - The SVR model score versus fault scenarios for f_1

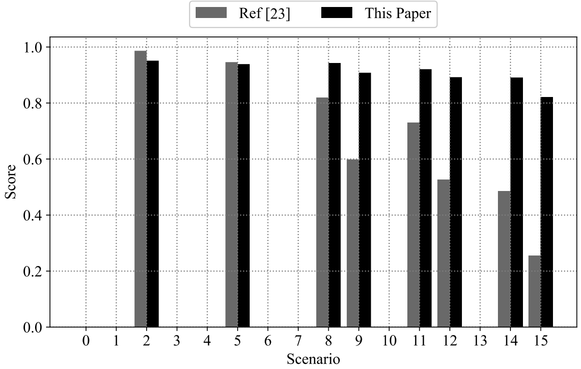


Figure 5.8 - The SVR model score versus fault scenarios for f_2

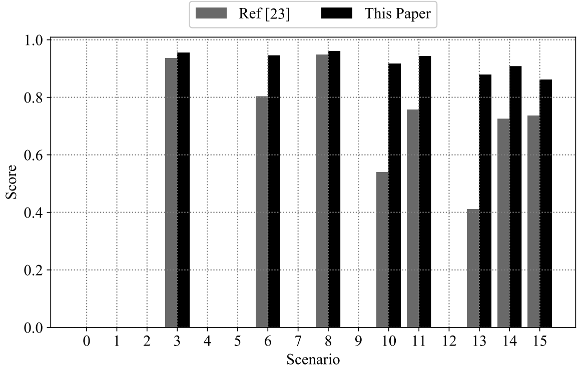


Figure 5.9 - The SVR model score versus fault scenarios for f_3

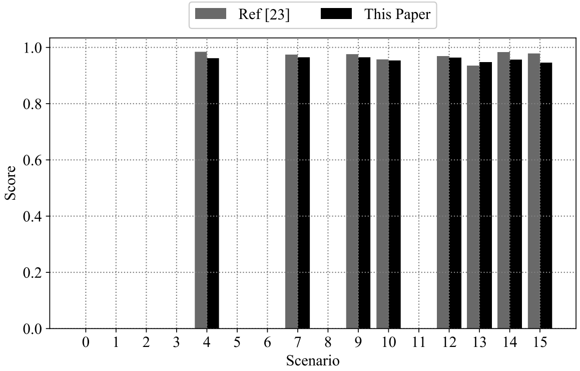


Figure 5.10 - The SVR model score versus fault scenarios for f_4

As the results show, the score is high for all fault parameters and scenarios while it decreases as the number of active faults increases. The lowest score belongs to scenario 15, which includes 4 active faults, i.e. when all of the CMG units are faulty. This behaviour can be attributed to the overlap the faults have with each other for cases where more than one fault is simultaneously present. For example, for scenario 15, there are 4 faults active at the same time. The overlap between these faults makes it challenging to predict the correct values for each fault parameter.

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

In this work, a data-driven fault diagnosis scheme is established for isolating and identifying multiple in-phase CMG faults onboard a 3-axis stabilized satellite. Different methods/approaches are used for extracting the features, feature reduction/selection, and machine learning to find the optimum method/approach for moving forward with the proposed scheme. The optimized fault diagnosis scheme is then evaluated through a case study. This includes evaluating the models' performance using the test data and performing a comprehensive sensitivity analysis. The results show that the proposed scheme can isolate and identify the faulty CMG units for different possible fault scenarios with a reasonable score. The sensitivity analysis results show that the proposed scheme is robust enough to be used in real applications, including noisy data, missing values, and missing sensors. Overall, this work proves the possibility of implementing a data-driven FDI scheme for isolating and identifying multiple in-phase faults in CMGs onboard satellites.

6.2 Future Works

Improving the models' performance by doing more research on feature engineering (e.g. investigating new and advanced signal processing techniques) and machine learning methods can be one of the possible paths for future works. Then, applying the proposed scheme to the real data from the satellite telemetry system can be considered a reasonable extension of this work. For doing this, there may be a need to use some new methods such as transfer learning [57,58]. Further research can also be conducted to evaluate the performance of the proposed diagnostic scheme on the other types of satellite ACS, including reaction wheels and momentum wheels. In addition, future studies can also focus on fault prognosis and remaining useful life estimation (RUL).

REFERENCES

[1] Ekanayake, T., Dewasurendra, D., Abeyratne, S., Ma, L., and Yarlagadda, P. “Model-Based Fault Diagnosis and Prognosis of Dynamic Systems: A Review.” *Procedia Manufacturing*, Vol. 30, 2019, pp. 435–442. https://doi.org/10.1016/j.promfg.2019.02.060.

[2] Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., and Nandi, A. K. “Applications of Machine Learning to Machine Fault Diagnosis: A Review and Roadmap.” *Mechanical Systems and Signal Processing*, Vol. 138, 2020, p. 106587. https://doi.org/10.1016/j.ymssp.2019.106587.

[3] Md Nor, N., Che Hassan, C. R., and Hussain, M. A. “A Review of Data-Driven Fault Detection and Diagnosis Methods: Applications in Chemical Process Systems.” *Reviews in Chemical Engineering*, Vol. 36, No. 4, 2020, pp. 513–553. https://doi.org/10.1515/revce-2017-0069.

[4] Hassanien, A. E., Darwish, A., and Abdelghafar, S. “Machine Learning in Telemetry Data Mining of Space Mission: Basics, Challenging and Future Directions.” *Artificial Intelligence Review*, 2019. https://doi.org/10.1007/s10462-019-09760-1.

[5] Cerrada, M., Sánchez, R. V., Li, C., Pacheco, F., Cabrera, D., Valente de Oliveira, J., and Vásquez, R. E. “A Review on Data-Driven Fault Severity Assessment in Rolling Bearings.” *Mechanical Systems and Signal Processing*, Vol. 99, 2018, pp. 169–196. https://doi.org/10.1016/j.ymssp.2017.06.012.

[6] Zhao, S., and Zhang, Y.-C. “SVM Classifier Based Fault Diagnosis of the Satellite Attitude Control System.” *2008 International Conference on Intelligent Computation Technology and Automation (ICICTA)*, Vol. 2, 2008, pp. 907–911. https://doi.org/10.1109/ICICTA.2008.409.

[7] Nozari, H. A., Castaldi, P., Banadaki, H. D., and Simani, S. “Novel Non-Model-Based Fault Detection and Isolation of Satellite Reaction Wheels Based on a Mixed-Learning Fusion Framework.” *IFAC-PapersOnLine*, Vol. 52, No. 12, 2019, pp. 194–199. https://doi.org/10.1016/j.ifacol.2019.11.222.

[8] Al-Zaidy, A. M., Hussein, W. M., Sayed, M. M. A., and El-Sherif, I. “Data Driven Models for Satellite State-of-Health Monitoring and Evaluation.” *International Journal of Robotics and Mechatronics*, Vol. 5, No. 1, 2018, pp. 1–11.

https://doi.org/10.21535/ijrm.v5i1.982.

[9] Hu, D., Dong, Y., and Sarosh, A. "An Improved PSO-SVM Approach for Multi-Faults Diagnosis of Satellite Reaction Wheel." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 6320 LNAI, No. PART 2, 2010, pp. 114–123. https://doi.org/10.1007/978-3-642-16527-6_16.

[10] Omran, E. A., and Murtada, W. A. "Efficient Anomaly Classification for Spacecraft Reaction Wheels." *Neural Computing and Applications*, Vol. 31, No. 7, 2019, pp. 2741–2747. https://doi.org/10.1007/s00521-017-3226-y.

[11] Sheng, G., Wei, Z., Xu, H., and Yunxia, C. "Neural Network-Based Fault Diagnosis Scheme for Satellite Attitude Control System." *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, 2018, pp. 3990–3995. https://doi.org/10.1109/CCDC.2018.8407816.

[12] Sun, B., Wang, J., He, Z., Zhou, H., and Gu, F. "Fault Identification for a Closed-Loop Control System Based on an Improved Deep Neural Network." *Sensors (Basel, Switzerland)*, Vol. 19, No. 9, 2019. https://doi.org/10.3390/s19092131.

[13] Wen, X., Wang, J., and Li, X. "A Feed-Forward Wavelet Neural Network Adaptive Observer-Based Fault Detection Technique for Spacecraft Attitude Control Systems." *Chinese Journal of Electronics*, Vol. 27, No. 1, 2018, pp. 102–108. https://doi.org/10.1049/cje.2017.11.010.

[14] Song, Y., Zhong, M., Xue, T., Ding, S. X., and Li, W. "Parity Space-Based Fault Isolation Using Minimum Error Minimax Probability Machine." *Control Engineering Practice*, Vol. 95, No. April 2019, 2020, p. 104242. https://doi.org/10.1016/j.conengprac.2019.104242.

[15] Mazzoleni, M., Maccarana, Y., and Previdi, F. "A Comparison of Data-Driven Fault Detection Methods with Application to Aerospace Electro-Mechanical Actuators." *IFAC-PapersOnLine*, Vol. 50, No. 1, 2017, pp. 12797–12802. https://doi.org/10.1016/j.ifacol.2017.08.1837.

[16] Frpsrqhqw, S., Vlv, D., Kxd, D. Q., Ql, L. F. L. D., Tldqj, X. H., Hl, L. D., Dq, M. X. Q., Frp, Q., Lpsuryh, F. D. Q., Uholdelolw, W. K. H., and Vwhp, R. I. V. "The Flywheel Fault Detection Based on Kernel Principal Component Analysis." No.

Itnec, 2019, pp. 425–432. https://doi.org/10.1109/ITNEC.2019.8729163.

[17] Lu, C., Wang, S., and Makis, V. "Fault Severity Recognition of Aviation Piston Pump Based on Feature Extraction of EEMD Paving and Optimized Support Vector Regression Model." *Aerospace Science and Technology,* Vol. 67, No. March, 2017, pp. 105–117. https://doi.org/10.1016/j.ast.2017.03.039.

[18] Liang, Z., Cao, J., Ji, X., and Wei, P. "Fault Severity Assessment of Rolling Bearings Method Based on Improved VMD and LSTM." *Journal of Vibroengineering,* Vol. 22, No. 6, 2020, pp. 1338–1356. https://doi.org/10.21595/jve.2020.21292.

[19] Sharma, A., Amarnath, M., and Kankar, P. K. "Feature Extraction and Fault Severity Classification in Ball Bearings." *JVC/Journal of Vibration and Control,* Vol. 22, No. 1, 2016, pp. 176–192. https://doi.org/10.1177/1077546314528021.

[20] Tan, X. D., and Zhang, Y. "Quantitative Fault Severity Degree Assessment for Rolling Bearings Based on Multivariate Regression Model Optimization." *Proceedings of 2019 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering, QR2MSE 2019,* No. Qr2mse, 2019, pp. 14–20. https://doi.org/10.1109/QR2MSE46217.2019.9021207.

[21] Feature Selection and Fault-severity Classification–Based Machine Health Assessment - Atamuradov2019.Pdf.

[22] Cabrera, D., Sancho, F., Li, C., Cerrada, M., Sánchez, R. V., Pacheco, F., and de Oliveira, J. V. "Automatic Feature Extraction of Time-Series Applied to Fault Severity Assessment of Helical Gearbox in Stationary and Non-Stationary Speed Operation." *Applied Soft Computing Journal,* Vol. 58, 2017, pp. 53–64. https://doi.org/10.1016/j.asoc.2017.04.016.

[23] Rahimi, A., and Raad, E. "Machine Learning Applied to Control Moment Gyroscope Fault Diagnosis." *124th International Conference on Robotics, Aeronautics, Mechanics and Mechatronics (ICRAMM),* No. August, 2019.

[24] Varvani Farahani, H., and Rahimi, A. "Fault Diagnosis of Control Moment Gyroscope Using Optimized Support Vector Machine." *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC),* 2020, pp. 3111–3116. https://doi.org/10.1109/SMC42975.2020.9283402.

[25] Sobhani-Tehrani, E., Talebi, H. A., and Khorasani, K. "Hybrid Fault Diagnosis of

Nonlinear Systems Using Neural Parameter Estimators." *Neural Networks,* Vol. 50, 2014, pp. 12–32. https://doi.org/10.1016/j.neunet.2013.10.005.

[26] Dong, Guozhu, and Liu, H. *Feature Engineering for Machine Learning and Data Analytics*. CRC Press, 2018.

[27] Wang, L., Wu, X., Zhang, C., and Shi, H. "Hydraulic System Fault Diagnosis Method Based on a Multi-Feature Fusion Support Vector Machine." *The Journal of Engineering,* Vol. 2019, No. 13, 2019, pp. 215–218. https://doi.org/10.1049/joe.2018.9028.

[28] Yan, X., and Jia, M. "A Novel Optimized SVM Classification Algorithm with Multi-Domain Feature and Its Application to Fault Diagnosis of Rolling Bearing." *Neurocomputing,* Vol. 313, 2018, pp. 47–64. https://doi.org/10.1016/j.neucom.2018.05.002.

[29] Xie, Y., Li, Z., and Xu, C. "Feature Extraction of Space Targets via Variational Mode Decomposition." *2018 IEEE International Conference on Information and Automation, ICIA 2018,* No. August, 2018, pp. 1530–1534. https://doi.org/10.1109/ICInfA.2018.8812476.

[30] Ren, H., Liu, W., Shan, M., and Wang, X. "A New Wind Turbine Health Condition Monitoring Method Based on VMD-MPE and Feature-Based Transfer Learning." *Measurement,* Vol. 148, 2019, p. 106906. https://doi.org/10.1016/j.measurement.2019.106906.

[31] Hu, Q., Qin, A., Zhang, Q., He, J., and Sun, G. "Fault Diagnosis Based on Weighted Extreme Learning Machine with Wavelet Packet Decomposition and KPCA." *IEEE Sensors Journal,* Vol. 18, No. 20, 2018, pp. 8472–8483. https://doi.org/10.1109/JSEN.2018.2866708.

[32] Tang, J., Liu, Q., Hu, J., Huo, J., and Wang, L. "Leakage Fault Diagnosis Method of Aircraft Landing Gear Hydraulic Cylinder Based on Wavelet Packet." *The Journal of Engineering,* Vol. 2019, No. 13, 2019, pp. 427–431. https://doi.org/10.1049/joe.2018.9037.

[33] Akoglu, H. "User's Guide to Correlation Coefficients." *Turkish Journal of Emergency Medicine,* Vol. 18, No. 3, 2018, pp. 91–93. https://doi.org/10.1016/j.tjem.2018.08.001.

[34] Wang, J., and Zheng, N. “Measures of Correlation for Multiple Variables.” 2014, pp. 1–18.

[35] Bose, R., Samanta, K., and Chatterjee, S. “Cross-Correlation Based Feature Extraction from EMG Signals for Classification of Neuro-Muscular Diseases.” *2016 International Conference on Intelligent Control, Power and Instrumentation, ICICPI 2016*, 2017, pp. 241–245. https://doi.org/10.1109/ICICPI.2016.7859710.

[36] Chandaka, S., Chatterjee, A., and Munshi, S. “Support Vector Machines Employing Cross-Correlation for Emotional Speech Recognition.” *Measurement: Journal of the International Measurement Confederation*, Vol. 42, No. 4, 2009, pp. 611–618. https://doi.org/10.1016/j.measurement.2008.10.005.

[37] Mcbain, J., and Timusk, M. “Cross Correlation for Condition Monitoring of Variable Load and Speed Gearboxes.” Vol. 2014, 2014.

[38] Paranjape, P. N., Dhabu, M. M., Deshpande, P. S., and Kekre, A. M. “Cross-Correlation Aided Ensemble of Classifiers for BCI Oriented EEG Study.” *IEEE Access*, Vol. 7, 2019, pp. 11985–11996. https://doi.org/10.1109/ACCESS.2019.2892492.

[39] Chen, P. Y., and Popovich, P. M. *Correlation: Parametric and Nonparametric Measures*. SAGE Publications, Inc., 2002.

[40] Shaikh, S. M., Halepoto, I. A., Phulpoto, N. H., Memon, M. S., Hussain, A., and Laghari, A. A. “Data-Driven Based Fault Diagnosis Using Principal Component Analysis.” *International Journal of Advanced Computer Science and Applications*, Vol. 9, No. 7, 2018, pp. 175–180. https://doi.org/10.14569/IJACSA.2018.090725.

[41] SMOLA, A. J., and SCHOLKOPF, B. “A Tutorial on Support Vector Regression.” *Statistics and Computing*, Vol. 14, 2004, pp. 199–222.

[42] Serpen, G., and Gao, Z. “Complexity Analysis of Multilayer Perceptron Neural Network Embedded into a Wireless Sensor Network.” *Procedia Computer Science*, Vol. 36, No. C, 2014, pp. 192–197. https://doi.org/10.1016/j.procs.2014.09.078.

[43] Livni, R., Shalev-Shwartz, S., and Shamir, O. “On the Computational Efficiency of Training Neural Networks.” *Advances in Neural Information Processing Systems*, Vol. 1, No. January, 2014, pp. 855–863.

[44] Abdiansah, A., and Wardoyo, R. “Time Complexity Analysis of Support Vector

Machines (SVM) in LibSVM." *International Journal of Computer Applications*, Vol. 128, No. 3, 2015, pp. 28–34. https://doi.org/10.5120/ijca2015906480.

[45] Chen, T., and Guestrin, C. "XGBoost: A Scalable Tree Boosting System." *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Vols. 13-17-Augu, 2016, pp. 785–794. https://doi.org/10.1145/2939672.2939785.

[46] Hassine, K., Erbad, A., and Hamila, R. "Important Complexity Reduction of Random Forest in Multi-Classification Problem." *2019 15th International Wireless Communications and Mobile Computing Conference, IWCMC 2019*, 2019, pp. 226–231. https://doi.org/10.1109/IWCMC.2019.8766544.

[47] Rahimi, A., Dev Kumar, K., and Alighanbari, H. "Fault Detection and Isolation of Control Moment Gyros for Satellite Attitude Control Subsystem." *Mechanical Systems and Signal Processing*, Vol. 135, 2020, p. 106419. https://doi.org/10.1016/j.ymssp.2019.106419.

[48] Musa, M. H. H., He, Z., Fu, L., and Deng, Y. "A Correlation Coefficient-Based Algorithm for Fault Detection and Classification in a Power Transmission Line." *IEEJ Transactions on Electrical and Electronic Engineering*, Vol. 13, No. 10, 2018, pp. 1394–1403. https://doi.org/10.1002/tee.22705.

[49] Mahadevan, S., and Shah, S. L. "Fault Detection and Diagnosis in Process Data Using One-Class Support Vector Machines." *Journal of Process Control*, Vol. 19, No. 10, 2009, pp. 1627–1639. https://doi.org/10.1016/j.jprocont.2009.07.011.

[50] Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., and Mueller, A. "Scikit-Learn." *GetMobile: Mobile Computing and Communications*, Vol. 19, No. 1, 2015, pp. 29–33. https://doi.org/10.1145/2786984.2786995.

[51] Li, K., Yu, N., Li, P., Song, S., Wu, Y., Li, Y., and Liu, M. "Multi-Label Spacecraft Electrical Signal Classification Method Based on DBN and Random Forest." *PLoS ONE*, Vol. 12, No. 5, 2017, pp. 1–19. https://doi.org/10.1371/journal.pone.0176614.

[52] Zhang, R., Li, B., and Jiao, B. "Application of XGboost Algorithm in Bearing Fault Diagnosis." *IOP Conference Series: Materials Science and Engineering*, Vol. 490, No. 7, 2019. https://doi.org/10.1088/1757-899X/490/7/072062.

[53] Wu, Z., Wang, X., and Jiang, B. "Fault Diagnosis for Wind Turbines Based on

ReliefF and EXtreme Gradient Boosting.” *Applied Sciences (Switzerland),* Vol. 10, No. 9, 2020. https://doi.org/10.3390/app10093258.

[54] Sapountzoglou, N., Lago, J., De Schutter, B., and Raison, B. “A Generalizable and Sensor-Independent Deep Learning Method for Fault Detection and Location in Low-Voltage Distribution Grids.” *Applied Energy*, Vol. 276, No. April, 2020, p. 115299. https://doi.org/10.1016/j.apenergy.2020.115299.

[55] Chen, D., Yang, S., and Zhou, F. “Transfer Learning Based Fault Diagnosis With.” *Sensors (Switzerland)*, 2019, pp. 19–24.

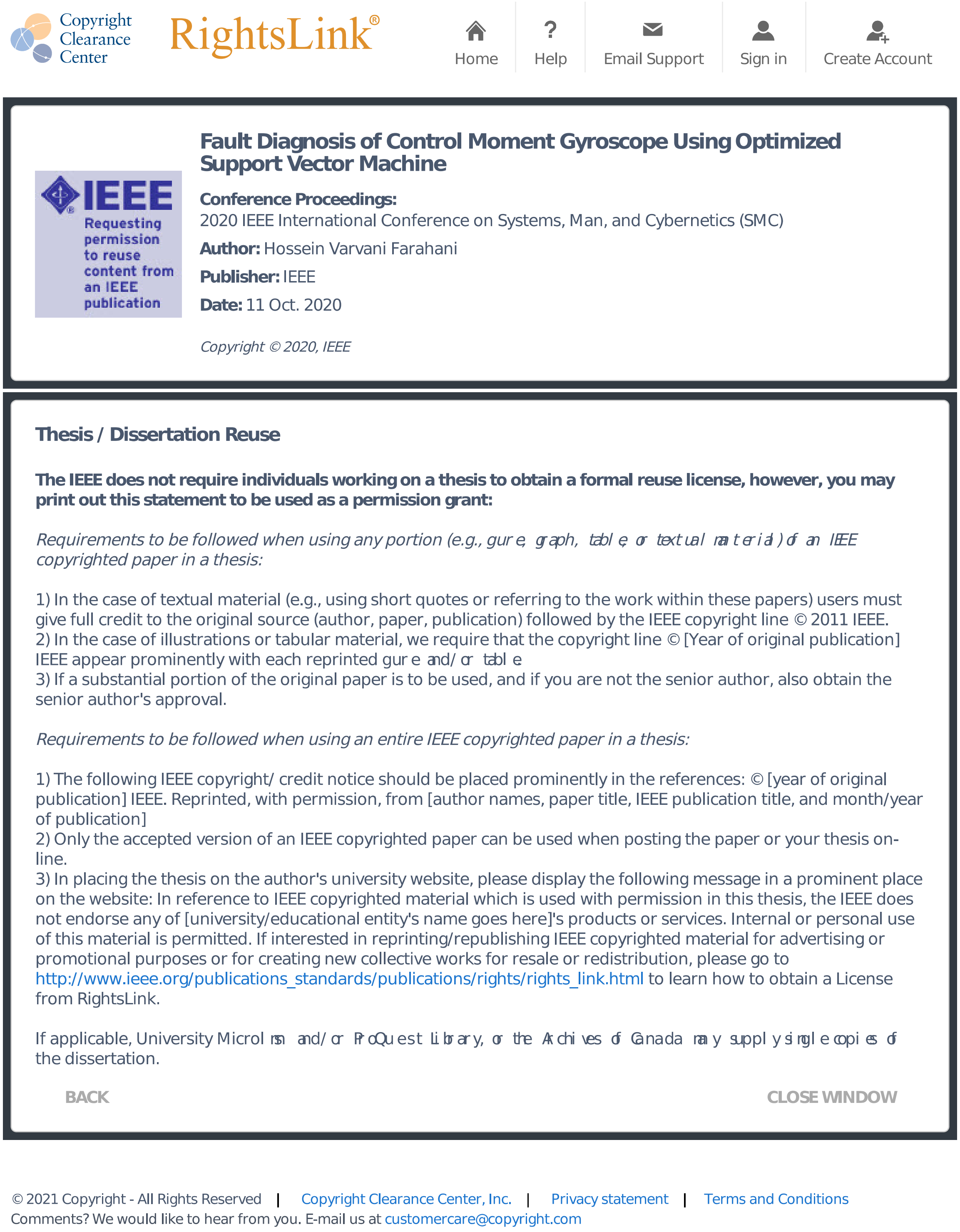
[56] Zhou, J., and Huang, Z. “Recover Missing Sensor Data with Iterative Imputing Network.” *arXiv*, 2017.

[57] Li, C., Zhang, S., Qin, Y., and Estupinan, E. “A Systematic Review of Deep Transfer Learning for Machinery Fault Diagnosis.” *Neurocomputing*, Vol. 407, 2020, pp. 121–135. https://doi.org/10.1016/j.neucom.2020.04.045.

[58] Li, W., Gu, S., Zhang, X., and Chen, T. “Transfer Learning for Process Fault Diagnosis: Knowledge Transfer from Simulation to Physical Processes.” *Computers and Chemical Engineering,* Vol. 139, 2020. https://doi.org/10.1016/j.compchemeng.2020.106904.

APPENDICES

Appendix A: Copyright Permissions


Copyright Clearance Center
RightsLink®
Home
Help
Email Support
Sign in
Create Account
IEEE
Requesting permission to reuse content from an IEEE publication
BACK
CLOSE WINDOW

Fault Diagnosis of Control Moment Gyroscope Using Optimized Support Vector Machine

Conference Proceedings:
2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)

Author: Hossein Varvani Farahani

Publisher: IEEE

Date: 11 Oct. 2020

Copyright © 2020, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

© 2021 Copyright - All Rights Reserved | Copyright Clearance Center, Inc. | Privacy statement | Terms and Conditions
Comments? We would like to hear from you. E-mail us at customercare@copyright.com

VITA AUCTORIS

NAME: Hossein Varvani Farahani

PLACE OF BIRTH: Arak, Iran

YEAR OF BIRTH: 1981

EDUCATION: Imam Khomeini High School, Booshehr, Iran, 1995-1999

K. N. Toosi University of Technology, B.Sc. of Mechanical Engineering, Tehran, Iran, 1999-2003

Arak Azad University, M.Sc. of Mechanical Engineering, Arak, Iran, 2008-2011