Autonomous Diagnosis of Satellite Sensor Anomalies to Ensure Fault Tolerant Control

Ulrich Louw

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electronic) in the Faculty of Engineering at Stellenbosch University.

Supervisor: Dr H. W. Jordaan
Co-Supervisor: Dr J. C. Schoeman
Department of Electrical and Electronic Engineering

March 2023

Acknowledgements

I would like to thank the following people for their support during my Master's.

- My wife, Danielle Louw, for her love, support and patience.
- My supervisor, Willem Jordaan, and my co-supervisor, JC Schoeman, for their wisdom, valuable insight and advice.
- Benjamin Evans and Raynhardt van Zyl for their practical advice during the execution of this thesis.
- My family and my church (Sola Gemeente Stellenbosch) for their prayers and support.
- My Lord and Saviour, Jesus Christ, for enabling me to finish this thesis. Soli Deo gloria.

UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
   Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
   I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.
   I also understand that direct translations are plagiarism.

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
   Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
   I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / *Student number*	Handtekening / *Signature*
UJ Louw Voorletters en van / *Initials and surname*	05/11/2022 Datum / *Date*

Abstract

English

Many satellites rely on attitude estimation algorithms to fuse information from multiple sensor sources. Some of these sensors might encounter false and erroneous measurements due to practical disturbances. In this thesis, methods are developed to increase the reliability of an extended Kalman filter when practical environmental anomalies occur. This is done by firstly detecting whether the current sensor measurements are anomalous and thereafter classifying the anomalous sensor. Different recovery methods are compared based on varying prediction accuracies. It was concluded that for prediction accuracies above 90%, the recovery technique that omits anomalous sensor measurements from the extended Kalman filter, is the best proposed recovery method. The random forest algorithm performs the best for isolating anomalies and using it separately or in conjunction with the local outlier factor algorithm for anomaly detection provide similar results. Implementing the local outlier factor algorithm increases the time efficiency. From simulations, it was found that a satellite attitude determination and control system that can predict and isolate practical anomalies improves the robustness and accuracy of the estimated attitude.

Afrikaans

Baie satelliete is afhanklik van oriëntasie afskattings algoritmes om die informasie van verskeie sensore te vereenselwig. Sommige sensore kan vals of foutiewe lesings ervaar weens praktiese versteurings. In hierdie tesis word metodes ontwikkel wat die betroubaarheid van die uitgebreide Kalman filter vergroot wanneer praktiese omgewings versteurings voorkom. Dit word gedoen deur eerstens die sensor anomalie waar te neem en daarna die verantwoordelike sensor te klassifiseer. Verskeie regstellende metodes word vergelyk met ’n variënde akkuraatheid. Die gevolgtrekking is dat vir akkuraathede bo 90% is die beste regstellende tegniek, die een wat slegs sensor afmetings verwyder van die Kalman filter se afmeting opdatering. Die lukrake woud algoritme is die beste isolasie metode en om dit te gebruik in samehorigheid met die lokale uitskietsel faktor algoritme vir anomalie waarneming of daarsonder produseer dieselfde resultate. Deur die lokale uitskietsel faktor algoritme te gebruik verhoog die tyd effektiwiteit van die stelsel. Deur die gebruik van simulasie is dit gevind dat ’n sisteem wat die praktiese anomalieë kan klassifiseer, verbeter die robustheid en akkuraatheid van die oriëntasie afksatting.

Contents

Declaration ii

Abstract iii

List of Figures vi

List of Tables x

Nomenclature xi

1. Introduction 1
- 1.1. Research Motivation . . . 1
- 1.2. Solution Overview . . . 3
- 1.3. Thesis Outline . . . 5

2. Literature Study 6
- 2.1. Fault Detection, Isolation and Recovery in Time-Varying Systems . . . 6
- 2.2. Fault Tolerant Control of Satellites . . . 8
- 2.3. Fault Tolerant Control with Focus on Sensor Anomalies . . . 9
- 2.4. Innovation in Fault Detection . . . 10
- 2.5. Summary . . . 11

3. Conceptual Satellite Mission 12
- 3.1. Standard Aerospace Industry Implementation . . . 12
- 3.2. Conceptual CubeSat Design . . . 13
- 3.3. Sensor Parameters . . . 15
- 3.4. Actuator Parameters . . . 15
- 3.5. Summary . . . 15

4. Conceptual Satellite Simulation Environment 17
- 4.1. Satellite Fundamentals . . . 17
- 4.2. Environmental Factors and Disturbances . . . 23
- 4.3. Attitude Determination and Control System . . . 33
- 4.4. Simulation Overview . . . 40
- 4.5. Summary . . . 43

5. Anomalies **44**
5.1. Reflection of Solar Panels on Sun Sensor . . . 44
5.2. The Moon in Field of View of the Horizon Sensor . . . 48
5.3. Magnetic Moment Disturbance . . . 55
5.4. Higher Noise on Sensor Measurements . . . 59
5.5. Summary . . . 61

6. Anomaly Detection and Isolation **62**
6.1. Moving Average of the Estimated Sensor Measurements . . . 63
6.2. Local Outlier Factor Algorithm . . . 70
6.3. Decision Tree Algorithm . . . 76
6.4. Random Forest . . . 82
6.5. Support Vector Machines Algorithm . . . 86
6.6. Computational Efficiency . . . 90
6.7. Summary . . . 91

7. Anomaly Recovery **93**
7.1. Recovery Methods . . . 93
7.2. Analysis of Recovery Methods . . . 95
7.3. Robustness of Recovery Methods . . . 99
7.4. Summary . . . 101

8. Results of Fault Detection, Isolation and Recovery Techniques **103**
8.1. Comparison of Feature Extraction Methods . . . 103
8.2. Integration of the Recovery Method . . . 106
8.3. Fault Detection and Isolation on General Actuator Anomalies . . . 111
8.4. Fault Detection, Isolation and Recovery on General Sensor Anomalies . . . 113
8.5. Summary . . . 116

9. Conclusion **118**
9.1. Summary and Evaluation . . . 118
9.2. Observations and Findings . . . 120
9.3. Future Work . . . 120

Bibliography **122**

List of Figures

1.1. On-orbit failures categorised by the subsystem responsible for the mission failure . . . 1
1.2. System block diagram of the ADCS with the additional FDIR technique . 4

2.1. Development of FDIR technique according to *expert-system-based* approach 6
2.2. Development of FDIR technique according to *data-driven* approaches . . . 7
2.3. Overview diagram of sub categories of fault tolerant control of satellites . . 9
2.4. Comparison of specific anomaly examples and general anomalies 11

3.1. The percentage of occurrences of different ADCS sensors on nano satellites 12
3.2. The percentage occurrences of different ADCS actuators on nano satellites 13
3.3. The conceptual satellite with solar panels 14

4.1. Illustration of orbit parameters and the Earth Inertial Coordinate frame . . 20
4.2. Euler angle transformation of an object 20
4.3. Graphical representation of a quaternion to explain the relationship between a quaternion and the Euler angles 22
4.4. Geometry to determine satellite eclipse phases 26
4.5. Geomagnetic field of the Earth 27
4.6. The simulated gravity gradient torque in the SBC frame 28
4.7. The simulated aerodynamic torque in the SBC frame 30
4.8. Modelling of the static reaction wheel imbalance 31
4.9. Modelling of the dynamic reaction wheel imbalance 32
4.10. The simulated wheel disturbance torques in SBC 32
4.11. The measured unit vector of the Sun in the SBC frame 34
4.12. Estimation metric in degrees during the first two orbits during normal operation 38
4.13. The efficacy of the quaternion feedback controller and the EKF provided as the pointing metric 40
4.14. Magnetic control torques during the implementation of momentum dumping 41
4.15. Simulation block diagram 42

5.1. Modelling for the Sun reflection from the solar panels unto the sun sensor . 45
5.2. Measured sun vector with the Sun reflections in SBC 47
5.3. Estimation metric during the first two orbits due to the Sun reflection anomaly 47

5.4. Geometry of the relationship between the Earth, the Moon and the satellite 48
5.5. Plane perpendicular to the satellite-to-Earth vector and at the centre of the Earth . . . 50
5.6. Projection of the Moon, the Earth and the horizon sensor's field of view on the geometric plane . . . 51
5.7. Visual representation of the circular fit algorithm for the horizon sensor . . 52
5.8. The Earth unit vector in SBC due to the Moon on the Earth's horizon anomaly . . . 54
5.9. Estimation metric during the first two orbits due to the Moon on the Earth's horizon anomaly . . . 54
5.10. Dipole moment due to current in circular loop of circuit in solar panel . . . 55
5.11. Shadow created by CubeSat body on solar panels . . . 56
5.12. Magnetic moment disturbance torque in SBC . . . 57
5.13. Magnetometer unit vector in SBC due to the magnetic moment disturbance 58
5.14. Estimation metric during the first two orbits due to magnetic moment disturbance . . . 58
5.15. Sun sensor measurement with high noise as anomaly . . . 59
5.16. Estimation metric for the first two orbits due to high sun sensor noise without any recovery . . . 60
5.17. Average estimation metric per orbit for 30 orbits of reaction wheel failure without any recovery . . . 61

6.1. The grouping of the various algorithms and methods into the subgroups of feature extraction, detection and isolation . . . 63
6.2. The input and output of the moving average of the estimated sensor measurements method . . . 67
6.3. Summation of absolute values in the moving average during first two orbits during the Sun reflection anomaly . . . 68
6.4. Summation of absolute values in the moving average during first two orbits during the Sun reflection anomaly . . . 68
6.5. Summation of absolute values in the moving average during first two orbits during the magnetic moment disturbance anomaly . . . 69
6.6. Summation of absolute values in the moving average during first two orbits during the magnetic moment disturbance anomaly . . . 69
6.7. Local outlier factor of a random data set to demonstrate the outlier score produced by the local outlier factor algorithm . . . 71
6.8. Reachability distance of the object p_o with $k = 5$ . . . 71
6.9. Local outlier factor during the first two orbits of the Sun reflection anomaly 73

6.10. Local outlier factor during first two orbits of the magnetic moment disturbance anomaly . . . 74
6.11. The input and output of the local outlier factor algorithm for anomaly detection . . . 75
6.12. The input and output of the decision tree algorithm for anomaly detection 78
6.13. The decision tree algorithm as a binary classification of the Sun reflection anomaly . . . 79
6.14. Average estimation metric per orbit of decision tree for both the sun sensor anomaly and the magnetic moment disturbance anomaly . . . 81
6.15. Average estimation metric per orbit of random forest for both the sun sensor anomaly and the magnetic moment disturbance anomaly . . . 85
6.16. Support vector machine example of splitting data into two groups based on a hyperplane . . . 87
6.17. Average prediction accuracy per orbit of support vector machines as anomaly detection for both the Sun reflection and magnetic moment disturbance anomaly . . . 89

7.1. Comparison of the average estimation metric per orbit for each recovery method during the Sun reflection anomaly . . . 96
7.2. Comparison of the average estimation metric per orbit for each recovery method during the magnetic moment disturbance anomaly . . . 96
7.3. Comparison of the average estimation metric per orbit for *EKF-ignore* recovery method and with no recovery during the magnetic moment disturbance anomaly . . . 97
7.4. Comparison of the average estimation metric per orbit for *EKF-ignore* recovery method and with no recovery during the magnetic moment disturbance anomaly with the rearrangement of the sequence of the sensor update . . . 98
7.5. Comparison of the average estimation metric per orbit for the *EKF-ignore* method with varying prediction accuracies during the Sun reflection anomaly 100
7.6. Comparison of the average estimation metric per orbit for the *EKF-reset* method with varying prediction accuracies during the Sun reflection anomaly 101
7.7. Comparison of the average estimation metric per orbit for the *EKF-combination* method with varying prediction accuracies during the Sun reflection anomaly . . . 102

8.1. The random forest algorithm with and without an additional feature $\mathbf{V}_k$ as input . . . 107
8.2. Comparison of average estimation metric per orbit of moving average as additional input and no additional feature to the random forest isolation method . . . 108

8.3. Comparison of average isolation accuracy per orbit of moving average as additional input and no additional feature to the random forest isolation method 108
8.4. The two different architectures for classifying a data sample 110
8.5. Comparison of average estimation metric per orbit of LOF as detection method or without any additional detection method to the random forest isolation method 110
8.6. Comparison of average prediction time for each time step per orbit of LOF as detection method and the random forest isolation method without any additional detection method 111
8.7. Average prediction accuracy per orbit of LOF as detection method and random forest as isolation method for reaction wheel failure 112
8.8. Average isolation accuracy per orbit of LOF as detection method and random forest as isolation method for reaction wheel failure 113
8.9. The three different ways of training and classification of the random forest algorithm for isolation 114
8.10. Comparison of average estimation metric per orbit of the different training methods of the random forest algorithm as isolation 116

List of Tables

3.1. Physical dimensions of the simulated CubeSat 14
3.2. Standard deviation for each sensor 15

4.1. Description and definition of the Earth orbit parameters 24

6.1. An example of a confusion matrix 75
6.2. Comparison of confusion matrices of the local outlier factor algorithm with varying contamination values 76
6.3. Confusion matrix of the decision tree for detection on a test data set . . . 80
6.4. Confusion matrix of isolation for the decision tree algorithm 82
6.5. Confusion matrix of the random forest for detection on a test data set . . . 84
6.6. Confusion matrix of isolation for the random forest algorithm 86
6.7. Confusion matrix of the support vector machines for detection on a test data set 89
6.8. Confusion matrix of isolation for Support Vector Machines 90
6.9. Average time per prediction method for both anomaly detection and anomaly isolation 91

8.1. Confusion matrices for anomaly detection with moving average as additional feature input 104
8.2. Confusion matrices for anomaly isolation with moving average as additional feature input 105
8.3. Confusion matrices for anomaly detection with outlier score as additional feature input 105
8.4. Confusion matrices for anomaly isolation with outlier score as additional feature input 106
8.5. Confusion matrix for random forest algorithm trained on both general and orbit sensitive practical modelled anomalies that are considered as separate 114
8.6. Confusion matrix for random forest algorithm trained on both general and orbit sensitive practical modelled anomalies that are considered as grouped 115
8.7. Confusion matrix for random forest algorithm trained on only the general anomalies 115

Nomenclature

Notation

r	Scalar.
$\mathbf{r}$	Vector.
$\bar{\mathbf{r}}$	Unit vector.
$\dot{\mathbf{r}}$	Derivative of vector.
$\|\mathbf{r}\|$	Norm of vector.
$\mathbf{r}^-$	Vector before a specified change.
$\mathbf{r}^+$	Vector after a specified change.
$\mathbf{r}_t$	Vector as a function of time.
$\mathbf{r}_k$	Vector at time step k.
$\mathbf{r}_1 \times \mathbf{r}_2$	Cross product between two vectors.
$\mathcal{A} = \{\bar{\mathbf{x}}_\mathcal{A},\ \bar{\mathbf{y}}_\mathcal{A},\ \bar{\mathbf{z}}_\mathcal{A}\}$	Coordinate frame specified by the set of unit vectors of each axis.
$\mathbf{r}_\mathcal{A}$	Vector within a coordinate frame $\mathcal{A}$.
r_x	X component of a vector.
$\mathbf{R}$	Matrix.
$\mathbf{R}^\mathrm{T}$	The transpose of a matrix.
$\mathbf{I}_{n\times n}$	An identity matrix of size n.
$\boldsymbol{R}^\mathcal{B}_\mathcal{A}$	Transformation matrix from the coordinate frame $\mathcal{A}$ to the coordinate frame $\mathcal{B}$.
$\mathbf{r}^\mathcal{B}_\mathcal{A}$	Change of vector $\mathbf{r}_\mathcal{B}$ as seen from the coordinate frame $\mathcal{A}$.

Acronyms and abbreviations

ADCS	Attitude determination and control system
AP	Argument of perigee
BST	Binary search tree
CART	Classification and regression tree
CDH	Command and data handling
CoM	Centre of mass
CoP	Centre of pressure
DCM	Direct cosine matrix
DMD	Dynamic mode decomposition
EIC	Earth inertial coordinate
EFC	Earth fixed coordinate
EKF	Extended Kalman filter
FDIR	Fault detection, isolation and recovery
FoV	Field of view
GHA	Greenwich hour angle
GPS	Global positioning system
IAGA	International association of geomagnetism and aeronomy
IGRF	International geomagnetic reference field
IR	Infrared
IRC	Inertial-referenced coordinate
LEO	Low Earth orbit
LOF	Local outlier factor
ORC	Orbit-referenced coordinate
RAAN	Right ascension of the ascending Node
RW	Reaction wheel
SAVMA	Summation of the absolute values in the moving average
SBC	Satellite body coordinate
SGP	Simplified general perturbation
SVM	Support vector machine
SVMA	Summation of the values in the moving average
TLE	Two-line element
TTC	Telemetry, tracking and control

Chapter 1

Introduction

1.1. Research Motivation

Since all systems are prone to failure, an engineer needs to ensure that crucial systems are robust to failure and that proper testing and continual maintenance are performed. In the case of satellites, the need to design a system that is robust to failure is even more crucial, since most failures are unrecoverable and may lead to mission failure. Satellite systems must therefore be tested thoroughly and be robust to most common anomalies. According to M. Tafazoli [1] the attitude determination and control system (ADCS) contributes to the largest percentage of satellite failures as shown in Figure 1.1. A study conducted by S. Jacklin [2] on small satellite mission failures provides a deeper insight into the ADCS's role in satellite failures, since most missions are highly dependent on the ADCS for complete mission success. This is due to many mission requirements relying on accurate control of the satellite to provide the desired orientation for the operation of payloads.

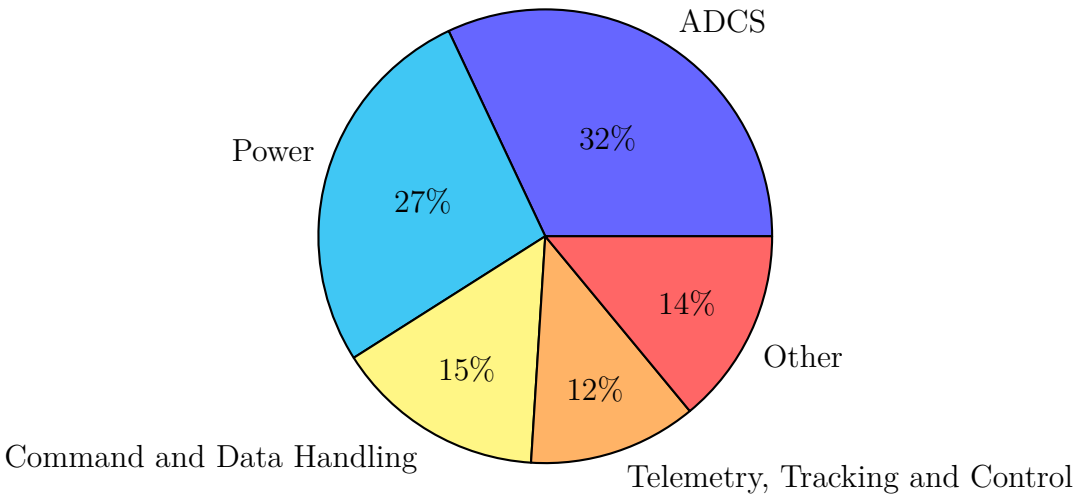

Figure 1.1: On-orbit failures categorised by the subsystem responsible for the mission failure [1].

A benchmark database provided by M. Swartwout [3], containing historical data of satellite launches, demonstrates the increase in the number of satellites launched every year. The drastic increase of launched satellites over the past few years further emphasizes this need of ensuring that satellites are robust to failures. Using traditional methods

of tracking satellites with ground stations and manually checking for possible failures are therefore not feasible, especially in the case of large satellite constellations [4]. Most subsystems of the satellite are therefore required to operate autonomously, especially in the case of the ADCS. The focus of this thesis is on a specific aspect of the ADCS which is sensitive to anomalies, namely the attitude estimation.

The extended Kalman filter (EKF) is a sensor fusion algorithm implemented for attitude estimation of the satellite. The EKF uses the sensors to provide a measurement update of the estimated state vector. The ADCS is therefore sensitive to sensor anomalies as this influences the estimated state vector which in turn influences the attitude control. The change in the estimated state vector depends on the severity of the sensor anomaly.

If an erroneous, or false measurement is present in the collection of sensors, it might deter and influence the outcome of the fusion algorithm. Depending on the number of sensors available and the severity of the erroneous sensor, the EKF can be influenced by reducing the estimation accuracy, producing divergence or creating instability. It is good practice to develop appropriate tests to protect the EKF against incorrect measurements. Since the EKF is reliant on the mathematical model of the system, the control inputs should also be accurate. Due to actuator failure on satellites, the command control input and the actual control input can differ significantly. This must be detected and isolated to ensure robust estimation.

Fault detection, isolation and recovery (FDIR) is required for the satellite operation, where the fault is any anomaly or failure that reduces the efficiency of the system. The detection phase produces a binary output on whether the system is currently operating with a fault or not. The isolation phase is required to classify the current fault into subsystems or components that are responsible for the fault. The recovery phase is responsible for ensuring that the fault's influence on the system is reduced. The satellite must be able to autonomously detect, isolate and recover from common anomalies to ensure safe operation during orbit. Practical sensor and actuator anomalies influence the estimation of the EKF that is commonly used in the ADCS of satellites. To ensure that these sensor anomalies can be recovered from, this thesis focusses on different FDIR methods to provide robust estimation of the EKF even with practical sensor and actuator anomalies.

Previous research on anomaly detection in satellite sensors have been investigated where the current trend is to use generic sensor anomalies, such as bias drift, high noise, sudden failure or any drastic change in the behaviour of the sensor to develop techniques to detect these anomalies [5–10]. This is only a subset of possible errors and does not assist in diagnosing the anomaly, detecting intermittent errors, or coupled events between sensors. An example of a practical anomaly which can occur and which is difficult to detect using standard techniques are solar reflections from solar panels on a sun sensor. The majority of satellites, even with relatively low attitude requirements, have some form of sun sensor. The sun sensor provides an accurate measurement during the periods of the

orbit where targeting and solar tracking is most likely and where the attitude requirement is the highest. Thus, it would be beneficial to have good interventions to ensure robust sun vector measurements for the EKF.

1.2. Solution Overview

This project aims to develop and test various methods of detecting sensor anomalies with a satellite's ADCS, and isolating the sensor responsible for the anomaly. The FDIR techniques are tested in a simulation model wherein practical anomalies are simulated. The simulation is also used to create a database of sensor measurements produced by different anomalies. This database provides labelled data for the training of binary and multi-class isolation models for detection and isolation, respectively (where multi-class refers to more than two classes). The trained models should be tested on the simulation environment and the estimation accuracy should be compared between different models and different recovery methods.

The solution overview can be visually demonstrated with the system block diagram provided in Figure 1.2. Each block within Figure 1.2 will be individually discussed and analysed as far as possible. The simulated environment will be built based on a conceptual design of the satellite. This is required to create a realistic environment and provide the current state vector $\mathbf{x}_k$ for the modelling of the anomalies as well as the sensors. This in turn provides the sensor measurements $\mathbf{v}_b$ as inputs to the EKF. The estimated state vector $\hat{\mathbf{x}}_k$ from which the controller can provide accurate control torques, $\boldsymbol{\tau}_c$, to the simulated environment.

It is within this realistic simulation that anomalies can be modelled and create disturbances to the sensors of the satellite. It is at this point where the FDIR is required to ensure robust and accurate estimation of the EKF during anomalous periods. The feature extraction method provides addition features to the detection and isolation models to determine whether an anomaly occurred and which sensor is responsible for the anomaly. The recovery technique can thereafter incorporate techniques to lower the influence of the anomaly on the EKF.

For the FDIR techniques there are multiple different techniques and methods for detecting and isolating anomalies. Learning methods are frequently used in the current research trend of FDIR methods. The testing of these learning methods do not focus on practical orbit sensitive sensor anomalies. It is for this reason that learning methods are therefore selected for the implementation and analysis of anomaly detection and isolation.

One aim is to develop an unsupervised learning detection algorithm that is only trained on the "normal data", wherein no anomalies occur. This detection algorithm is required to label data samples as anomalies where the relationship between the sensors are considered as an outlier. Although this is one of the aims of this thesis, supervised

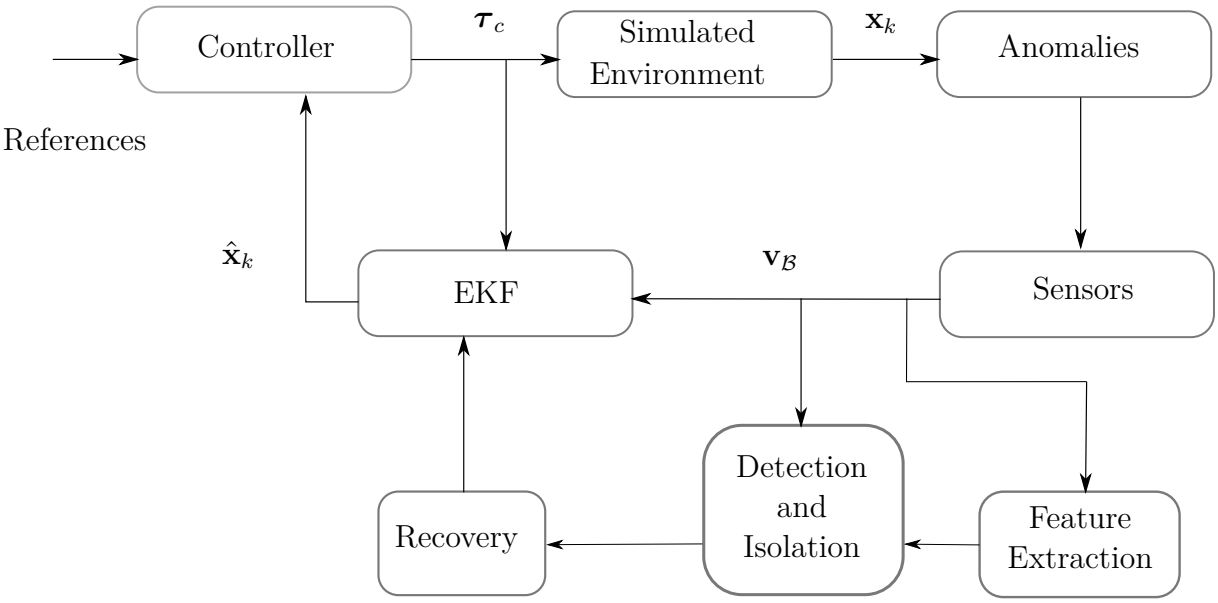

Figure 1.2: System block diagram of the ADCS with the additional FDIR technique.

learning methods will also be used for detection to provide a comparison between the two groups. Unsupervised learning for detection is desired since it only requires normal data for training and provides a method of detecting anomalies based on a pattern of what normal is. This means that various anomaly simulations are not required for the detection method.

The process of classifying the sensor responsible for the anomaly is the isolation phase of the FDIR. Anomaly isolation requires labelled data, since the isolation method needs to determine which sensor is faulty. Supervised learning methods are therefore used to isolate the sensor that is experiencing the anomaly.

A comparison of the detection accuracy as well as the estimation accuracy of a model in the simulation environment after training on either the generic sensor anomalies or the practical anomalies are discussed in Chapter 8. A thorough analysis of each individual component of the FDIR should be conducted and discussed. In summary the objectives of this thesis can be listed as:

- Develop a realistic simulation environment for testing of FDIR methods as well as providing a database for training classification algorithms.
- Model specific anomalies for each sensor used in the measurement update of the EKF.
- Test various recovery methods to ensure the robustness of the EKF during the occurrence of a sensor anomaly.
- Compare various supervised and unsupervised learning methods for detection and isolation of the anomalies.

- Compare the developed FDIR method trained on general sensor anomalies such as high noise with the FDIR method trained on the specific modelled anomalies for each sensor.

1.3. Thesis Outline

The overview of the system developed for FDIR of the ADCS is shown in Figure 1.2. This thesis will discuss each block in Figure 1.2 and analyse the results of these blocks in isolation. The thesis is outlined in order to ensure that the research motivation is accomplished. To provide the relevant context of this research within the wider field of FDIR the literature study is provided in Chapter 2. Chapter 2 also provides the background of the current research trends such as focusing on general sensor anomalies instead of common practical anomalies that are specific for each sensor. Chapter 3 provides the overview of the conceptual satellite design to ensure that the simulation is realistic and to provide context of the satellite design and mission requirements. This sets the stage for the discussion on the development of the simulation environment in Chapter 4 with the development of the key elements required for the ADCS environment. This provides the analysis for the normal operation of the satellite. The mathematical models of the anomalies for each sensor is discussed in Chapter 5 after the development of the simulation environment.

The anomalies require FDIR techniques to ensure that the anomalies do not negatively influence the estimation of the EKF. The order in which the FDIR is discussed is in the order of implementation. The feature extraction, anomaly detection and anomaly isolation are discussed and analysed in Chapter 6 after the development of the anomalies. Thereafter the recovery methods are developed and discussed in Chapter 7 with perfect detection and isolation. Chapter 6 and Chapter 7 provides the analysis of the FDIR methods in isolation. This can then be implemented on the various anomalies and the results is discussed in Chapter 8. Thereafter the discussion of modelling orbit sensitive practical anomalies as training data for the isolation method and the conclusion of this thesis is provided in Chapter 9.

Chapter 2

Literature Study

Fault detection, isolation and recovery (FDIR) is implemented in systems that require robust operation and desires continual functionality. For this thesis, FDIR for time-varying systems which require fault tolerant control, are relevant as this is the broader research field. It is therefore necessary to discuss the research done in the wider field of fault tolerant control before focusing on FDIR for only satellites. Thereafter, the research done in FDIR on satellites and, specifically the ADCS, will be discussed. An even narrower research space is fault tolerant control with the focus on recovery of Kalman filters due to faults caused by sensor anomalies.

2.1. Fault Detection, Isolation and Recovery in Time-Varying Systems

Anomaly detection is a field in robotics which can be divided into three main categories, namely *data-driven*, *model-based* and *expert-system-based* approaches [11, 12]. The *expert-system-based* approach is implemented with human knowledge based logic tests or rules, also known as rule-based [13, 14]. *expert-system-based* approaches are traditionally implemented in many FDIR systems. An expert can provide certain rules based on experience and knowledge of the field to ensure that the system is behaving as expected. A simplified and generic diagram thereof is shown in Figure 2.1.

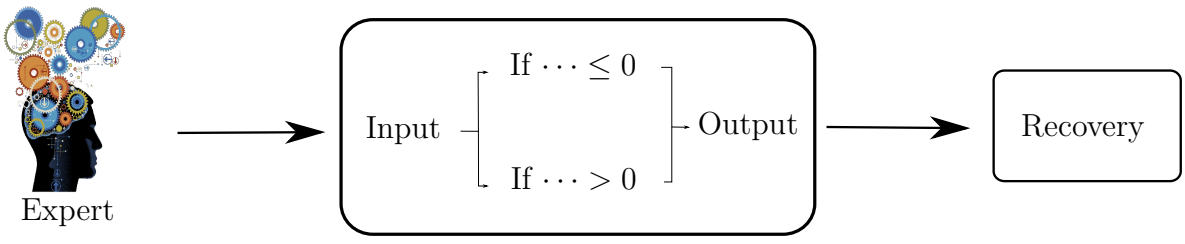

Figure 2.1: Development of FDIR technique according to *expert-system-based* approach. Where the human expert provides logical rules based on the expert's experience within the field to detect and isolate faults or anomalies.

Expert-system-based methods are, however, very closely aligned to model based methods and the lines between the two categories are sometimes blurred. This is due to many

expert-system-based methods relying on a model and sensor inputs to ensure that the system is behaving as required. This is why many *expert-system-based* methods compare sensor measurements with a mathematical model of a given sensor measurement to develop a fault detection system.

Expert-system-based methods are required to create different rules for each sensor's implementation [13, 14]. Focused tests with in-depth knowledge of the nominal operation of the unit can eliminate many fault conditions. Unfortunately, these tests are normally limited to the perspective of each sensor and does not take any other sensors or the state of the satellite into consideration. *Data-driven* approaches allows one to classify complex anomalies by training models on data across many sources specific to the anomaly at hand. A diagram of the *data-driven* approach is shown in Figure 2.2.

In Figure 2.2 the database is provided as input to a training algorithm that after training produces a model that can predict whether the current data sample is anomalous or not. This can also be expanded to a model that can classify which component is responsible for the anomaly, if the algorithm is capable of providing a multi-class prediction. This approach requires a prediction model that is trained from the data. The prediction model can consist of any algorithm from a wide repertoire of algorithms, such as K-nearest neighbours, long short term memory (LSTM), Kalman filters, decision tree and isolation forest [15–17]. These are used depending on the nature of the data. The decision of which method to implement relies heavily on whether the data is based on a time-series model or not. It also depends on whether labelled data is available, since certain methods require labelled data to predict whether or not a data sample is anomalous.

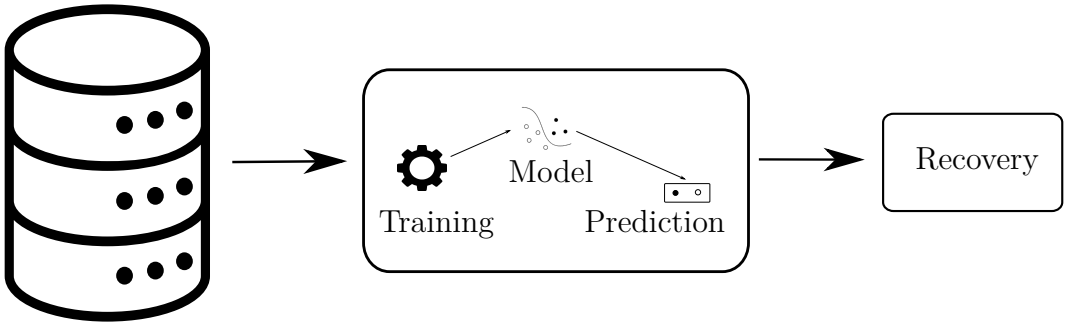

Figure 2.2: Development of FDIR technique according to *data-driven* approaches. The database is provided as input to a training algorithm that after training produces a model that can predict whether the current data sample is anomalous or not.

This database can be either simulated data or data from actual satellite missions. The problem with using actual satellite data is that it is difficult and expensive to get the specific data required for practical anomalies and to have it labelled. Many research therefore choose simulation environments to implement and test the developed methods for FDIR. It is within these simulations that anomalies can be simulated and FDIR techniques can be developed and tested.

2.2. Fault Tolerant Control of Satellites

Traditionally, the *expert-system-based* approach combined with the *model-based* approach dominated satellite fault detection. An expert in the field typically develops a rule-based system for each failure the expert is aware of [13]. The expert is also required to manually observe the satellite regularly. If the expert is able to observe anomalous data, the system must be adapted to the failure. This should be done for every subsystem of the satellite. Some subsystems are not as critical as others and, since many subsystems are dependent on the ADCS for mission success, the ADCS must be able to autonomously recover from faults to control the attitude of the satellite.

This leads to the need for fault tolerant control of satellites where the fault detection is specifically focused on the ADCS subsystem. The fault tolerant control can be separated into two categories, namely robust control to ensure accurate control during failed actuators or robust estimation developed typically for failed sensor or anomalous sensor events. The sensor or actuator fault is not necessarily a complete failure of the component or complete failure of the mission, but negatively influences the mission success of the satellite or the efficiency of a subsystem of the satellite. To ensure robust control after actuator failure, FDIR methods for reaction wheels, gyroscopes and other actuator failures were developed and focussed on the robust control and not the robust estimation.

Rahimi et al. [18] developed a data driven method for fault isolation which incorporates random forests, decision trees, and nearest neighbours to classify failures of reaction wheels. These failures were modelled for time-varying faults where the bus voltage or current of the reaction wheels were varied to induce failures. This, however, only focuses on isolating the failures and not recovering from these failures, whereas Jin et al. [19], implements a fault tolerant control with four reaction wheels based on bias faults implemented on the reaction wheels. On the other hand, a *model-based* technique that implements interacting multiple model filters was implemented to recover from modelled reaction wheel failures, such as viscous friction variations due to temperature [20]. Both Tudoroiu et al. [20] and Rahimi et al. [18] provide two different implementations for fault tolerant control based on practical modelled actuator failure on satellites, where Jin et al. [19] tests the provided method on a bias fault condition and not a practical modelled failure.

The research conducted on the robust control category can be divided into research that test methods on practical modelled failures of actuators or research which test their methods on general failures such as bias drift. This thesis attempts to develop a FDIR method for sensor anomalies and compare the developed method by training it on both general sensor anomalies and practical modelled sensor anomalies.

2.3. Fault Tolerant Control with Focus on Sensor Anomalies

Although the robust control aspect of fault tolerance is of high importance, the attitude estimation of the satellite should also be able to recover from anomalies to ensure a successful mission. Sensor and actuator failures should also be detected and recovered from, since it directly influences the attitude estimation. Most of the research conducted to ensure robust estimation only focuses on sensor faults. The main categories within fault tolerant control of satellites can be seen in Figure 2.3.

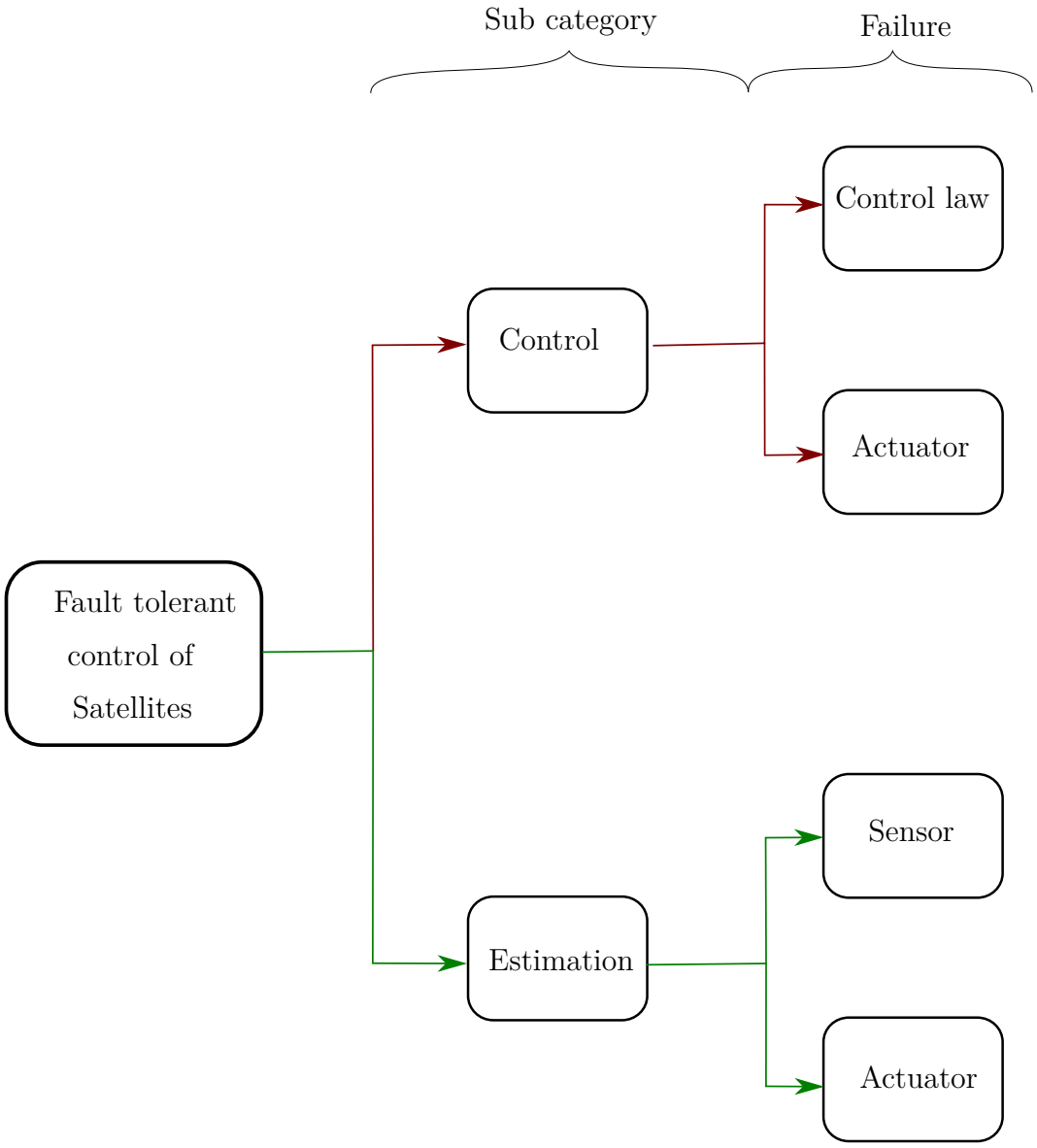

Figure 2.3: Overview diagram of sub categories of fault tolerant control of satellites. The green path provides the focus of this thesis.

Based on the assumptions of typical sensor failures, some work was done on the fault detection of attitude sensors. For instance, due to considerable noise in sensors, Wang

et al. [5] proposed an adaptive unscented Kalman filter with multiple-model adaptive estimation for sensor fault estimation and isolation. The performance of this method was tested on a simulation model where gradual failures, abrupt failures and high noise were implemented on the sensors. Xiong et al. [6] provide a fault detection method by using the residuals generated by an unscented Kalman filter to detect anomalies with a threshold based on a confidence level. This method was tested on a simulation environment where a sun sensor, horizon sensor and gyroscope was used for attitude determination. A sudden bias failure for the sun sensor, horizon sensor and gyroscope was implemented as well as an incipient fault on the sun sensor. Zhou et al. [7] implement a fault tolerant federated Kalman filter with three sub-filters for multi-sensor fault estimation. The failures of the sensors are measurements that are equal to $\mathbf{0}$, or experience constant bias faults or noise amplification. Nasrolahi et al. [8] provide a fault detection and recovery method by implementing a non-linear observer, to detect anomalies in attitude and rate sensors. The recovery was implemented through the tuning of controller gains after the classification of sensor failures. Another example was the development of an algorithm by Carvajal-Godinez et al. [9] to evaluate the control of a gyroscope and to detect whether drifting exists. If drifting was detected, another algorithm was deployed to ensure the recovery of the gyroscope drift by updating the error state vector. Van Eykeren et al. [10] developed an adaptive modification to the EKF with the testing thereof on an aircraft. The methods that are provided in these research articles were tested on faults such as oscillation, bias drift and increased noise of sensors.

All these research examples on sensor FDIR only test the techniques on general sensor faults and not specific modelled faults for each sensor. This thesis models practical anomalies for each sensor that are sensitive to the satellite orbit and not general anomalies as given in the examples [5–10]. This is visually demonstrated in Figure 2.4, where the norm of sensor fault detection is the red path of general anomalies, while the focus of this thesis is demonstrated by the green path, which test methods on practical modelled anomalies.

2.4. Innovation in Fault Detection

Detection of sensor anomalies and developing robust Kalman filters are not isolated to satellites. All systems governed by underlying physics have many similarities in the approach of fault detection. de Silva et al. [21] developed a novel method for feature extraction which focuses on systems governed by underlying physics. This method is based on the assumption that a complex relationship exists between different sensor measurements and that the next measurement for a sensor can be predicted based on the current sensor measurements. This leads to the development of an innovative moving average of the predicted sensor vector, as estimated with dynamic mode decomposition

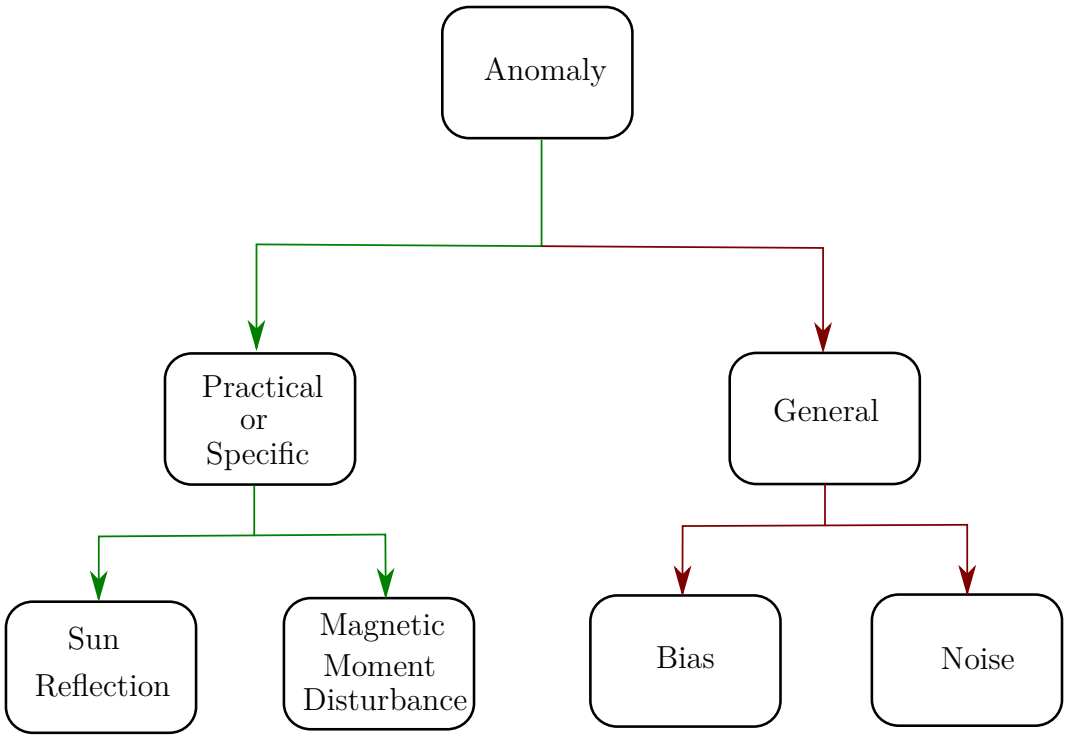

Figure 2.4: Comparison of specific anomaly examples and general anomalies. The green path provides the focus of this thesis.

(DMD) and a Kalman filter. This is provided as additional input to a predictive model – a decision tree – to detect sensor anomalies.

2.5. Summary

Methods based on the research of various authors [5–10] were tested on sensor failures that are not modelled by the orbital nature of the satellite or specific design failures. Within the field of fault tolerant control it is clear that are various approaches to ensure robust EKF model and measurements updates against anomalies and failures. These failures are, however, limited to sudden failure, bias drift, oscillations or an increase in sensor noise.

This work is an example of implementing the anomaly detection on the sensor level and diagnosing which sensor is experiencing the anomaly. Rather than building robust Kalman filters for any sensor failure and still updating the Kalman filter with an anomalous sensor measurement, the sensor measurement can, for instance, be excluded from the measurement update sequence.

Furthermore, the FDIR techniques in this thesis are not developed and tested on only generic sensor failures, but on specific practical failures and anomalies. This provides more relevant and specific analysis of the techniques and it can be extended to the testing of FDIR techniques on other modelled failures.

Chapter 3

Conceptual Satellite Mission

To simulate the ADCS of the satellite, the sensors and actuators which will be used for the CubeSat must be chosen. This provides a context for the satellite simulation and the influence of the anomalies. The design decisions are based on historic trends and previous implementation of CubeSats. The design decisions are also based on what is required for a mission which points a payload towards the Earth during the eclipse phase and pointing the solar panels towards the Sun during the sunlit phase. This mission is a realistic example where the mission success relies on the operation of the ADCS.

3.1. Standard Aerospace Industry Implementation

To developed a realistic conceptual design the historic trend is accounted for to enhance the relevance of this research. The trend from 1990 to 2015 of which sensors and actuators were implemented for in nano satellites the ADCS is provided in Figure 3.1 and Figure 3.2 [22]. A study conducted by S. Jacklin [2] provides examples of two partially failed missions that were due to sensor failure. One was due to magnetometer faults and the other due to sun sensor failure. An example of total mission failure is when both the magnetometer and sun sensor failed.

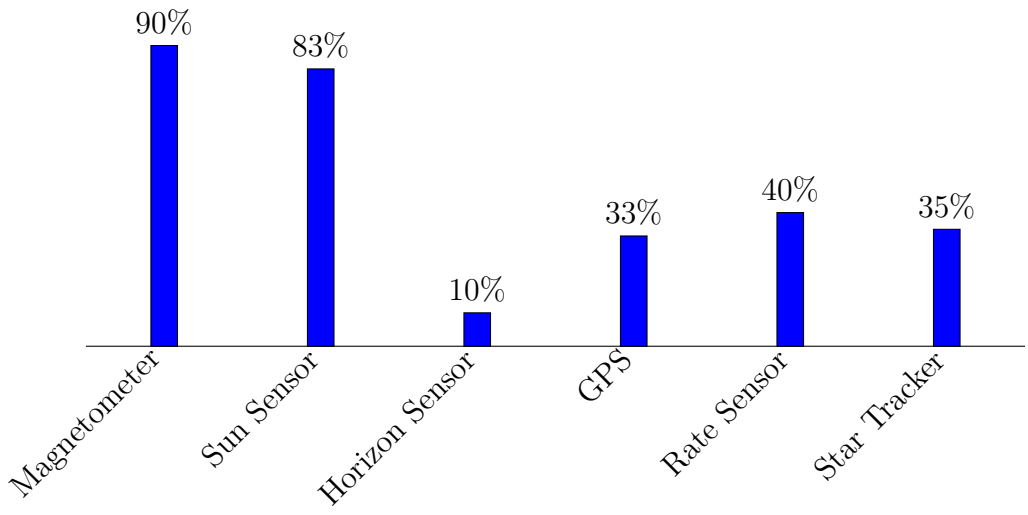

Figure 3.1: The percentage of occurrences of different ADCS sensors on nano satellites [22].

The percentage of ADCSs which utilize a variety of actuators is visualised in Figure 3.2. It is evident in Figure 3.2 that the two actuators which are utilized the most are magnetorquers and reaction wheels. The reaction wheels can have a significant influence on the estimation due to the quaternion feedback controller and are thus dependent on current attitude estimates. The magnetorquers will have a smaller influence, since it is mostly used for momentum dumping. In order for a satellite to point nadir and then to be able to track the Sun, it requires a number of sensors and actuators. The proposed simulated satellite will be equipped with a sun sensor, magnetometer and horizon sensor to determine its attitude. Reaction wheels and magnetorquer will also be simulated for accurate attitude control and momentum control respectively.

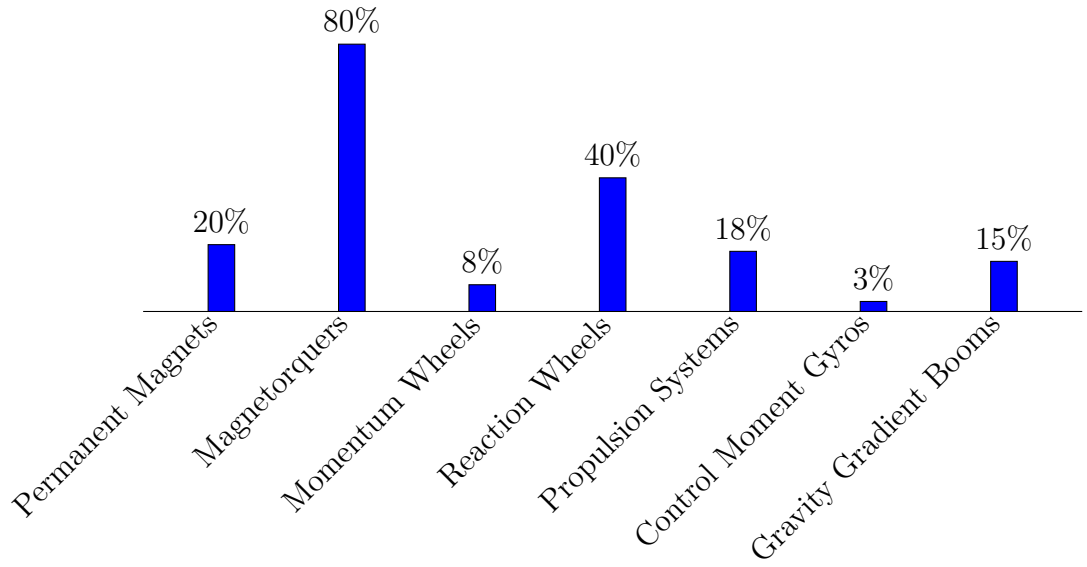

Figure 3.2: The percentage occurrences of different ADCS actuators on nano satellites [22].

3.2. Conceptual CubeSat Design

In order to ensure accurate modelling and simulation of both the satellite orbit and anomalies, the dimensions of the exterior of the CubeSat is required. The elements which require specific dimensions for the dynamics and kinematics are the solar panels and the satellite body.

Since most nano satellites implement a magnetometer and a sun sensor and both sensors cause partial or total failure of a satellite mission according to S. Jacklin [2], these two sensors are used in the simulation environment of this thesis and practical anomalies are modelled for these two sensors. The horizon sensor, although it is only utilized by 10% of nano satellite ADCS operations, has an interesting practical anomaly [23]. The anomaly occurs when the Moon is present on the Earth's horizon and in the horizon sensor's field of view (FoV). This might influence the algorithm for calculating the centre

of the Earth. This anomaly is therefore modelled and the influence of the anomaly on the sensor measurement is discussed in Chapter 5.

The sun sensor also requires specific dimensions for the Sun reflection anomaly discussed in Section 5.1. The dimensions of the sun sensor are from the Sputnix CubeSat sun sensor model [24]. A summary of the exterior dimensions are given in Table 3.1.

Directions	Satellite Body	Solar Panels	Sun Sensor
$\mathbf{x}_{\mathcal{B}}$	0.3 m	0.3 m	0.028 m
$\mathbf{y}_{\mathcal{B}}$	0.3 m	0.3 m	0.023 m
$\mathbf{z}_{\mathcal{B}}$	0.4 m	0.002 m	N/A

Table 3.1: Physical dimensions of the simulated CubeSat.

A solar panel is denoted as the surface area ABCD, as shown in Figure 3.3. This is referenced for the Sun reflection anomaly, as well as the magnetic moment disturbance anomaly as both anomalies are due to behaviours of the solar panel due to interaction with the Sun. This is once again provided to ensure accurate modelling of the satellite during normal operation and during practical modelled anomalies. The coordinate frames and the background thereof is discussed, in detail, in Section 4.1.1. The moment of inertia of the satellite can therefore be given as

$$\begin{bmatrix} \mathbf{I}_{xx} & \mathbf{I}_{yy} & \mathbf{I}_{zz} \end{bmatrix} = \begin{bmatrix} 0.4 & 0.45 & 0.3 \end{bmatrix} \text{kgm}^2. \tag{3.1}$$

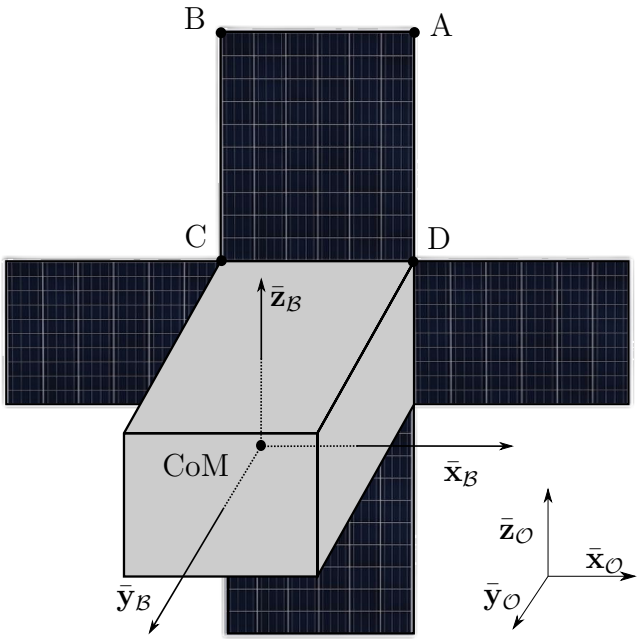

Figure 3.3: The conceptual satellite with solar panels.

3.3. Sensor Parameters

The sensor parameters are design decisions to ensure realistic performance of the sensors during normal operation and to create realistic responses thereof due to practical anomalies. The noise of all the sensors are assumed to be zero-mean Gaussian noise. To model the anomalies accurately, the position of the sensors are required. The standard deviation of the noise, as well as the position for each sensor in the satellite body coordinate (SBC) frame is therefore given in Table 3.2. The sensor model is discussed in further detail in Section 4.3.1 after the theoretical background of the satellite simulation.

Sensor	Standard deviation (%)	Position $\{\bar{\mathbf{x}}_{\mathcal{B}},\ \bar{\mathbf{y}}_{\mathcal{B}},\ \bar{\mathbf{z}}_{\mathcal{B}}\}$	FoV
Magnetometer	0.75 [25]	$\{0.1, 0.1, 0.13\}$ m	N/A
Horizon Sensor	0.14 [23]	$\{0.15, 0, -0.2\}$ m	180°
Sun Sensor	0.055 [24]	$\{0.15, 0, 0.2\}$ m	180°

Table 3.2: Standard deviation for each sensor.

3.4. Actuator Parameters

The actuator parameters are also required to provide accurate response to control torque outputs and the simulation of the control of the satellite during normal operation and during anomalies. The magnetorquers and reaction wheels both have maximum torque outputs, as well as parameters that influence disturbance models. The reaction wheel chosen is the RW3-0.06 from Sinclair Interplanetary [26] and the magnetorquer is the medium CubeTorquer by CubeSpace [27]. The parameters for that of the reaction wheel is the static and dynamic wheel imbalance required to simulate the wheel imbalance discussed in Section 4.2.3. The static imbalance and dynamic imbalance is given as $U_s = 0.0208$ gcm and $U_d = 0.0208$ gcm^2 respectively. The maximum torque output of the reaction wheel is 20 mNm and peak momentum is 0.18 Nms. The moment of inertia of the wheel is given as $I_w = 88.1 \times 10^{-6}$ kgm^2. The only relevant parameter of the magnetorquer is the maximum magnetic moment, which is 0.66 Am^2.

Three magnetorquers as well as three reaction wheels are implemented to ensure accurate and robust attitude control of the satellite. A single magnetorquer and reaction wheel are positioned in each axis of the SBC frame. This allows for three-axis rotation and control of the satellite.

3.5. Summary

This chapter provides the conceptual satellite design and mission overview for the simulation environment. The satellite is designed with reaction wheels and magnetorquers as actuators.

It is also provided with a sun sensor, magnetometer and a horizon sensor for attitude determination. This design is based on common practice as well as research on sensor failures. The satellite mission is to point a payload towards the Earth during an eclipse and point the solar panels towards the Sun during the sunlit phase. This chapter provides the context in which the normal operation of the satellite is simulated as well as the modelling of the practical anomalies and the analysis of the FDIR methods.

Chapter 4

Conceptual Satellite Simulation Environment

In order to implement practical anomalies and research FDIR methods on a satellite, a simulation that contains the satellite dynamics and kinematics needs to be developed. The focus of this thesis is directed towards nano satellites and, more specifically, CubeSats. The development of the satellite simulation is based on previous research [22, 28–30]. The simulation was developed in Python to simulate the dynamics and kinematics during a satellite orbit. This simulation environment also provides the environmental factors and disturbances of the satellite orbit to ensure a realistic simulation for the EKF and the testing of the FDIR methods.

4.1. Satellite Fundamentals

The main operational goal of the ADCS on this specific satellite mission is to control the payload so that it points towards the centre of the Earth during an eclipse and points the solar panels towards the Sun during the sunlit phase. In this section the different coordinate frames dominating a satellite orbit, the attitude of the satellite, as well as the satellite dynamics and kinematics are discussed.

4.1.1. Coordinate Frames

A coordinate frame $\mathcal{A}$ consists of three orthogonal vectors which is commonly referred to as $\bar{\mathbf{x}}_{\mathcal{A}}$, $\bar{\mathbf{y}}_{\mathcal{A}}$, and $\bar{\mathbf{z}}_{\mathcal{A}}$. A vector $\mathbf{r}_{\mathcal{A}}$ within the coordinate frame can thus be expressed as

$$\mathbf{r}_{\mathcal{A}} = a\bar{\mathbf{x}}_{\mathcal{A}} + b\bar{\mathbf{y}}_{\mathcal{A}} + c\bar{\mathbf{z}}_{\mathcal{A}}, \tag{4.1}$$

where the magnitude of $\mathbf{r}$ is

$$\|\mathbf{r}_{\mathcal{A}}\| = \sqrt{a^2 + b^2 + c^2}. \tag{4.2}$$

The other aspect of vector notation that is required is that of the unit vector. The calculation of the unit vector $\bar{\mathbf{r}}_{\mathcal{A}}$ is given as

$$\bar{\mathbf{r}}_{\mathcal{A}} = \frac{\mathbf{r}_{\mathcal{A}}}{\|\mathbf{r}_{\mathcal{A}}\|}. \tag{4.3}$$

The coordinate frames is a fundamental part of the ADCS. In order to determine the orientation and position of the satellite relative to the Earth, the Earth Inertial Coordinate (EIC) frame $\mathcal{E} = \{\bar{\mathbf{x}}_{\mathcal{E}},\ \bar{\mathbf{y}}_{\mathcal{E}},\ \bar{\mathbf{z}}_{\mathcal{E}}\}$ is used. The EIC is defined as the Z-axis pointing towards the north pole, the X-axis pointing towards the Vernal Equinox Υ and the Y-axis completing the orthogonal set.

To determine the satellite orientation, satellite-centred coordinate frames are used. Three satellite-centred coordinate frames are used, namely the inertial-reference coordinate (IRC) frame, $\mathcal{I} = \{\bar{\mathbf{x}}_{\mathcal{I}},\ \bar{\mathbf{y}}_{\mathcal{I}},\ \bar{\mathbf{z}}_{\mathcal{I}}\}$, the orbit-referenced coordinate (ORC) frame, $\mathcal{O} = \{\bar{\mathbf{x}}_{\mathcal{O}},\ \bar{\mathbf{y}}_{\mathcal{O}},\ \bar{\mathbf{z}}_{\mathcal{O}}\}$ and the satellite body coordinate (SBC) frame, $\mathcal{B} = \{\bar{\mathbf{x}}_{\mathcal{B}},\ \bar{\mathbf{y}}_{\mathcal{B}},\ \bar{\mathbf{z}}_{\mathcal{B}}\}$.

The IRC coordinate frame's Z-axis points towards the centre of the Earth at perigee (the position of the satellite orbit when the distance between the satellite and the centre of the Earth is the smallest), with the Y-axis being in the opposite direction of the orbit normal-vector and the X-axis completing the orthogonal set. The ORC coordinate frame's Z-axis points towards the centre of the Earth during orbit, with the Y-axis being the orbit anti-normal and the X-axis completing the orthogonal set. The SBC on the other hand rotates with the satellite body and the Z-axis, Y-axis and X-axis are defined based on the positions on the satellite body and not other astronomical objects.

To transform a vector from the EIC frame to the ORC frame the satellite's position vector $\mathbf{r}_{\text{sat}}$ and the satellite's velocity vector $\mathbf{v}_{\text{sat}}$ in EIC is required [31]. Both the satellite position vector $\mathbf{r}_{\text{sat}}$ and the satellite's velocity vector $\mathbf{v}_{\text{sat}}$ is in the EIC frame throughout this thesis unless specified otherwise. The EIC to ORC transformation matrix $\boldsymbol{A}_{\mathcal{E}}^{\mathcal{O}}$ is calculate as

$$\begin{aligned} \boldsymbol{A}_{\mathcal{E}}^{\mathcal{O}} &= \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} \end{bmatrix}^{\mathrm{T}} \\ \text{where} \quad \mathbf{w} &= -\bar{\mathbf{r}}_{\text{sat}} \\ \mathbf{v} &= -\frac{\bar{\mathbf{r}}_{\text{sat}} \times \bar{\mathbf{v}}_{\text{sat}}}{\|\bar{\mathbf{r}}_{\text{sat}} \times \bar{\mathbf{v}}_{\text{sat}}\|} \\ \mathbf{u} &= \mathbf{v} \times \mathbf{w}. \end{aligned} \tag{4.4}$$

4.1.2. Orbit Propagation

The satellite position $\mathbf{r}_{\text{sat}}$ and velocity $\mathbf{v}_{\text{sat}}$ at a given time step is required to determine the multiple different variables required for the simulation environment. The refined version and fourth generation of the simplified general perturbations (SGP) model, namely SGP4, is therefore used as orbit propagator of the satellite after each time step [32].

To determine $\mathbf{r}_{\text{sat}_k}$ and $\mathbf{v}_{\text{sat}_k}$ at a specific time step k the two-line element (TLE) set of

the satellite is required. The TLE set is an encoding of the specified satellite orbit, which require orbit parameters. These parameters and the chosen value for the orbit parameter include the:

- Semimajor axis $a = 6879.55$km which is half the distance of the longest diameter of the ellipse.
- Right ascension of the ascending node (RAAN) defined as the angle $\Omega = 275°$ from the origin of longitude to the direction of the ascending node.
- Argument of perigee (AP) defined as the angle $\omega = 57.4°$ from the ascending node to perigee.
- Inclination $i = 97.4°$ as the tilt of the satellite orbit with respect to the EIC frame.
- Eccentricity e calculated as

$$e = \sqrt{1 - \frac{b^2}{a^2}} = 0.000092 \tag{4.5}$$

  where b is the semi-minor axis defined as half the distance of the shortest diameter of the ellipse.

The time at the beginning of the orbit as a Julian date J_t is also required for the TLE. A Julian date is a date represented as a single number. The numbers begin on the first of January, 4713BC and ends on the 22$^{\text{nd}}$ of January, 3268AD. With these parameters and the elapsed time since J_t, both $\mathbf{r}_{\text{sat}_k}$ and $\mathbf{v}_{\text{sat}_k}$ can be determined from the World Geodetic System 72 constants that is implemented through the SGP4 model. An example of the satellite orbit parameters are illustrated in Figure 4.1.

The SGP4 is implemented with the SGP4 Python package [33]. The SGP4 outputs the $\mathbf{r}_{\text{sat}_k}$ and $\mathbf{v}_{\text{sat}_k}$ in the EIC reference frame. Therefore, with the satellite position $\mathbf{r}_{\text{sat}}$ and velocity $\mathbf{v}_{\text{sat}}$ known, the EIC to ORC transformation matrix $\boldsymbol{A}_{\mathcal{E}}^{\mathcal{O}}$ can be calculated according to Equation 4.4.

4.1.3. Attitude of a Satellite

After discussing the background for the position of the satellite relative to the EIC frame the background of the attitude of the satellite is required. The attitude of the satellite is determined through the relative orientation of the SBC frame to the ORC frame. To determine the attitude of the satellite the fundamentals of 3D object rotation is required. This is firstly explained through the Euler angles method which is easily understandable and can be demonstrated visually with Figure 4.2.

Euler angles describe the rotation of an object around three orthogonal axes. The three axes of the SBC frame, denoted by $\bar{\mathbf{x}}_{\mathcal{B}}$, $\bar{\mathbf{y}}_{\mathcal{B}}$ and $\bar{\mathbf{z}}_{\mathcal{B}}$, rotate with the object as depicted

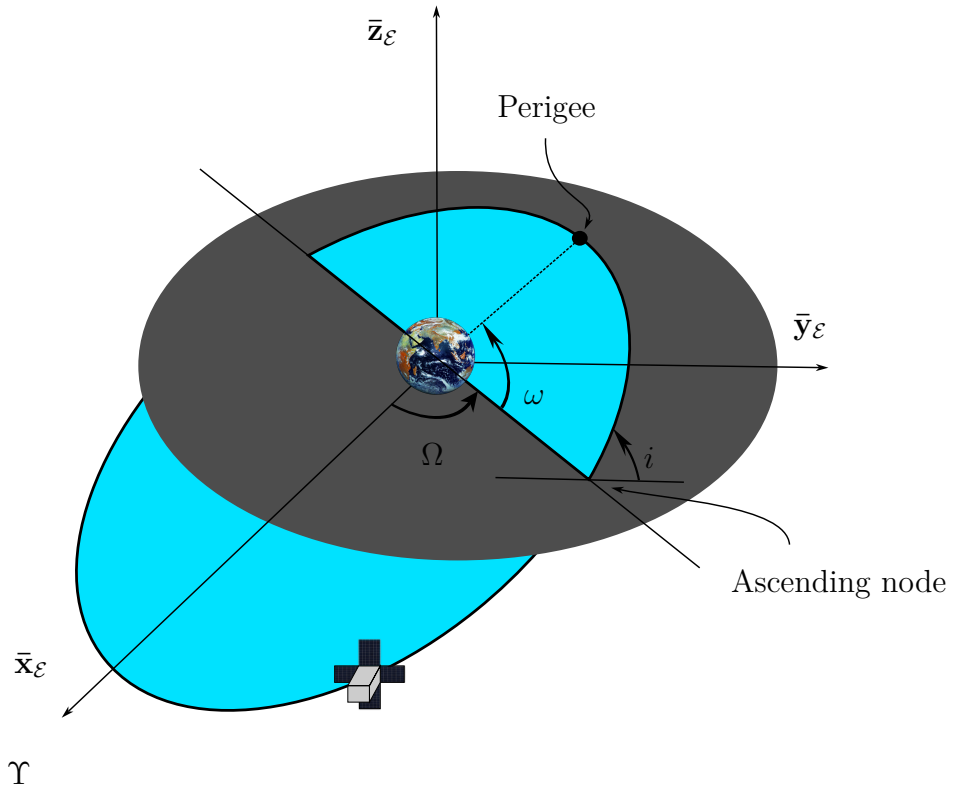

Figure 4.1: Illustration of orbit parameters and the Earth Inertial Coordinate frame. This illustration is not according to scale.

in Figure 4.2. After the object is rotated by the angle ψ around $\bar{\mathbf{y}}_{\mathcal{B}}$, the axis $\bar{\mathbf{x}}_{\mathcal{B}}$ and $\bar{\mathbf{z}}_{\mathcal{B}}$ translate to $\bar{\mathbf{x}}'_{\mathcal{B}}$ and $\bar{\mathbf{z}}'_{\mathcal{B}}$ respectively. Thereafter the object is rotated by the angle ϕ around $\bar{\mathbf{x}}'_{\mathcal{B}}$ and the axis $\bar{\mathbf{y}}_{\mathcal{B}}$ and $\bar{\mathbf{z}}'_{\mathcal{B}}$ translate to $\bar{\mathbf{y}}'_{\mathcal{B}}$ and $\bar{\mathbf{z}}''_{\mathcal{B}}$ respectively. The object is then rotated by the angle θ around $\bar{\mathbf{z}}''_{\mathcal{B}}$ where the axis $\bar{\mathbf{x}}'_{\mathcal{B}}$ and $\bar{\mathbf{y}}'_{\mathcal{B}}$ translate to $\bar{\mathbf{x}}''_{\mathcal{B}}$ and $\bar{\mathbf{y}}''_{\mathcal{B}}$ respectively.

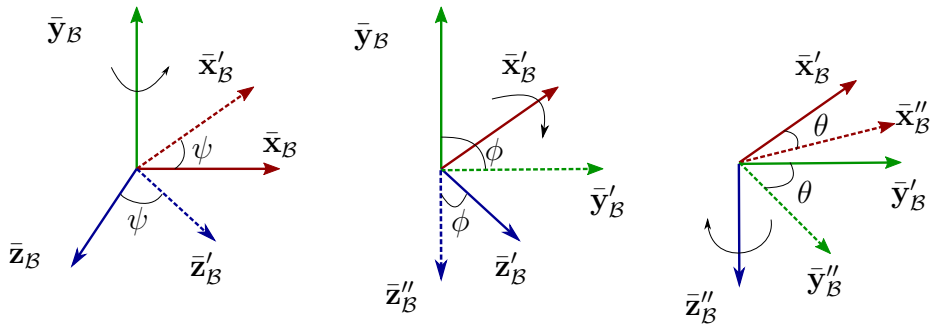

Figure 4.2: Euler angle transformation of an object. The object rotates around $\bar{\mathbf{y}}_{\mathcal{B}}$ by the angle ψ, then it rotates around $\bar{\mathbf{x}}'_{\mathcal{B}}$ by the angle ϕ and lastly rotates around $\bar{\mathbf{z}}''_{\mathcal{B}}$ by the angle θ. The dotted lines represent the transformed axes after the rotation of an angle around the other axis.

The Euler angles provide a method of rotating an object in 3D. This can be utilised to transform a vector from one coordinate frame to another. The Euler angles can also be given as a single matrix (transformation matrix) to transform the vector from a given coordinate frame to a desired coordinate frame. The transformation of the vector is

required in the EKF algorithm to compare the modelled vector $\mathbf{v}_{\mathcal{O}}$ in the ORC frame with the measured vector $\mathbf{v}_{\mathcal{B}}$ in the SBC frame.

To transform a vector from the ORC to SBC frame, the direct cosine matrix (DCM), also referred to as $\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}$, is used. The DCM $\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}$ can be used to calculate the attitude transformation from given Euler angle rotations. This is done by multiplying the transformation matrices representing each individual Euler angle rotation. $\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}$ can therefore be calculated as

$$\begin{aligned}\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}} &= \boldsymbol{A}_{\psi}\boldsymbol{A}_{\phi}\boldsymbol{A}_{\theta} \\ &= \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}.\end{aligned} \tag{4.6}$$

Euler angles, however, do not always prove a suitable method for determining the attitude of a satellite. This is due to singularities that can occur, such as the gimbal-lock effect, where two rotational axes coincide to form a single rotational axis. Consequently, not all 3D rotations can be described with Euler angles, because with gimbal-lock only two effective rotations can occur instead of three [34]. The method of describing 3D rotation with quaternions is therefore more convenient and more often used [35]. A quaternion $\mathbf{q}$ has four components that are dependent of one another and constrained by

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \tag{4.7}$$

The attitude quaternion is also related to the Euler angles in that if the Euler rotational axis from ORC to SBC is defined as a unit vector $\mathbf{e} = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^{\mathrm{T}}$ and the angle of the Euler rotation is Φ, then

$$\mathbf{q} = \begin{bmatrix} e_1 \sin(\frac{\Phi}{2}) \\ e_2 \sin(\frac{\Phi}{2}) \\ e_3 \sin(\frac{\Phi}{2}) \\ \cos(\frac{\Phi}{2}) \end{bmatrix}. \tag{4.8}$$

Although it is difficult to visualize a quaternion, the most simplistic method of conceptualising it is shown in Figure 4.3. A quaternion can be simplified to a unit vector protruding from the centre point of an object as well as the angle of rotation of that object around the unit vector. As seen in Figure 4.2 the angle θ is the angle of rotation around the $\bar{\mathbf{z}}''_{\mathcal{B}}$-axis. For quaternions the angle of rotation is a similar concept, however, the axis around which the object is rotating, is the unit vector, q_{1-3}. The quaternion element q_4, therefore provides the angle of rotation while q_{1-3} represents the unit vector with the condition of Equation 4.7.

To adjust the DCM to rely on quaternions, $\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}$ can be transformed as a function of $\mathbf{q}$

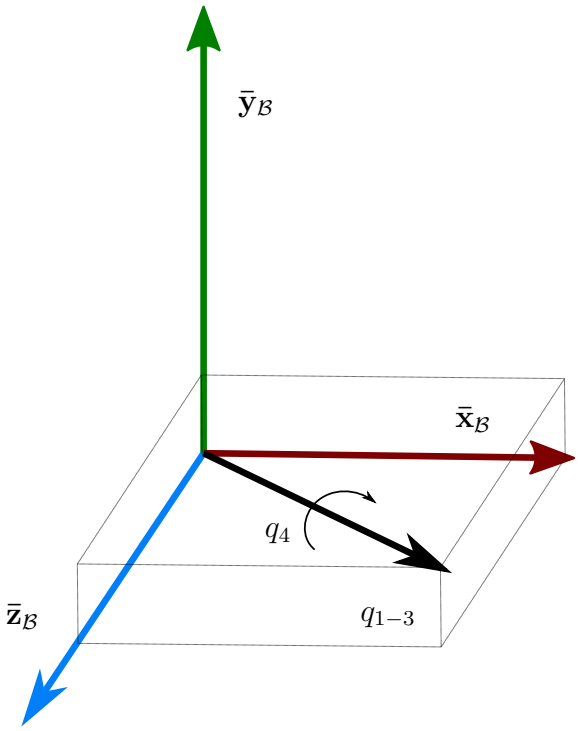

Figure 4.3: Graphical representation of a quaternion to explain the relationship between a quaternion and the Euler angles. Where q_{1-3} represents the unit vector and q_4 represents the angle or rotation.

through

$$\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{4.9}$$

as provided by James Wertz [36]. With this the DCM can be calculated with quaternions. This section is therefore concluded since the attitude can be determined by transforming the vectors in the ORC frame to the SBC frame with the implementation of the quaternions and the transformation matrices. This is a basic requirement for simulating the normal operations of the satellite's ADCS as well as the practical anomalies.

4.1.4. Satellite Kinematics and Dynamics

The dynamics of a satellite can be calculated with the Euler dynamic equation. This consists of the torques applied to the satellite and are mainly control torques $\boldsymbol{\tau_c}$ and disturbance torques $\boldsymbol{\tau_d}$ as well as the moment of inertia of the satellite $\mathbf{J}$ multiplied by the inertial-referenced angular acceleration of the satellite $\dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}$. The control torques used in this design are only reaction wheel torques $\boldsymbol{\tau}_w$ and magnetorquer torques $\boldsymbol{\tau}_m$. The disturbance torques accounted for in this thesis are the gravity gradient torque $\boldsymbol{\tau}_{gg}$, the wheel imbalance torque $\boldsymbol{\tau}_{rw}$, the gyroscopic coupling torque $\boldsymbol{\tau}_{\text{gyro}}$, and the aerodynamic

disturbance torque $\boldsymbol{\tau}_{\text{aero}}$. The Euler dynamic equation can therefore be given as

$$\begin{aligned} \mathbf{J}\dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}} &= \boldsymbol{\tau_c} + \boldsymbol{\tau_d}, \\ \text{where} \quad \boldsymbol{\tau_d} &\approx \boldsymbol{\tau}_{\text{aero}} - \boldsymbol{\tau}_{\text{gyro}} + \boldsymbol{\tau}_{gg} + \boldsymbol{\tau}_{rw}, \\ \text{and} \quad \boldsymbol{\tau_c} &= \boldsymbol{\tau}_m - \boldsymbol{\tau}_w. \end{aligned} \tag{4.10}$$

The kinematics of the satellite is represented by the rate of change of the quaternions, $\dot{\mathbf{q}}$. To determine $\dot{\mathbf{q}}$ the angular rate vector $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}}$ is required. Where the angular rate $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}}$ is the angular rate of the SBC frame $\mathcal{B}$ relative to the ORC frame $\mathcal{O}$. The angular rate vector $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}}$ is related to $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}$ through

$$\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}} = \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}} - \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}} \begin{bmatrix} 0 \\ -\omega_o \\ 0 \end{bmatrix}, \tag{4.11}$$

where ω_o is the angular rate of the satellite about the centre of the Earth. The angular rate vector $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}}$ can be divided into components ω_x, ω_y and ω_z, from which $\dot{\mathbf{q}}$ can be calculated as [22]

$$\dot{\mathbf{q}} = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \mathbf{q}. \tag{4.12}$$

Equation 4.10-4.12 are the overarching equations that will be used to determine the control torque as well as the model update of the EKF. This section is implemented in the simulation environment to determine the actual attitude of the satellite represented as a quaternion $\mathbf{q}$ and the angular rate of the SBC frame relative to the ORC frame $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}}$. This section provides the underlying kinematics and dynamics of the satellite which provides the unique environment of a satellite in which the practical anomalies are modelled and the FDIR methods can be developed.

4.2. Environmental Factors and Disturbances

To simulate the environment, certain aerospace phenomena must be modelled to create a realistic representation of the satellite orbit and therefore ensure that the sensor measurements and all anomalies can be accurately modelled. These anomalies also ensure that the developed EKF is robust and accurate in normal operations when general environmental disturbances occur. The position of the satellite with respect to the Earth (orbit propagation) is required for the simulation environment and is already discussed in the previous section. The Sun's position is also required since the combination of the Sun's position and the Earth's position determines the eclipse and sunlit phases of the satellite orbit.

The position of the Earth also determines the geomagnetic field that is simulated for the measurements of the magnetometer. These environmental factors provide the elements for the simulation environment as well as the modelling of the anomalies and therefore the Sun's position as well as the geomagnetic field is discussed in this section.

4.2.1. The Sun's Position and Eclipse

For a mission to be successful it is critical to determine the position of the Sun relative to the satellite. This is because the satellite must determine whether it is in an eclipse to determine the required mode of operation. The model from D. Vallado [37] is therefore implemented to determine the position of the Sun in the EIC frame.

From this model, the vector from the centre of the Earth to the centre of the Sun $\mathbf{r}_{sun}$ is provided in the EIC frame. For this calculation, the difference between the current Julian date J_t and the J_{2000} epoch is required. Where $J_{2000} = 2\,451\,545$ and the difference is thereafter converted to the amount of Julian centuries (365.25 days). The time difference in Julian centuries T_{JC} can therefore be calculated as

$$T_{JC} = \frac{J_t - 2\,452\,545}{365.25}. \tag{4.13}$$

In addition to the Julian Centuries a few other parameters are required to calculate the Sun's position. The definitions and descriptions of these parameters are tabulated in Table 4.1.

Symbol	Definition	Description
$r_\oplus$	Sun's position magnitude	The absolute distance of the Earth to the Sun
λ_e	Ecliptic longitude	The angle between the primary direction 0° of the plane in which the Earth is orbiting and the current position of the Earth.
$M_\oplus$	Mean anomaly	The fraction of the Earth's orbit after it has passed the furthest position from the Sun
ϵ	Obliquity	The inclination of the plane of orbit to the celestial equator
$\lambda_{M_\oplus}$	Sun's mean longitude	The average angle subtended at the Earth between the vernal equinox and the sun [38].

Table 4.1: Description and definition of the Earth orbit parameters.

With T_{JC} known and with the background of the parameters in Table 4.1, the Sun's position $\mathbf{r}_{sun}$ can be calculated as

$$
\begin{aligned}
\mathbf{r}_{sun} &= r_{\oplus} \begin{bmatrix} \cos(\lambda_e) \\ \cos(\epsilon)\sin(\lambda_e) \\ \sin(\epsilon)\sin(\lambda_e) \end{bmatrix}, \\
\text{where} \quad r_{\oplus} &= 1.000\,140\,612 - 0.016\,708\,617\cos(M_{\oplus}) - 0.001\,395\,89\cos(2M_{\oplus}), \\
M_{\oplus} &= 357.527\,723\,300^{\circ} + 35\,999.050\,340\,T_{JC}, \\
\lambda_e &= \lambda_{M_{\oplus}} + 1.914\,666\,471\sin(M_{\oplus}) + 0.019\,994\,643\sin(2M_{\oplus}), \\
\lambda_{M_{\oplus}} &= 280.460\,618\,400^{\circ} + 36\,000.770\,053\,610\,T_{JC}, \\
\text{and} \quad \epsilon &= 23.439\,291^{\circ} - 0.013\,004\,200\,T_{JC}.
\end{aligned}
\tag{4.14}
$$

After determining the Sun's position, it is crucial to calculate whether the satellite is in an eclipse or not. This can be done with basic geometry after calculating the position of the Sun relative to the satellite through

$$\mathbf{s}_{\mathcal{E}} = \mathbf{r}_{sun} - \mathbf{r}_{\text{sat}}. \tag{4.15}$$

The assumption is made that whenever the satellite is not able to view the centre of the Sun it is in an eclipse. This a valid assumption given the small angle required to change the satellite from a partial eclipse (only a part of the satellite is in the shadow of the Earth) to a full eclipse (the entire satellite body is in the shadow of the Earth). This is due to the comparative distances of the Sun to the satellite and the satellite to the Earth as well as the comparative sizes between the satellite and the other celestial bodies. The eclipse is therefore defined as the period during which θ_s is smaller than θ_E. Where

$$\theta_E = \sin\left(\frac{R_E}{\|\mathbf{r}_{\text{sat}}\|}\right) \tag{4.16}$$

and

$$\theta_s = \cos\left(\frac{(\mathbf{r}_{\text{sat}} \cdot \mathbf{s}_{\mathcal{E}})}{(\|\mathbf{r}_{\text{sat}}\|\,\|\mathbf{s}_{\mathcal{E}}\|)}\right) \tag{4.17}$$

as shown in Figure 4.4, where R_E is the radius of the Earth. The satellite position $\mathbf{r}_{\text{sat}}$ is calculated according to the SPG4 model from the orbit propagation. This therefore concludes the calculation for the Sun's position in the EIC frame and the eclipse calculation.

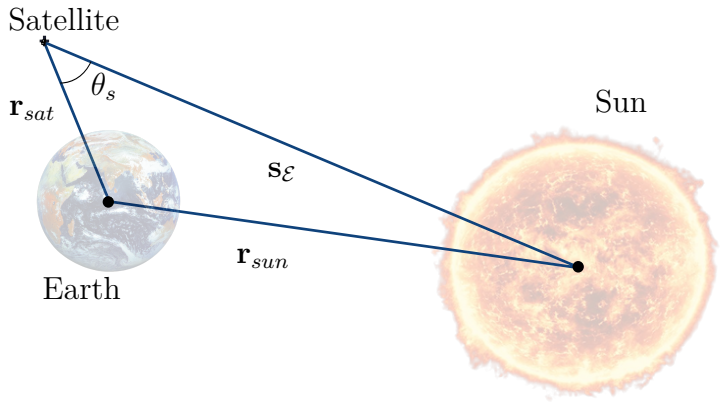

Figure 4.4: Geometry to determine satellite eclipse phases. To calculate whether the satellite is in the eclipse the Equations 4.16 and 4.17 are implemented.

4.2.2. Geomagnetic Field

The geomagnetic field is another environmental factor that should be part of the simulation environment. This is to ensure that the simulation of the magnetorquer — magnetic alloy rod that provides additional control of the satellite by reacting with the Earth's geomagnetic field — and the magnetometer is realistic.

The Earth generates a magnetic field through electric currents due to motion within the molten core of the Earth, which is commonly referred to as the geomagnetic field [39]. The magnetorquers interact with the geomagnetic field mostly for momentum dumping and the magnetometers measure the geomagnetic field for attitude estimation.

The geomagnetic field is modelled with the time-varying International Geomagnetic Reference Field (IGRF) model released by the International Association of Geomagnetism and Aeronomy (IAGA). This model is used for increased ADCS accuracy and the 13$^{\text{th}}$ generation of the model is implemented [40]. To calculate the geomagnetic field, the scalar potential function as well as the function's parameters is required. The parameters required for the scalar potential function is the mean of the Earth radius ($R_E = 6371.2$km), the radial distance from the centre of the Earth r_s, the latitude θ and the longitude ϕ. Other parameters required is the Gauss coefficients $g_n^m(t)$ and $h_n^m(t)$ which change slowly with time and, consequently, the IGRF-13 provide values for these coefficients at 5-year epoch intervals. The Legendre functions $P_n^m(\cos(\theta))$ of the degree n and m are also part of the geomagnetic field model [41]. The scalar potential function

$$V(r_s, \theta, \phi, t) = R_E \sum_{n=1}^{N} \left(\frac{R_E}{r_s}\right)^{n+1} \sum_{m=0}^{n} \left(g_n^m(t)\cos(m\phi) + h_n^m(t)\sin(m\phi)\right) P_n^m(\cos(\theta)), \quad (4.18)$$

is used to calculate the geomagnetic field. Where the geomagnetic field $\mathbf{b}_{\mathcal{E}}$ in EIC is the

gradient of the scalar potential function

$$\mathbf{b}_{\mathcal{E}} = -\nabla V. \tag{4.19}$$

To provide insight on the change of the geomagnetic field during the satellite orbit, the magnitude of the geomagnetic field at varying longitudes and latitudes is visually demonstrated in Figure 4.5. Figure 4.5 provides a static map of the magnitude of the geomagnetic field provided by the IGRF model [42].

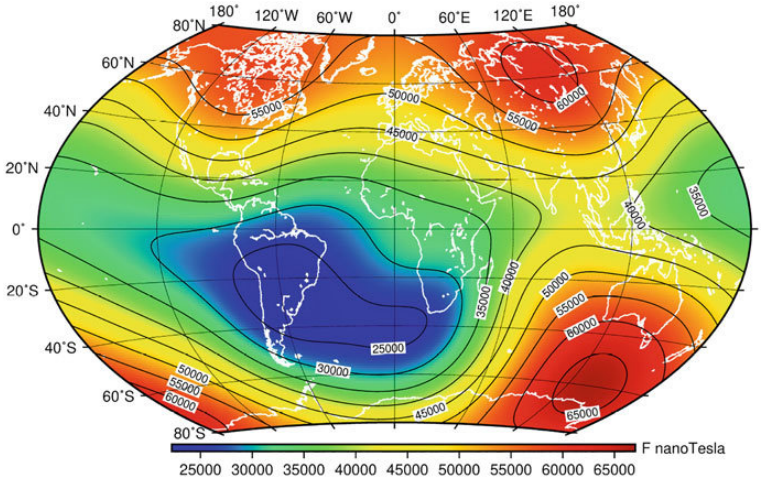

Figure 4.5: A map of the Earth with varying longitude and latitude to provide the magnitude of the geomagnetic field according to the 13th generation of the IGRF model [42].

4.2.3. Disturbance Models

A satellite is exposed to various disturbance torques during an orbit. It is these torques that cause the modelled attitude to differ from the actual attitude. These torques are therefore modelled and are assumed to influence the attitude of the satellite continuously. Minor disturbances are excluded from the simulation environment and only the major sources of disturbance torques are modelled and discussed.

Gyroscopic Coupling

The first disturbance torque is that of the gyroscopic coupling torque. A gyroscopic coupling torque results due to the rotation of the reaction wheel's axis and is perpendicular to the rotor axis [43]. The gyroscopic coupling torque acts perpendicular to the rotor axis which can be calculated as

$$\boldsymbol{\tau}_{\text{gyro}} = \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}} \times (\mathbf{J}\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}} + \mathbf{h}_w), \tag{4.20}$$

where $\mathbf{h}_w$ is the angular momentum of the reaction wheels, $\mathbf{J}$ is the moment of inertia of the satellite and $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}$ is the angular rate of the SBC frame relative to the IRC frame.

Gravity Gradient

The second major disturbance torque is the gravity gradient torque caused by both the centrifugal force on the satellite due to the orbit around the Earth as well as the gravitational force. The part of the satellite nearest to the Earth will experience the largest gravitational force and the smallest centrifugal force, while the part of the satellite furthest from the Earth will experience the smallest gravitational force and the largest centrifugal force. According to J. Wertz [36] the gravity gradient disturbance torque $\boldsymbol{\tau}_{gg}$ can be calculated as

$$\boldsymbol{\tau}_{gg} = 3\,\omega_o^2(\mathbf{z}_{\mathcal{B}} \times \mathbf{J}\mathbf{z}_{\mathcal{B}}). \tag{4.21}$$

Where ω_o is the orbit rate and the orbit nadir vector $\mathbf{z}_{\mathcal{B}}$ is calculated as

$$\mathbf{z}_{\mathcal{B}} = \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}. \tag{4.22}$$

The gravity gradient torque is the only torque that can be accurately modelled on-board the satellite, and is therefore also included in the model update of the EKF. To provide insight into the value of the gravity gradient disturbance torque the vector is provided in the SBC frame in Figure 4.6. Although the gravity gradient torque is in the order of a few dozen $\mathrm{nN} \cdot \mathrm{m}$, this torque is required to simulate a realistic simulation environment as it is comparable to the magnitude of the other major disturbance torques.

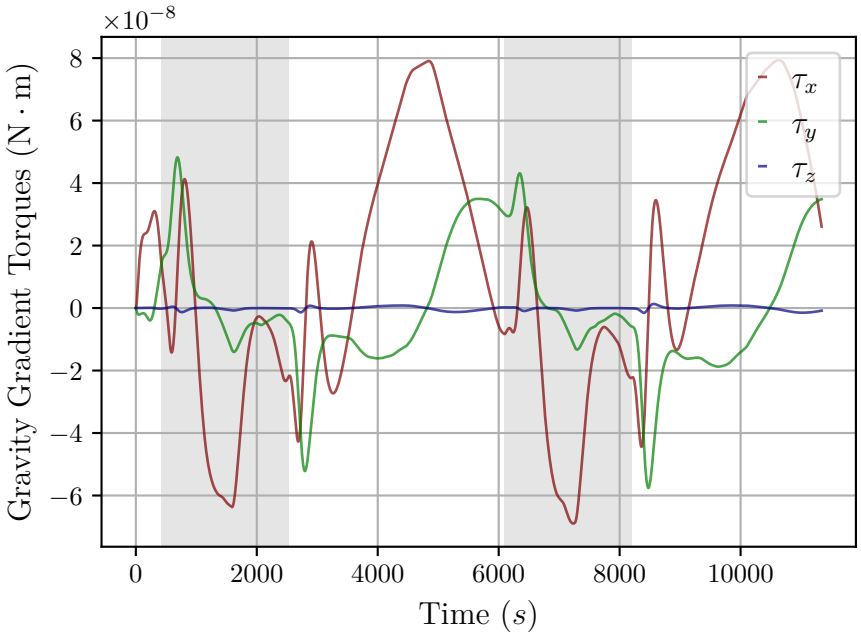

Figure 4.6: The simulated gravity gradient torque in the SBC frame. The grey background sections of the graphs are the eclipse periods, while the sections with the white background is the sunlit phase of the orbit. The legend provides the axis of the coordinate frame for which the magnitude of the torque is given.

Aerodynamic Disturbance

The third simulated disturbance torque is that of the aerodynamic torque caused by air in the atmosphere creating a force on each individual segment of the satellite [44]. The aerodynamic disturbance is significant due to the low Earth orbit (LEO) of the satellite, where the atmospheric density is higher. The aerodynamic disturbance torque $\boldsymbol{\tau}_{\text{aero}}$ is therefore a summation of all the torques created by the air force on each segment's surface area A_i. The aerodynamic disturbance torque $\boldsymbol{\tau}_{\text{aero}}$ according to Steyn et al. [44] can be calculated as

$$\begin{aligned}\boldsymbol{\tau}_{\text{aero}} = \sum_{i=1}^{n} \Big(\rho \left\| \mathbf{v}_{\mathcal{B}\text{a}} \right\|^2 A_i H\{\cos(\alpha_i)\} \cos(\alpha_i) \Big(\sigma_t (\mathbf{r}_{pi} \times \bar{\mathbf{v}}_{\mathcal{B}}) \\ + \Big[\sigma_n S + (2 - \sigma_n - \sigma_t) \cos(\alpha_i) \Big] (\mathbf{r}_{pi} \times \bar{\mathbf{n}}_i) \Big) \Big),\end{aligned} \tag{4.23}$$

where n is the number of segments of the satellite. The factors that influence the aerodynamic disturbance torque is the atmospheric velocity $\mathbf{v}_{\mathcal{B}\text{a}}$ in SBC, the atmospheric density ρ and the offset vector $\mathbf{r}_p$ between the segment's centre of mass (CoM) and the centre of pressure (CoP). $H\{x\}$ is the Heaviside function which is equal to 0 when x is smaller than 0 and otherwise equal to 1. The other parameters required to calculated the aerodynamic disturbance is the incidence angle α_i of $\mathbf{v}_{\mathcal{B}\text{a}}$ on segment i, the tangential accommodation coefficient σ_t and the normal accommodation coefficient σ_n. S is furthermore the ratio of molecular exit velocity to $\mathbf{v}_{\mathcal{B}\text{a}}$ and $\bar{\mathbf{n}}_i$ is the unit inward normal vector of segment i [22]. The parameters σ_n, σ_t and S are assumed to be equal to 0.8 [45].

The atmospheric density is a function of the distance from the Earth surface. The density model provided by D. Vallado [37] is given as

$$\rho = \rho_o e^{-\frac{h(t) - h_o}{H}}, \tag{4.24}$$

where ρ_o is the reference density at the reference altitude h_o and $h(t)$ is the satellite's altitude as a function of time and H is the scale height. According to Steyn et al. [45] the atmospheric density can be approximated as $\frac{1}{2}\rho$ during an eclipse. The atmospheric velocity $\mathbf{v}_{\mathcal{B}\text{a}}$ in SBC is furthermore calculated as

$$\begin{aligned}\mathbf{v}_{\mathcal{B}\text{a}} &= \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}} \boldsymbol{A}_{\mathcal{E}}^{\mathcal{O}} \mathbf{v}_{\mathcal{E}} \\ \text{where} \qquad \mathbf{v}_{\mathcal{E}} &= \begin{bmatrix} 0 \\ 0 \\ \omega_E \end{bmatrix} \times \mathbf{r}_{\text{sat}} - \mathbf{v}_{\text{sat}}.\end{aligned} \tag{4.25}$$

The angular rate ω_E of the Earth is equal to 1.99×10^{-7}rad/s. To provide insight into the value of the aerodynamic disturbance torque axis components of the vector is provided in the SBC frame in Figure 4.7 as calculated with Equation 4.23. The aerodynamic

disturbance torque is in the order of a few hundred nN · m and has a larger effect on the simulation of the satellite than the gravity gradient torque.

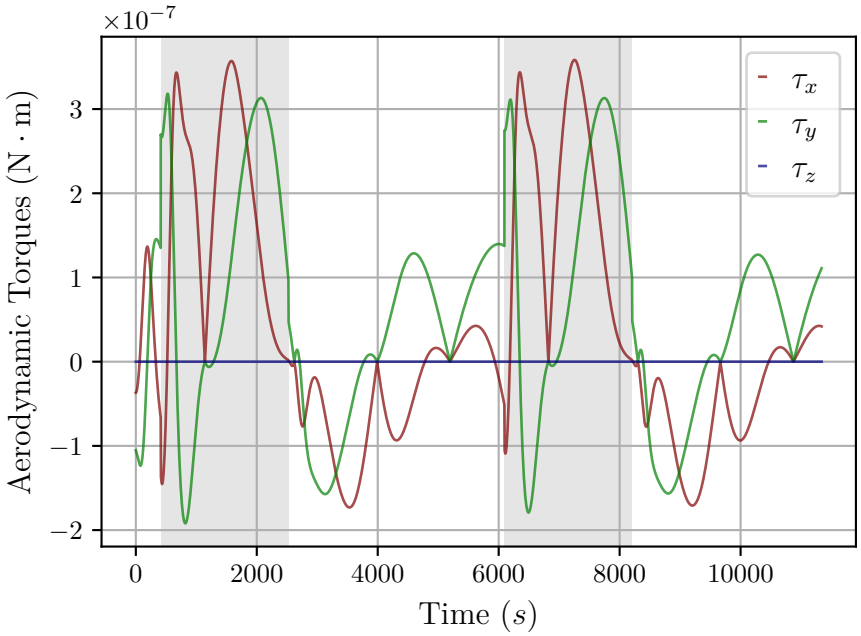

Figure 4.7: The simulated aerodynamic torque in the SBC frame. The grey background sections of the graph are the eclipse periods. The legend provides the magnitude component of each axiswithin the SBC coordinate frame.

Wheel Imbalance

The fourth and final major simulated disturbance torque is that of the reaction wheel imbalance. The reaction wheel imbalance torque is considered to be the most significant disturbance attributed to the reaction wheel [46]. It is therefore the only reaction wheel disturbance torque modelled for this simulation. Although reaction wheels are manufactured with low tolerances, the reaction wheel will have a slight imbalance, since the mass of the reaction wheel will not be perfectly uniform and evenly distributed.

The static imbalance of the reaction wheel is caused by the reaction wheel CoM offset from the rotational axis. To model the static imbalance of the reaction wheels it is therefore assumed that the unevenly distributed mass of the reaction wheel can be simplified to a point mass m a distance r from the rotational axis as shown in Figure 4.8. The static imbalance U_s is equal to mr and this value is usually provided by the reaction wheel manufacturers.

To determine the resulting torque from the wheel imbalance, the torque generated by each wheel is individually calculated. The force $\mathbf{f}_{sx}$ generated by U_s on the reaction wheel in the $\bar{\mathbf{x}}_{\mathcal{B}}$-direction denoted as RW_x, is dependent on the the angular rate ω of the reaction wheel as well as the current position of the point mass m. The force $\mathbf{f}_{sx}$ can be

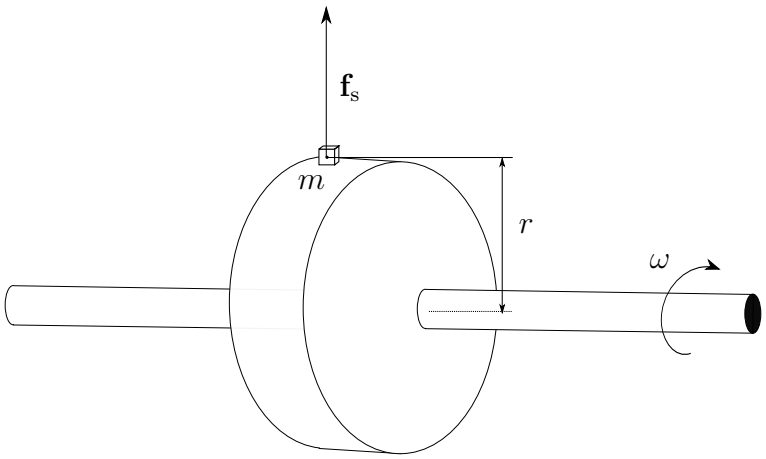

Figure 4.8: Visual representation of the modelling of the static reaction wheel imbalance.

calculated as

$$\mathbf{f}_{\mathrm{s}x} = U_\mathrm{s}\omega^2 \begin{bmatrix} 0 \\ \sin(\omega t + \phi_\mathrm{s}) \\ \cos(\omega t + \phi_\mathrm{s}) \end{bmatrix}, \tag{4.26}$$

where ϕ_s is an arbitrary phase. With the force $\mathbf{f}_{\mathrm{s}x}$ exerted on the reaction wheel RW_x known, the disturbance torque generated by the reaction wheel on the satellite can be calculated with the known position vector $\mathbf{w}$ of the force $\mathbf{f}_{\mathrm{s}x}$ to the satellite CoM. The disturbance torque due to the reaction wheel RW_x can be calculated as

$$\boldsymbol{\tau}_{\mathrm{s}x} = \mathbf{w} \times \mathbf{f}_{\mathrm{s}x}. \tag{4.27}$$

This is calculated for each reaction wheel to determine the resulting static imbalance disturbance torque on the satellite.

Another aspect of the reaction wheel imbalance is also modelled, namely the dynamic imbalance. The dynamic imbalance is caused by the principal inertia of the reaction wheel being misaligned with the rotational axis. This can be simplified to two equal point masses m with an axial displacement d and distance r from the rotational axis. These two masses are separated by an angle of 180° with respect to the rotational axis and consequently create two forces equal in magnitude and in opposite directions. The dynamic imbalance is graphically represented in Figure 4.9.

The dynamic wheel imbalance torque $\boldsymbol{\tau}_{\mathrm{d}x}$ for RW_x can be calculated as

$$\boldsymbol{\tau}_{\mathrm{d}x} = U_\mathrm{d}\omega^2 \begin{bmatrix} 0 \\ \sin(\omega t + \phi_\mathrm{d}) \\ \cos(\omega t + \phi_\mathrm{d}) \end{bmatrix}, \tag{4.28}$$

where $U_\mathrm{d} = mrd$ is the dynamic imbalance and ϕ_d is an arbitrary phase. Both U_d and U_s are provided by the manufacturer and based on the reaction wheel as discussed in

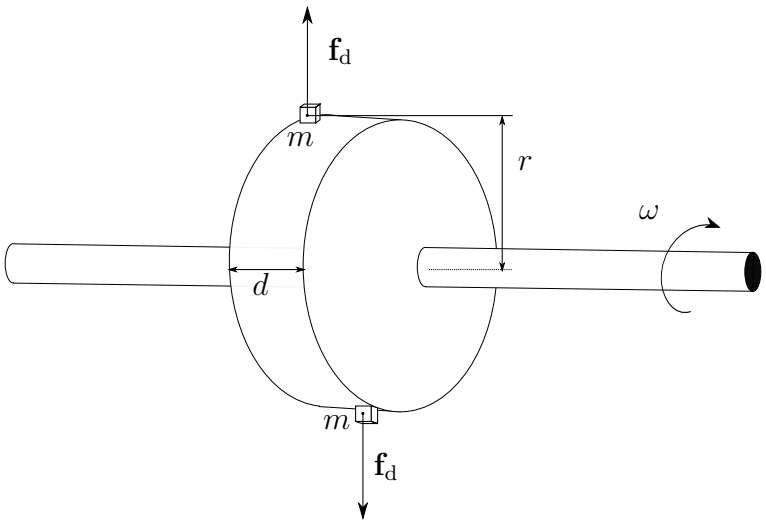

Figure 4.9: Visual representation of the modelling of the dynamic reaction wheel imbalance.

Section 3.4. The wheel imbalance torque from both the static and dynamic wheel imbalance is provided in Figure 4.10. It is clear from Figure 4.10 that the wheel imbalance disturbance torques are in the order of a few dozen $\mu\mathrm{N}\cdot\mathrm{m}$.

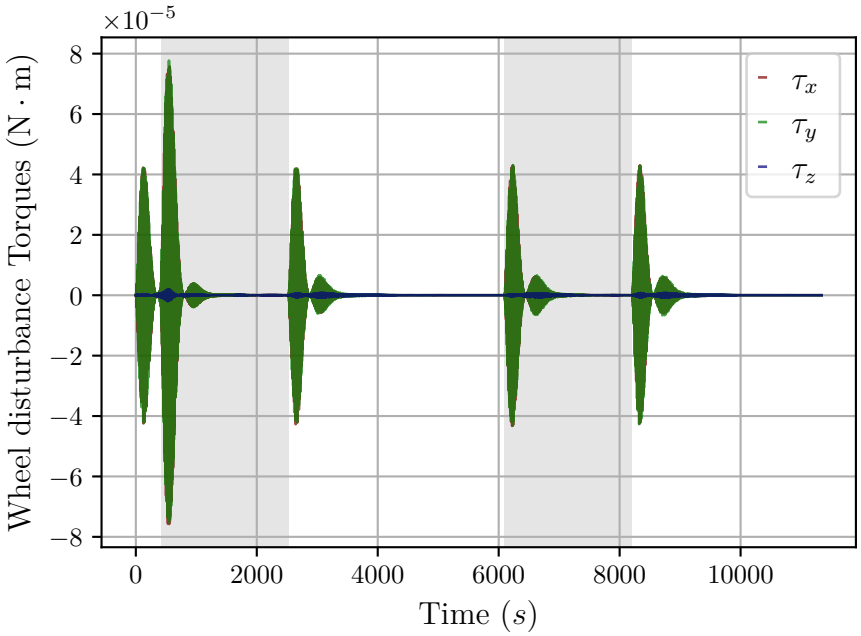

Figure 4.10: The simulated wheel disturbance torques in SBC. The grey background sections of the graphs are the eclipse periods, while the sections with the white background is the sunlit phase of the orbit. The legend provides the axis of the coordinate frame for which the magnitude of the torque is given.

This is the largest simulated disturbance torque on the satellite and momentum dumping is therefore required to reduce the disturbance torque. Figure 4.10, however, already provides the reaction wheel disturbance torque with the implementation of the

momentum dumping, since the satellite is unstable without the momentum dumping due to the magnitude of the reaction wheel disturbance torque. This therefore concludes the simulation block in Figure 1.2

4.3. Attitude Determination and Control System

This section discusses the overview of the ADCS implementation. This is required to develop and simulate the controller, EKF and sensor blocks in Figure 1.2. The sensor models, the extended Kalman filter (EKF) algorithm and the modes of control are therefore discussed in this section.

The ADCS is the system of the satellite for which the FDIR methods will be developed. This is to ensure accurate and robust operation of the EKF. This section therefore provides the development and normal operation of the EKF and control of the satellite.

4.3.1. Sensor Models

The positioning of the sensors on the satellite is necessary to meet mission requirements. The exact position of the sensors also impact the modelling of the anomalies on the sensors. The sensors positions on the satellite are therefore provided in Section 3.3. It is further assumed that each sensor has a zero-mean Gaussian noise and consequently, the low frequency noise such as drift is negligible. The sensor measurement $\mathbf{v}_{\mathcal{B}}$ in the SBC frame can be calculated as

$$\mathbf{v}_{\mathcal{B}} = \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\mathbf{v}_{\mathcal{O}} + \mathbf{m}_v, \tag{4.29}$$

where $\mathbf{m}_v$ is the measurement noise of the current sensor, $\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}$ is the DCM and $\mathbf{v}_{\mathcal{O}}$ is the reference ORC vector. The measured unit vector of the sun sensor is provided in Figure 4.11 as an example of the sensor measurements. This in addition to the simulation environment concludes the sensor block in Figure 1.2.

4.3.2. Attitude Determination

This section discusses the sensor fusion algorithm for attitude determination of the satellite. This is done with the EKF, which utilizes the sensor measurements as well as modelled vectors according to mathematical models to estimate the current attitude. The EKF is highly sensitive to sensor anomalies and actuator failures [4]. It is for this reason that FDIR is required to ensure that the estimation remains accurate during sensor anomalies.

Extended Kalman Filter

The EKF is implemented to estimate the current satellite attitude with sensor fusion of the magnetometer, horizon sensor and sun sensor. The EKF will be used due to the

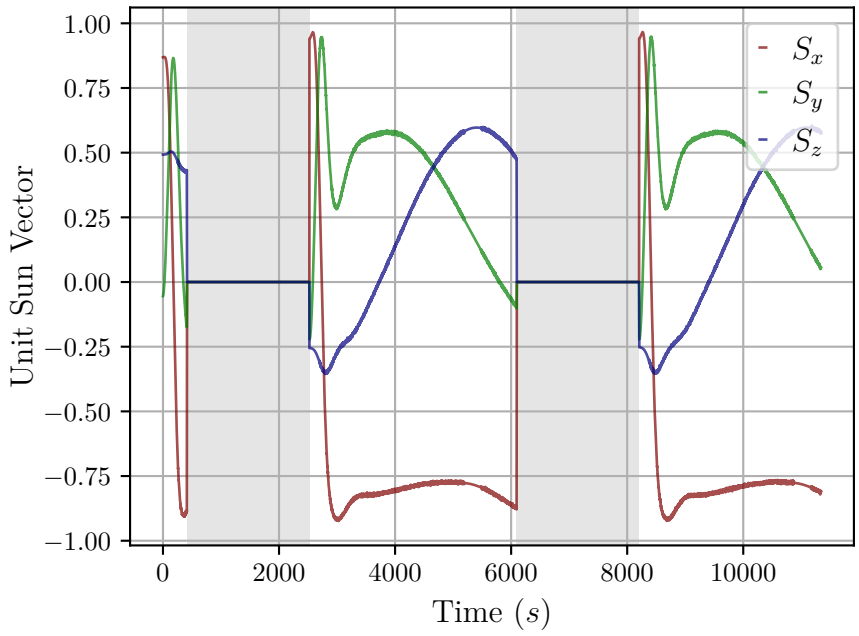

Figure 4.11: The measured unit vector of the Sun in the SBC frame. The grey background sections of the graphs are the eclipse periods, while the sections with the white background is the sunlit phase of the orbit. The legend provides the axis component of the coordinate frame for the unit vector of the sun measurement.

non-linear nature of the system. The EKF consists of two fundamental parts, the model update and the measurement update. The estimation of the state vector $\mathbf{x}$ will be denoted as $\hat{\mathbf{x}}$ and an estimated vector before and after the measurement update will be indicated with a superscript '$-$' and '$+$' respectively. The general form for a system model can be expressed as

$$\dot{\mathbf{x}}_t = \boldsymbol{f}(\mathbf{x}_t) + \mathbf{s}_t, \tag{4.30}$$

where $\boldsymbol{f}(\mathbf{x}_t)$ is a non-linear function of $\mathbf{x}_t$ and $\mathbf{s}_t$ is the system noise. To linearise the function $\boldsymbol{f}(\mathbf{x}_t)$ an approximation of $\boldsymbol{f}(\mathbf{x}_t)$ according to the Taylor series expansion is implemented as

$$\begin{aligned} \boldsymbol{f}(\mathbf{x}_t) &= \boldsymbol{f}(\hat{\mathbf{x}}_t) + \left[\frac{\partial \boldsymbol{f}}{\partial \hat{\mathbf{x}}_t}\right](\mathbf{x}_t - \hat{\mathbf{x}}_t) + \frac{1}{2!}\left[\frac{\partial^2 \boldsymbol{f}}{\partial \hat{\mathbf{x}}_t^2}\right](\mathbf{x}_t - \hat{\mathbf{x}}_t)^2 \\ &\approx \boldsymbol{f}(\hat{\mathbf{x}}_t) + \mathbf{F}_t \Delta\mathbf{x}_t, \\ \text{where} \quad \mathbf{F}_t &= \left[\frac{\partial \boldsymbol{f}}{\partial \hat{\mathbf{x}}_t}\right] \\ \text{and} \quad \Delta\mathbf{x}_t &= \mathbf{x}_t - \hat{\mathbf{x}}_t. \end{aligned} \tag{4.31}$$

The state vector $\mathbf{x}$ consists of the quaternion $\mathbf{q}$ and the inertial-referenced angular velocity $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}$ given as

$$\mathbf{x} = [\mathbf{q}, \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}]^{\mathrm{T}}. \tag{4.32}$$

The model update of the EKF is calculated with the dynamics and kinematics of the system model as discussed in Section 4.1.4. This is used to calculate $\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}$. The model update of the EKF is implemented with Algorithm 4.1. The disturbance torques only include $\boldsymbol{\tau}_{gg}$ and $\boldsymbol{\tau}_{\text{gyro}}$ since they can be accurately modelled on the satellite.

Algorithm 4.1: Model update of the EKF at time step k with Runge-Kutta 4^{th} order.

Satellite Body Moment of Inertia $\mathbf{J}$
Time step $(T_s) = 1s$
Number of iterations $(I) = 10$
Step size $h = \frac{T_s}{I}$
Disturbance torques $\boldsymbol{\tau_d} = \boldsymbol{\tau}_{gg} - \boldsymbol{\tau}_{\text{gyro}}$
Control torques $\boldsymbol{\tau_c} = \boldsymbol{\tau}_m - \boldsymbol{\tau}_w$
$\boldsymbol{\tau} = \boldsymbol{\tau_c} + \boldsymbol{\tau_d}$
for $n := 1$ **to** I **do**
    $k_1 = h(\mathbf{J^{-1}}\boldsymbol{\tau})$
    $k_2 = h(\mathbf{J^{-1}}\boldsymbol{\tau} + \frac{k1}{2})$
    $k_3 = h(\mathbf{J^{-1}}\boldsymbol{\tau} + \frac{k_2}{2})$
    $k_4 = h(\mathbf{J^{-1}}\boldsymbol{\tau} + k_3)$
    $\boldsymbol{\omega}_{n+1} = \boldsymbol{\omega}_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}$
end for
$(\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}})_k^- = \boldsymbol{\omega}_{n+1}$
return $(\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}})_k^-$

With reference to G. Janse van Vuuren [22], $\hat{\mathbf{q}}_k^-$ can be calculated as

$$\hat{\mathbf{q}}_k^- = \left[\cos(k_q)\mathbf{I}_{4\times 4} + \frac{1}{\|(\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{O}})_k^-\|}\sin(k_q)\boldsymbol{\Omega}_k^-\right]\hat{\mathbf{q}}_{k-1}^+$$

$$\text{where} \qquad k_q = \frac{Ts}{2}\|(\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{O}})_k^-\|$$

$$\begin{aligned}(\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{O}})_k^- &= (\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}})_k^- - \hat{\boldsymbol{A}}_{\mathcal{O}_k}^{\mathcal{B}}\begin{bmatrix}0 & -(\omega_o)_k & 0\end{bmatrix}^{\mathrm{T}} \\ &= \begin{bmatrix}\hat{\boldsymbol{\omega}}_{\bar{x}_\mathcal{O}} & \hat{\boldsymbol{\omega}}_{\bar{y}_\mathcal{O}} & \hat{\boldsymbol{\omega}}_{\bar{z}_\mathcal{O}}\end{bmatrix}^{\mathrm{T}}\end{aligned} \tag{4.33}$$

$$\|(\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{O}})_k^-\| = \sqrt{\hat{\boldsymbol{\omega}}_{\bar{x}_\mathcal{O}}^2 + \hat{\boldsymbol{\omega}}_{\bar{y}_\mathcal{O}}^2 + \hat{\boldsymbol{\omega}}_{\bar{z}_\mathcal{O}}^2}$$

$$\text{and } \boldsymbol{\Omega}_k^- = \begin{bmatrix}0 & \hat{\boldsymbol{\omega}}_{\bar{z}_\mathcal{O}} & -\hat{\boldsymbol{\omega}}_{\bar{y}_\mathcal{O}} & \hat{\boldsymbol{\omega}}_{\bar{x}_\mathcal{O}} \\ -\hat{\boldsymbol{\omega}}_{\bar{z}_\mathcal{O}} & 0 & \hat{\boldsymbol{\omega}}_{\bar{x}_\mathcal{O}} & \hat{\boldsymbol{\omega}}_{\bar{y}_\mathcal{O}} \\ \hat{\boldsymbol{\omega}}_{\bar{y}_\mathcal{O}} & -\hat{\boldsymbol{\omega}}_{\bar{x}_\mathcal{O}} & 0 & \hat{\boldsymbol{\omega}}_{\bar{z}_\mathcal{O}} \\ -\hat{\boldsymbol{\omega}}_{\bar{x}_\mathcal{O}} & -\hat{\boldsymbol{\omega}}_{\bar{y}_\mathcal{O}} & -\hat{\boldsymbol{\omega}}_{\bar{z}_\mathcal{O}} & 0\end{bmatrix}$$

The estimated state vector, $\hat{\mathbf{x}}_k^-$ can now be expressed as

$$\hat{\mathbf{x}}_k^- = \begin{bmatrix} (\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}})_k^- & \hat{\mathbf{q}}_k^- \end{bmatrix} \tag{4.34}$$

The derivation of certain matrices required for the model update are provided in the appendices. The overview of these matrices, however, are given to provide a better understanding of the EKF. The first matrix is the system noise covariance matrix $\mathbf{Q}_k$ which is assumed to be zero-mean and Gaussian and is derived in Appendix 9.3. The system noise covariance matrix $\mathbf{Q}_k$ represents the discrete system noise $\mathbf{s}_k$. The second matrix is that of the discrete system perturbation model Φ_k which is derived in Appendix 9.3.

The state covariance matrix $\mathbf{P}_k$ can be propagated as

$$\mathbf{P}_k^- = \Phi_k \mathbf{P}_{k-1}^+ \Phi_k^{\mathrm{T}}, \tag{4.35}$$

where Φ_k is the discrete system perturbation model calculated as

$$\begin{aligned} \Phi_k &= \left[e^{T_s \mathbf{F}_t} \right]_{\mathbf{x}=\hat{\mathbf{x}},\quad t=kT_s} \\ \text{and simplified as} \qquad \Phi_k &\approx \left[\mathbf{I} + T_s \mathbf{F}_t + \frac{1}{2!} T_s^2 \mathbf{F}_t^2 \right]_{\mathbf{x}=\hat{\mathbf{x}},\quad t=kT_s} \end{aligned} \tag{4.36}$$

according to W. Steyn [47]. This concludes the model update of the EKF and the EKF must now be updated with the sensor measurements.

The measurement update, thereafter, is implemented sequentially for each sensor starting with the sensor with the largest noise. The measurement model in it's general form is provided as

$$\mathbf{y}_t = \boldsymbol{h}(\mathbf{x}_t) + \mathbf{m}_t. \tag{4.37}$$

The function $\boldsymbol{h}(\mathbf{x}_t)$ is a non-linear function and to approximate the function discretisation is required. The discretisation of function $\boldsymbol{h}(\mathbf{x}_t)$ according to G. Janse van Vuuren [22] can be represented as

$$\begin{aligned} \boldsymbol{h}(\mathbf{x}_k) &\approx \boldsymbol{h}(\hat{\mathbf{x}}_k) + \mathbf{H}_k \Delta \mathbf{x}_k, \\ \text{where} \qquad \mathbf{H}_k &= \left[\frac{\partial \boldsymbol{h}}{\partial \mathbf{x}_k} \right]_{\mathbf{x}_k = \hat{\mathbf{x}}_k}. \end{aligned} \tag{4.38}$$

The discrete measurement perturbation Jacobian matrix $\mathbf{H}_k$ is derived in Appendix 9.3. The discrete measurement model $\mathbf{y}_k$ can be calculated as

$$\mathbf{y}_k = \boldsymbol{A}_{\mathcal{O}_k}^{\mathcal{B}} \mathbf{v}_{\mathcal{O}} + \mathbf{m}_k. \tag{4.39}$$

The estimated discrete measurement model $\hat{\mathbf{y}}_k$ can therefore be calculated as

$$\hat{\mathbf{y}}_k = \hat{\boldsymbol{A}}_{\mathcal{O}_k}^{\mathcal{B}} \mathbf{v}_{\mathcal{O}}. \tag{4.40}$$

The estimated DCM $\hat{\boldsymbol{A}}_{\mathcal{O}_k}^{\mathcal{B}}$ is calculated according to Equation 4.9 where $\mathbf{q} = \hat{\mathbf{q}}$. The innovation noise $\mathbf{m}_k$ of the measurement model is zero-mean Gaussian noise with covariance matrix $\mathbf{R}_k$ as derived in Appendix 9.3.

To determine the measurement update the error between the modelled sensor vector $\mathbf{v}_{\mathcal{O}}$ and the measured vector $\mathbf{v}_{\mathcal{B}}$ is required. The error $\mathbf{e}_k$ between the measured and modelled vector is calculated as

$$\mathbf{e}_k = \mathbf{v}_{\mathcal{B}} - \hat{\boldsymbol{A}}_{\mathcal{O}_k}^{\mathcal{B}} \mathbf{v}_{\mathcal{O}}, \tag{4.41}$$

To determine the influence of $\mathbf{e}_k$ on the updated state vector $\hat{\mathbf{x}}_k^+$ the gain matrix $\mathbf{K}_k$ is used. The gain matrix $\mathbf{K}_k$ can be calculated as

$$\mathbf{K}_k = \mathbf{P}_k^- (\mathbf{H}_k^-)^{\mathrm{T}} \left[\mathbf{H}_k^- \mathbf{P}_k^- (\mathbf{H}_k^-)^{\mathrm{T}} + \mathbf{R}_k \right]^{-1}, \tag{4.42}$$

after which the updated state vector $\hat{\mathbf{x}}_k^+$ can be calculated as

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{e}_k. \tag{4.43}$$

The state covariance matrix can then be updated as

$$\mathbf{P}_k^+ = \left[\mathbf{I}_{7\times 7} - \mathbf{K}_k \mathbf{H}_k^+ \right] \mathbf{P}_k \left[\mathbf{I}_{7\times 7} - \mathbf{K}_k \mathbf{H}_k^+ \right] + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^{\mathrm{T}}. \tag{4.44}$$

This concludes the measurement update of the EKF. To validate the results of the EKF, the estimation error or estimation metric for the first two orbits is given in Figure 4.12. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

The estimation metric indicates that the estimated quaternion $\hat{\mathbf{q}}_k$ is accurate to within 6° of the actual quaternion $\mathbf{q}_k$. This thesis does not focus on providing an estimation with a smaller estimation metric since the estimation metric of 6° provides stable estimation for control and therefore the EKF is considered to perform adequately.

During the measurement update, the error $\mathbf{e}_k$ is largely affected by anomalous behaviour in the sensor measurements. It is for this reason that FDIR of sensor anomalies are required.

4.3.3. Attitude Control

To ensure that the satellite is able to satisfy the mission requirements, control of the satellite attitude is required. The satellite's payload must therefore be in the direction of the Earth during eclipse and the solar panels should be pointing in the direction of the Sun during the sunlit phase. For this a quaternion feedback controller with the reaction wheels as actuators is implemented. Momentum dumping with the magnetorquers as actuators is implemented to ensure that the wheel disturbance remains within reasonable

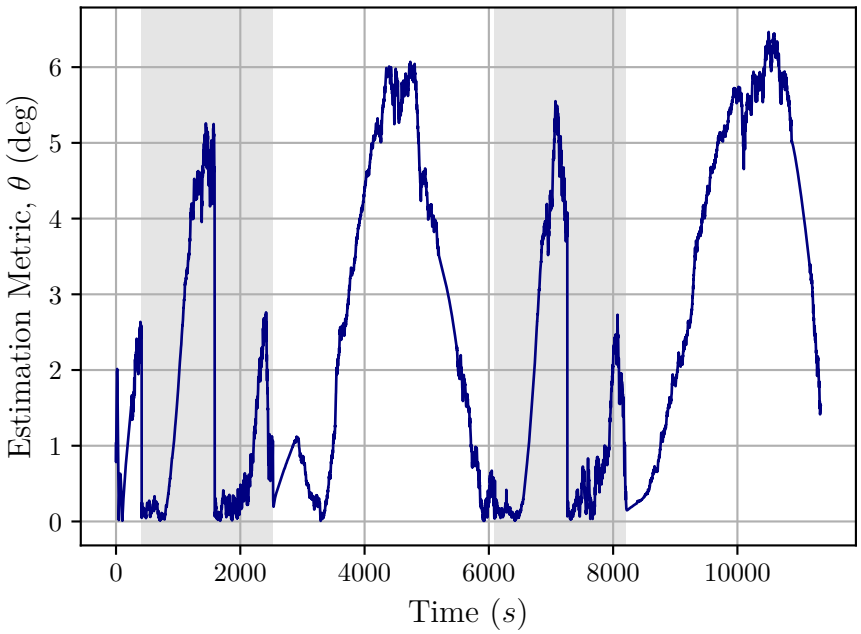

Figure 4.12: Estimation metric in degrees during the first two orbits during normal operation. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

boundaries [22].

Quaternion Feedback Controller

To ensure that the satellite is in the desired orientation with stable control in all three axes, the quaternion feedback reaction wheel controller is implemented [48]. The controller is provided with the state vector $\hat{\mathbf{x}}$ as input and outputs the desired reaction wheel torque $\boldsymbol{\tau}_w$. To calculate the required torque $\boldsymbol{\tau}_w$ the definition according to W. Steyn [49] for all cases at time step k is given as

$$\boldsymbol{\tau}_w = K_P \mathbf{J}\mathbf{q}_{err} + K_D \mathbf{J}\hat{\boldsymbol{\omega}}_{err} - \hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}} \times \left[\mathbf{J}\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}} + \mathbf{h}_w\right], \tag{4.45}$$

where the angular rate error $\hat{\boldsymbol{\omega}}_{err}$ is the difference between the estimated angular rate $\hat{\boldsymbol{\omega}}$ and the reference $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}} = \mathbf{0}$. The measured angular momentum of the wheels $\mathbf{h}_w$ is the actual momentum of the wheels with additional zero-mean Gaussian noise. The control gains according to G. Janse van Vuuren [22] can be defined as

$$\begin{aligned} K_P &= 2\omega_n^2 \\ K_D &= 2\zeta\omega_n. \end{aligned} \tag{4.46}$$

The quaternion error $\mathbf{q}_{err}$ is calculated with the quaternion difference operator Θ as

$$\begin{aligned} \mathbf{q}_{err} &= \mathbf{q_c}\Theta\hat{\mathbf{q}} \\ \begin{bmatrix} q_{1e} \\ q_{2e} \\ q_{3e} \\ q_{4e} \end{bmatrix} &= \begin{bmatrix} q_{4c} & q_{3c} & -q_{4c} & -q_{4c} \\ -q_{3c} & q_{4c} & q_{1c} & -q_{2c} \\ q_{2c} & -q_{1c} & q_{4c} & -q_{3c} \\ q_{1c} & q_{2c} & q_{3c} & q_{4c} \end{bmatrix} \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \\ \hat{q}_4 \end{bmatrix}. \end{aligned} \tag{4.47}$$

The current estimated quaternion $\hat{\mathbf{q}}$ is provided by the EKF and the command quaternion $\mathbf{q}_c$ is $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$ during eclipse. During the Sun following phase, the command quaternion $\mathbf{q}_c$ according to X. Chen [50] can be calculated as

$$\mathbf{q}_c = \begin{bmatrix} \bar{\mathbf{u}}_c \sin(\frac{\delta}{2}) \\ \cos(\frac{\delta}{2}) \end{bmatrix}, \tag{4.48}$$

where

$$\bar{\mathbf{u}}_c = \frac{\bar{\mathbf{u}}_\mathrm{s} \times \bar{\mathbf{s}}_\mathcal{O}}{\|\bar{\mathbf{u}}_\mathrm{s} \times \bar{\mathbf{s}}_\mathcal{O}\|}. \tag{4.49}$$

The modelled unit sun vector $\bar{\mathbf{s}}_\mathcal{O}$ in ORC is provided by the Sun's position model. The main solar panel's position is denoted as a unit vector $\bar{\mathbf{u}}_\mathrm{s}$. The angle between $\bar{\mathbf{u}}_\mathrm{s}$ and $\bar{\mathbf{s}}_\mathcal{O}$ denoted as δ, can be calculated with the vector dot-product. The command quaternion $\mathbf{q_c}$ can then be used as the reference for the control.

The pointing error or the pointing metric is shown in Figure 4.13 to demonstrate the efficacy of the quaternion feedback controller in combination with the EKF during the first two orbits of simulation. The pointing metric is the difference between the command quaternion $\mathbf{q}_c$ and the actual quaternion $\mathbf{q}$. It must be noted that the large spikes of the pointing metric in Figure 4.13 is due to change in the mode of operation.

Momentum Dumping Controller

Momentum dumping is crucial to ensure that the wheel disturbance does not cause the system to become unstable. Momentum dumping is implemented continually during orbit to ensure that the momentum of the reaction wheels are kept at a minimum. The momentum dumping is implemented with magnetorquers based on a Cross-Product controller [22]. The magnetic dipole moment $\mathbf{m}$ is calculated as

$$\mathbf{m} = \frac{\mathbf{e} \times \mathbf{b}_\mathcal{B}}{\|\mathbf{b}_\mathcal{B}\|^2}, \tag{4.50}$$

where $\mathbf{b}_\mathcal{B}$ is the geomagnetic field in SBC and the error vector $\mathbf{e}$ can be calculated as

$$\mathbf{e} = -K_w(\mathbf{h}_w - \mathbf{h}_{w,ref}) \tag{4.51}$$

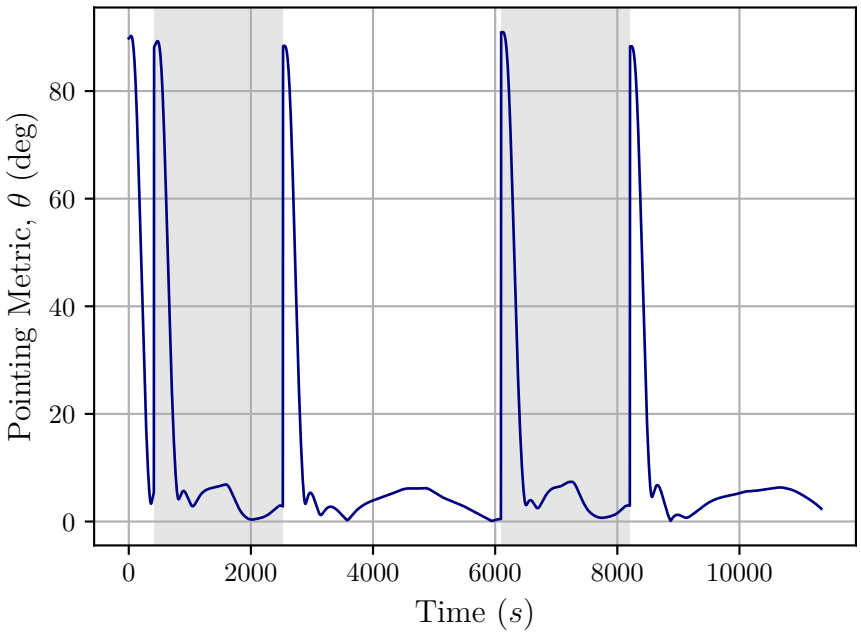

Figure 4.13: The efficacy of the quaternion feedback controller and the EKF provided as the pointing metric. The pointing metric is the difference between the command quaternion $\mathbf{q}_c$ and the actual quaternion $\mathbf{q}$.

where K_w is a positive gain, $\mathbf{h}_w$ is the measured angular momentum of the wheels and $\mathbf{h}_{w,ref}$ is the desired angular momentum of the wheels (set as $\mathbf{0}$). The magnetorquers torques $\boldsymbol{\tau}_m$ can then be calculated as [22]

$$\boldsymbol{\tau}_m = \mathbf{m} \times \mathbf{b}_{\mathcal{B}} \tag{4.52}$$

and are shown in Figure 4.14. It is evident that when the satellite control changes from eclipse to sunlit and from sunlit to eclipse the magnetorquers compensate for the increase in reaction wheel torques and minimise the reaction wheel disturbance.

4.4. Simulation Overview

To demonstrate how the different aspects of the simulation environment and the FDIR fit together a block diagram for the fault detection, isolation and recovery (FDIR) of the EKF within the ADCS system is provided in Figure 1.2. The FDIR is provided with both the inputs from the feature extraction component as well as the sensor measurements to predict whether an anomaly has occurred. Thereafter, the anomaly must be isolated and therefore classified as to which practical anomaly caused the current sensor measurements. The anomaly is then recovered depending on the recovery method and anomaly type.

The anomalies discussed and modelled in this thesis are specific to the design of the satellite as discussed in Chapter 3. The practical anomalies are the solar reflection from the

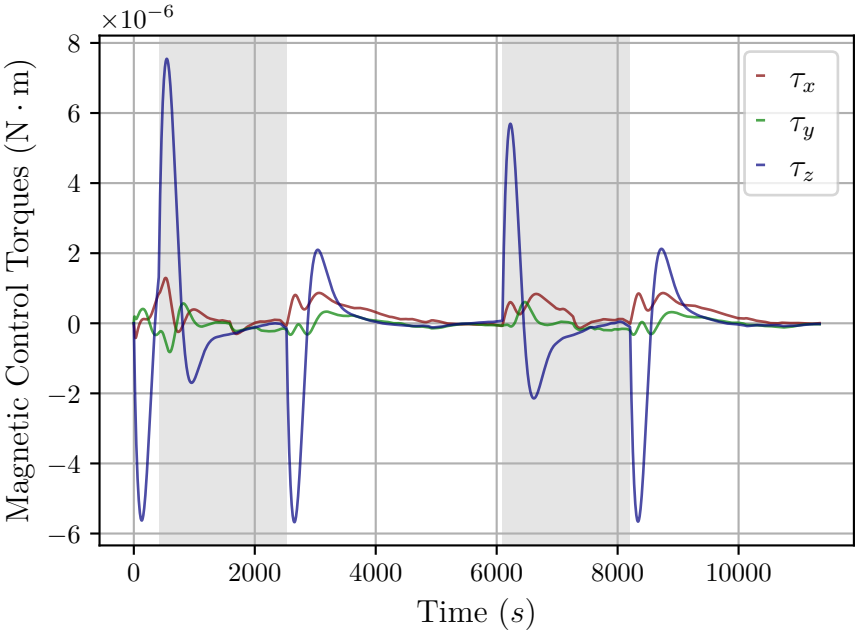

Figure 4.14: Magnetic control torques of the magnetorquers to illustrate the magnitude of the torques to ensure that the momentum of the reaction wheels remain at a minimum. This is implemented to ensure that the reaction wheel disturbance torque is decreased. The legend provides the axis of the coordinate frame for which the magnitude of the torque is given.

solar panels on the sun sensor, the Moon on the Earth's horizon influencing the algorithm of an infrared horizon sensor and magnetic moment disturbances caused by the magnetic induced dipole moment of the solar panels influencing the magnetometer. The general sensor anomalies is that of high noise. The actuator failure is that of a reaction wheel not responding to control inputs.

The theoretical background for the environment as well as the ADCS was discussed in this chapter. This provides all the necessary elements to simulate the environment to create a database for the training of the different detection methods. This also provides the environment to induce the anomalies as modelled in Chapter 5. With all these elements the overview of the software implementation can be given in Figure 4.15. The flow of the calculations are provided and none of the blocks in the flow diagram are *If Statements*, Figure 4.15 only provides the sequence of how the elements fits together.

The first block of Figure 4.15 represents the initialisation of the environment parameters. These parameters include, but are not limited to, the sensors and actuators positions as discussed in Chapter 3 as well as the initial quaternion $\mathbf{q}_0$ and $\boldsymbol{\omega}^{\mathcal{I}}_{\mathcal{B}_0}$. Thereafter the second block represents the environmental factors such as the Sun position, geomagnetic field, orbit propagation and the associated factors such as the eclipse. The disturbance torques $\boldsymbol{\tau}_d$ can be calculated according to the current state vector $\mathbf{x}_k$ and in parallel the sensor measurements can be calculated according to the sensor models. It is during the sensor

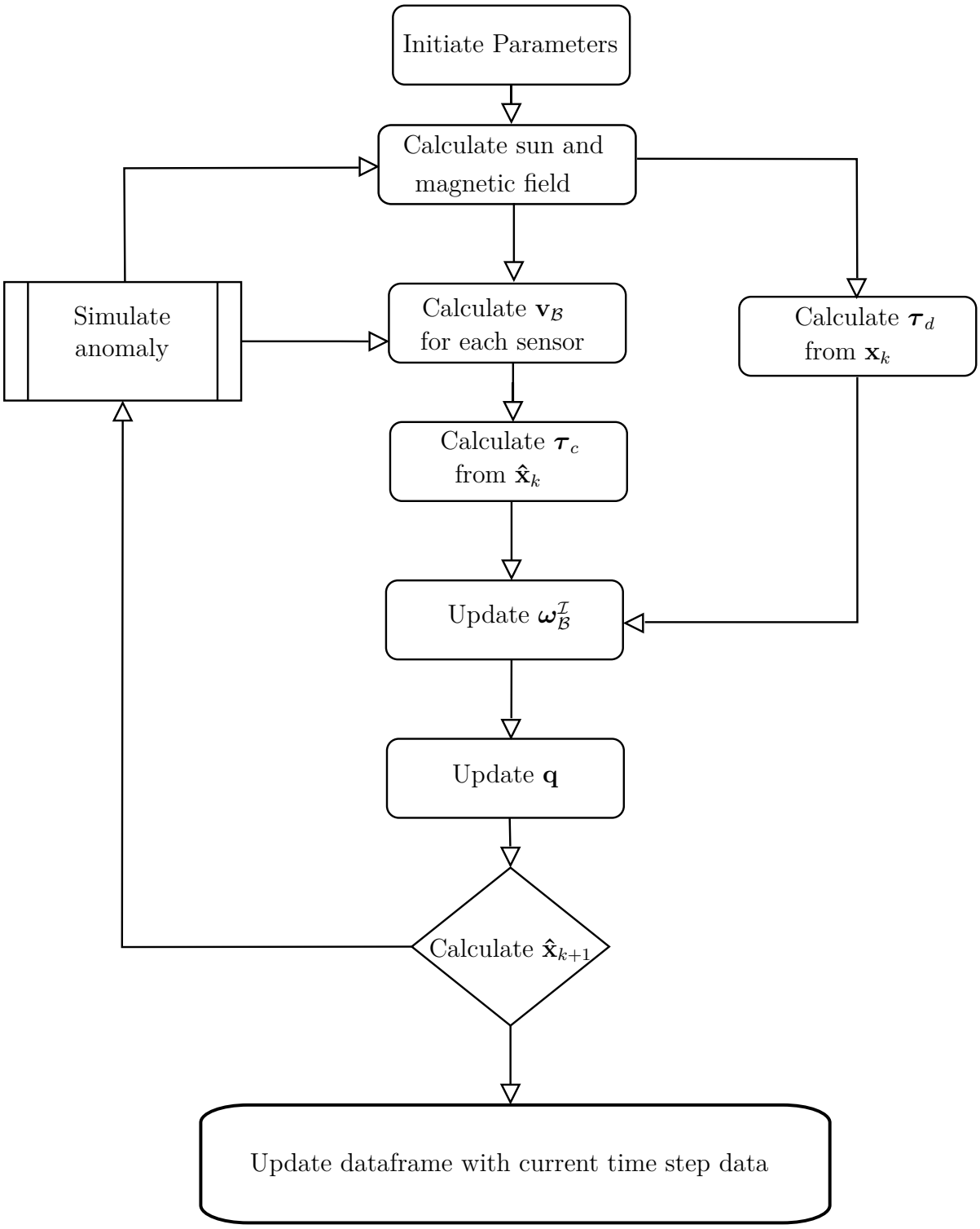

Figure 4.15: The overview of the simulation provided as a block diagram.

measurement calculations where the models of the anomalies are implemented. After the calculation of the current sensor measurements, the control torque $\boldsymbol{\tau}_c$ can be calculated from the quaternion feedback controller and momentum dumping.

The control torque $\boldsymbol{\tau}_c$ and the disturbance torque $\boldsymbol{\tau}_d$ is used to calculated the angular rate $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}$. The calculated angular rate can thereafter be used to calculated $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}}$ and $\mathbf{q}$. From

the current sensor measurements and the control torque $\boldsymbol{\tau}_c$ can implemented in the EKF algorithm to calculate the estimated state vector $\hat{\mathbf{x}}_{k+1}$. After a single time step of the simulation has passed, the database can be updated for the training of the detection and isolation algorithms.

4.5. Summary

In this chapter the theoretical background for the satellite environment is discussed. This includes the coordinate frames, orbit propagation, attitude, satellite kinematics and dynamics as well as environmental factors such as the Sun's position and disturbances torques such as the gravity gradient disturbance torque. This provides a realistic environment for the testing of the FDIR methods as well as an environment in which the anomalies can be modelled. The background of the EKF and the quaternion feedback control and momentum dumping is discussed to provide insight into the controller block of Figure 1.2.

This background is provided to ensure that the reader understands the specific environment in which the FDIR is required as well as the challenges for FDIR in the given context. The simulation can be implemented after the discussion of the ADCS implementation and the environmental factors, where the overview of the simulation is given in the software block diagram of Figure 4.15. This chapter provides the background for the normal operation of the satellite and discusses the controller, EKF, simulated environment and sensors blocks of Figure 1.2.

Chapter 5

Anomalies

To ensure that the prediction and classification of anomalies are not based on generalised sensor failures an anomaly for each sensor is modelled as discussed in Chapter 2. These anomalies are either chosen to show the significant effect of these anomalies on the ADCS or are modelled based on research that label the anomaly as a possible influence on the ADCS as discussed in Chapter 4. There is an anomaly for each sensor, that will create inaccuracies for that specific sensor measurement. An anomaly for the reaction wheels is also implemented to show the resulting estimation failure based on an inaccurate model update, since the control torque $\boldsymbol{\tau}_w$ and the torque implemented on the reaction wheel would not be the same. All anomalies will also be predicted based on the sensor readings and outputs from feature extractions, since the effect will be evident on all the sensors.

There are three separate classes of anomalies that influence the EKF. The first are anomalies that only influence sensor measurements. This can be general anomalies such as high noise or sudden failure as well us specific modelled anomalies that are highly dependent on the orbital nature of the satellite, such as the reflection of solar panels on a sun sensor and the Moon in the horizon sensor's field of view. Another class is that of anomalies that influence both the sensor measurements and cause an external disturbance torques on the satellite. An example of this is the magnetic moment disturbance anomaly that influence the magnetometer as well as the magnetorquers. The last class is that of actuator failures that influence the model of the EKF. Examples of this is that of sudden reaction wheel failures.

5.1. Reflection of Solar Panels on Sun Sensor

The Sun reflection anomaly occurs when the Sun reflects from the solar panels unto the sun sensor. This influences the sun vector measured by the sun sensor. This is a practical anomaly that is dependent on the orbital nature of the satellite.

The Sun reflection anomaly is modelled for the specific shape and design of the satellite as shown in Figure 3.3. The modelling takes place within the satellite body coordinate (SBC) frame. The modelling of the Sun reflection anomaly from the solar panels unto the sun sensor is illustrated in Figure 5.1. The incoming sun vector $\mathbf{r}_{\text{sun}}$ reflects from the solar

panel as the vector $\mathbf{r}_{\text{ref}}$.

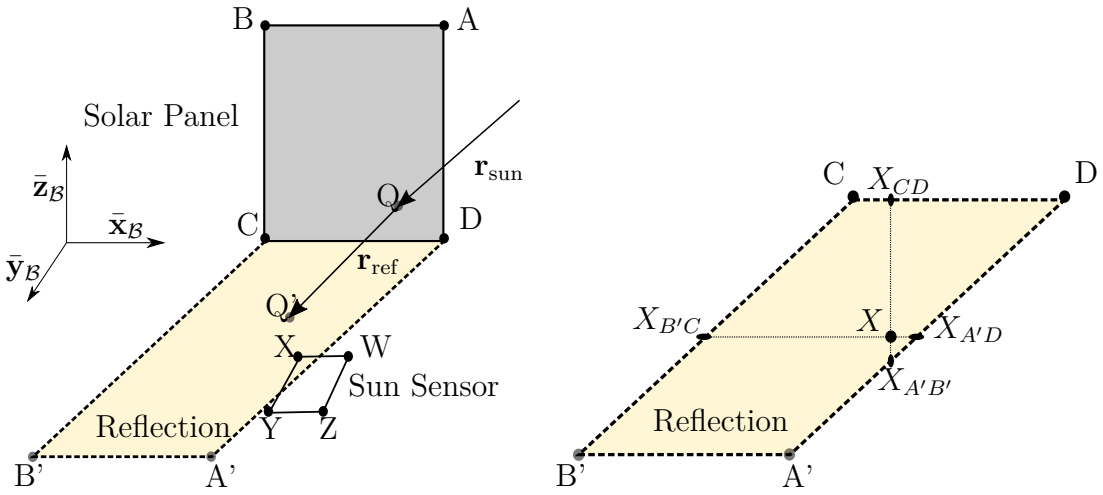

Figure 5.1: Visual representation of the modelling for the Sun reflection from the solar panels unto the sun sensor. The reflected position of a point A from the solar panel on the geometric plane of the sun sensor $WXYZ$ is denoted as A'.

The solar panel is modelled as a geometric plane, based on the assumption that the solar panel's surface is more or less smooth and the reflection area will be similar to that of a perfectly smooth mirror. This model also assumes, that if the sun sensor detects any reflection from the solar panel, the measured sun vector will default to the reflection ray instead of the direct sun vector. In practice, this is a function of the exact detection algorithm within the sensor, and some reflections might be ignored. This assumption will, however, produce the worst-case behaviour. The intensity of the light vector is also disregarded.

The solar panel geometric plane $ABCD$ can be represented in the SBC by a point and unit normal vector to the plane defined as

$$\begin{aligned} \mathbf{p}_{ABCD} &= [p_x,\ p_y,\ p_z]^T, \\ \text{and} \qquad \bar{\mathbf{n}}_{ABCD} &= [n_x,\ n_y,\ n_z]^T, \end{aligned} \tag{5.1}$$

respectively. Similarly, the sun sensor geometric plane $WXYZ$ is represented by the point $\mathbf{p}_{WXYZ}$ and normal vector $\bar{\mathbf{n}}_{WXYZ}$.

The reflected sun vector $\mathbf{r}_{\text{ref}}$ can be calculated by

$$\mathbf{r}_{\text{ref}} = \bar{\mathbf{r}}_{\text{sun}} - 2\bar{\mathbf{n}}_{ABCD}^{\top}(\bar{\mathbf{r}}_{\text{sun}} \cdot \bar{\mathbf{n}}_{ABCD}), \tag{5.2}$$

where $\bar{\mathbf{r}}_{\text{sun}}$ is the incoming unit sun vector. To calculate the intersection of the reflected vector with the geometric plane of the sun sensor $WXYZ$, the equation of $WXYZ$, the reflected unit sun vector $\bar{\mathbf{r}}_{\text{ref}}$, and the point of origin is required. The reflection of the sun vector is illustrated in Figure 5.1. The reflection from the point Q to the point Q' can thus be calculated as a projection of $\mathbf{r}_{\text{ref}}$ unto the $WXYZ$-plane.

To model reflection from the solar panels to the sun sensor, only two corners of the solar panel and two corners of the sun sensor are to be considered. From Figure 5.1 it is evident that if the solar panel reflects on Y that the reflection will also cover X. The same is true for corner Z and W. Since C' will be at the same position as C, which is valid for D' and D, the calculation can be omitted. It is therefore only necessary to calculate the reflected positions A' and B'. This simplifies the reflection model significantly.

The reflected position A' can be calculated as the intersection of the reflected vector $\mathbf{r}_{\text{ref}}$ with plane $WXYZ$. The position of A is also known, based on the satellite design, and A' can therefore be calculated. The same applies to B and B'. To determine whether Y or X is within the reflection region, it is assumed that the plane $WXYZ$ is a 2D plane, and the third dimension is omitted. The axes therefore change from x, y, z to only x, y. To calculate whether X is between the lines of $A'D$ and $B'C$ and between the lines CD and $A'B'$ the line equation between reflected points in the form

$$y_{A'B'} = mx_{A'B'} + c, \tag{5.3}$$

can be implemented. From the coordinates of A' and B', the corresponding $X_{A'B',y}$ can be calculated by substituting X_x into Equation 5.3. With the same method the coordinates of $X_{B'C}$, $X_{A'D}$, $X_{A'B'}$ and X_{CD} can be determined. After that, with logical *If Statements*, it can be determined whether X is in the reflection zone. If X_x is to the right of $X_{B'C,x}$ and to the left of $X_{A'D,x}$, as well as X_y is above $X_{A'B',y}$ and below $X_{CD,y}$ then X is within the reflection zone.

The results for the sun vector with reflection are shown in Figure 5.2. There is a clear difference between the sun vector during normal operation provided in Figure 4.11 and the measurement influenced by the reflection. This sun reflection vector may affect the estimation and, thus also the attitude control of the satellite.

5.1.1. Influence of the Sun Reflection Anomaly on the Estimation

To determine whether the reflection on the sun sensor has an influence on the ADCS, the estimation metric is shown in Figure 5.3. The estimation metric, is the angle difference between the true quaternion $\mathbf{q}$ and estimated quaternion $\hat{\mathbf{q}}$ in degrees. It is evident that the reflection has a large influence on the estimation when Figure 5.3 is compared with Figure 4.12. The maximum estimation error is 6° for normal operation and it is evident in Figure 5.3 that the estimation error is sometimes larger than 150°. It is also clear that during the eclipse the estimation returns to a more accurate estimation. This is due to the fact that sun measurement is not available during eclipse.

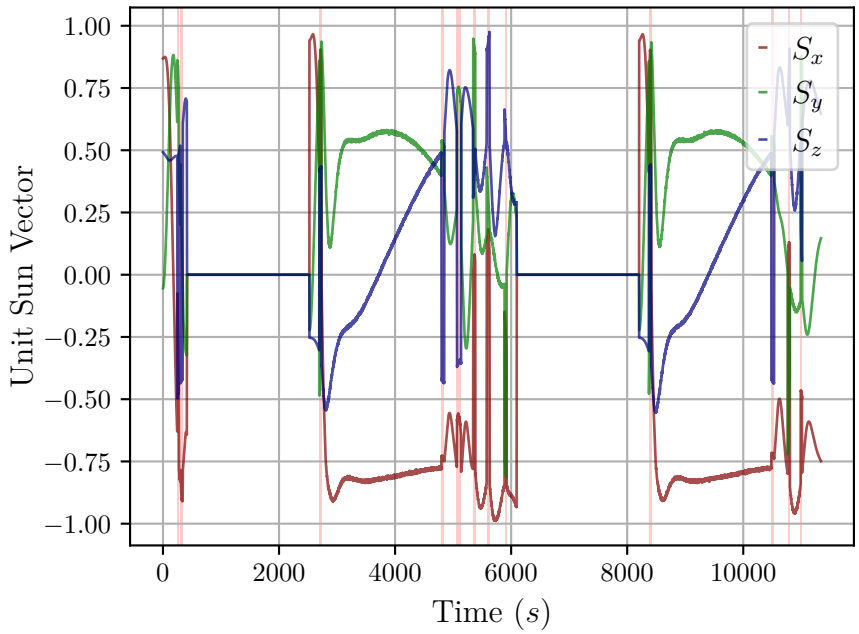

Figure 5.2: Measured sun vector with Sun reflections in SBC. The red lines of the graph are the periods when the anomaly occurs. The legend provides the axis of the coordinate frame for the unit vector of the sun measurement.

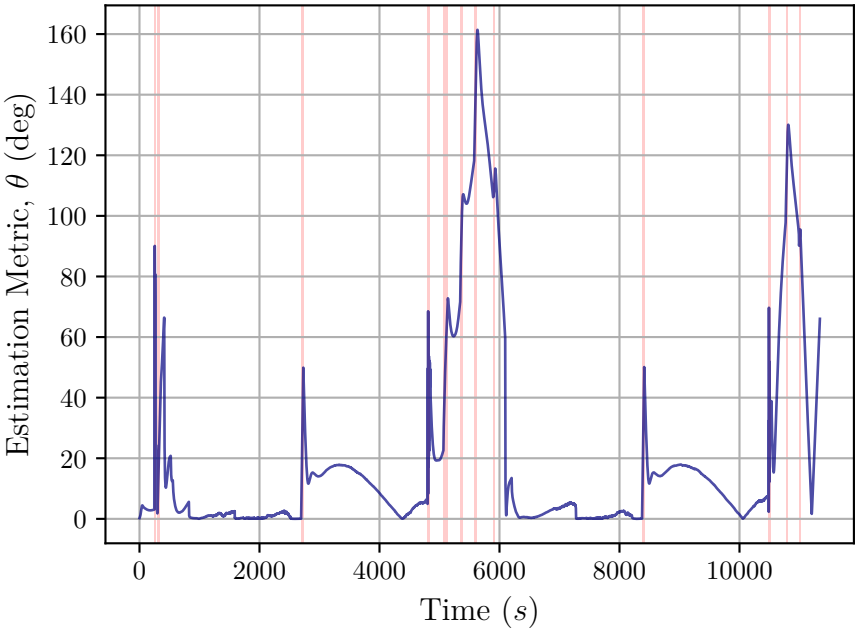

Figure 5.3: Estimation metric in degrees during the first two orbits due to the Sun reflection anomaly. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees. The red lines of the graph are the periods when the anomaly occurs.

5.2. The Moon in Field of View of the Horizon Sensor

An anomaly that can be experienced by an infrared (IR) horizon sensor is when the Moon is overlapping the horizon of the Earth in the horizon sensor's field of view (FoV). This influences the edge detection and circular fit algorithm [23, 51] and consequently the calculated centre of the Earth. It is required to simulate the image seen by the horizon sensor and thereafter the algorithm for detecting the centre of the Earth can be implemented.

5.2.1. Simulating the Horizon Sensor's Infra-red Image

To model and simulate the horizon sensor's image a few vectors providing the relative positions of the objects should be determined. Firstly, the vectors of both the satellite-to-Earth $\mathbf{r}_{SE}$ and Earth-to-Moon $\mathbf{r}_{EM}$ are required. The Moon's position is determined with the Julian date, since the propagation of the Moon position relative to the centre of the Earth has been calculated in advance. These vectors are shown in Figure 5.4.

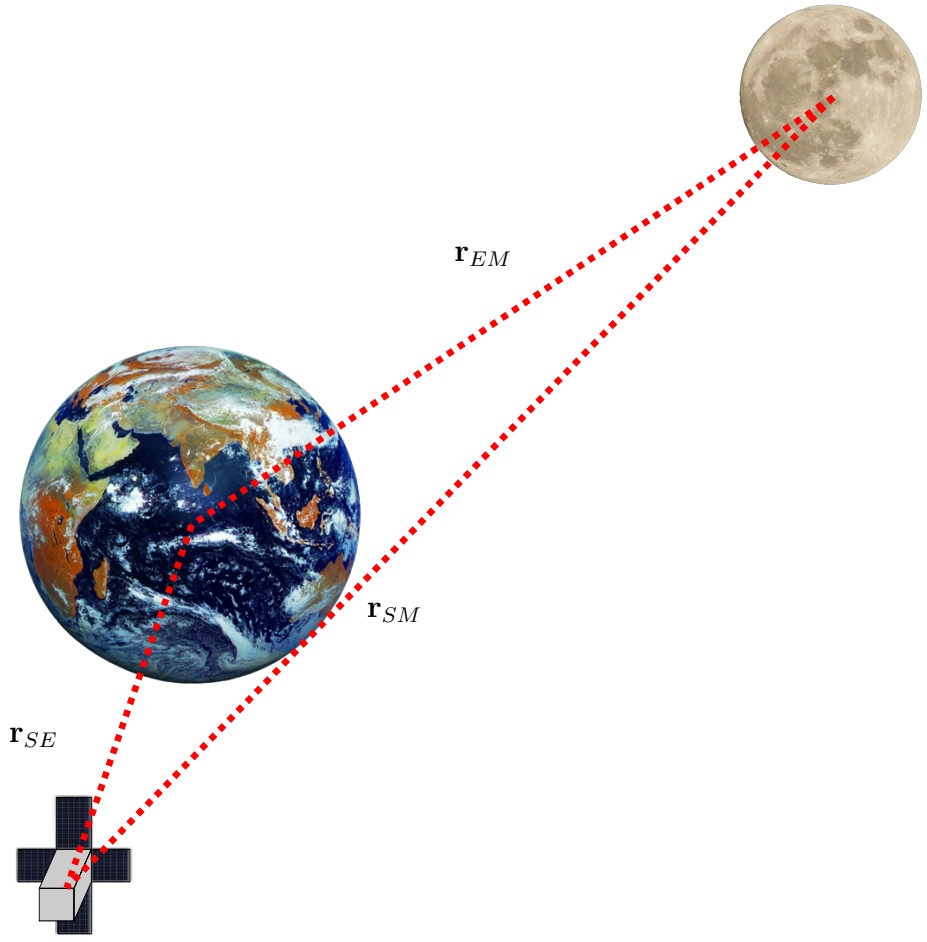

Figure 5.4: Visual representation of the Earth to the Moon $\mathbf{r}_{EM}$, the satellite-to-Earth $\mathbf{r}_{SE}$ and the satellite-to-Moon $\mathbf{r}_{SM}$ vectors.

From the vector $\mathbf{r}_{SE}$ and the position of the centre of the Earth P_{Earth} a 3D plane normal to the vector $\mathbf{r}_{SE}$ and at the position P_{Earth} can be calculated. The position P_{Earth} and the vector $\mathbf{r}_{SE}$ are defined as

$$P_{Earth} = [p_x,\ p_y,\ p_z], \tag{5.4}$$

and

$$\mathbf{r}_{SE} = [\mathbf{r}_x,\ \mathbf{r}_y,\ \mathbf{r}_z]. \tag{5.5}$$

The equation for the 3D plane is defined as

$$Ax + By + Cx = D, \tag{5.6}$$

where the parameters A, B, C, D can be calculated as

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \mathbf{r}_x \\ \mathbf{r}_y \\ \mathbf{r}_z \\ \mathbf{r}_x p_x + \mathbf{r}_y p_y + \mathbf{r}_z p_z \end{bmatrix} \tag{5.7}$$

This 3D plane slices the Earth in half as shown in Figure 5.5. The Moon and the Earth can both be projected unto the 3D plane to determine the image seen by the horizon sensor. The horizon sensor's centre of view vector must also be projected unto the 3D plane. A circle can be drawn for the Earth, the Moon and the horizon sensor's FoV based on each projected radius.

The radius, of the Moon as projected on the 3D plane can be calculated as

$$R_{moon} = \|\mathbf{r}_{SE}\| \frac{r_{moon}}{\|\mathbf{r}_{SM}\|}, \tag{5.8}$$

where the parameter r_{moon} is the actual radius of the Moon. The radius of the horizon sensor's FoV can calculated as

$$R_{FoV} = \|\mathbf{r}_{SE}\| \tan(\theta), \tag{5.9}$$

where the angle θ is the FoV angle of the IR horizon sensor. The edges of the Moon and the Earth within the nadir FoV can be determined with these variables defined and calculated.

Firstly the edges of the Moon and the Earth are discretely determined due to the pixel width of the horizon sensor. The discrete points are therefore based on a fixed number of points N for the Earth and the number of discrete points on the Moon is determined based on the ratio of R_{moon} to R_{Earth}. The discrete points for the Moon is therefore equal

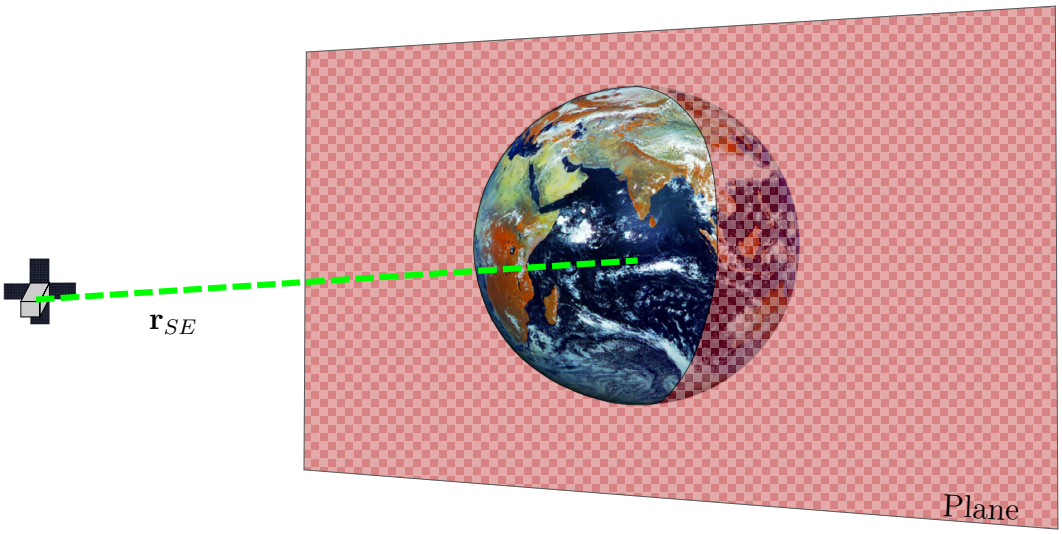

Figure 5.5: Plane perpendicular to $\mathbf{r}_{SE}$ and at centre of the Earth.

to $N\left(\frac{R_{moon}}{R_{Earth}}\right)$. The Earth and the Moon's outline projected as discrete points unto the 3D plane are shown in Figure 5.6. The discrete edges of the Earth and the Moon that is within the FoV of the IR horizon sensor will be used by the algorithm to calculate the centre of the Earth. The discrete points from the Earth must satisfy the following conditions to be used by the algorithm:

1. Distance between a point and the centre of the horizon sensor's FoV must be smaller than R_{FoV}.

2. Distance between a point and the centre of the Moon must be larger than R_{moon}.

The discrete edges of the Moon used for the algorithm must satisfy the following conditions:

1. Distance between any discrete point and the centre of the Earth must be smaller than R_{Earth} for the Moon to overlap the horizon. If any of the discrete points the Moon outline satisfies this condition then conditions 2 and 3 are considered.

2. Distance between a point and the centre of the horizon sensor's FoV must be smaller than R_{FoV}.

3. Distance between a point and the centre of the Earth must be larger than R_{Earth}.

This then creates the array of points that will be used in the algorithm to calculate the centre of Earth.

The scaling of the circles in Figure 5.6 depends on the orbit parameters. The influence of the anomaly will therefore depend on the satellite orbit parameters as this will change the number of points within the horizon sensor's FoV. This in turn will influence the circular fit algorithm.

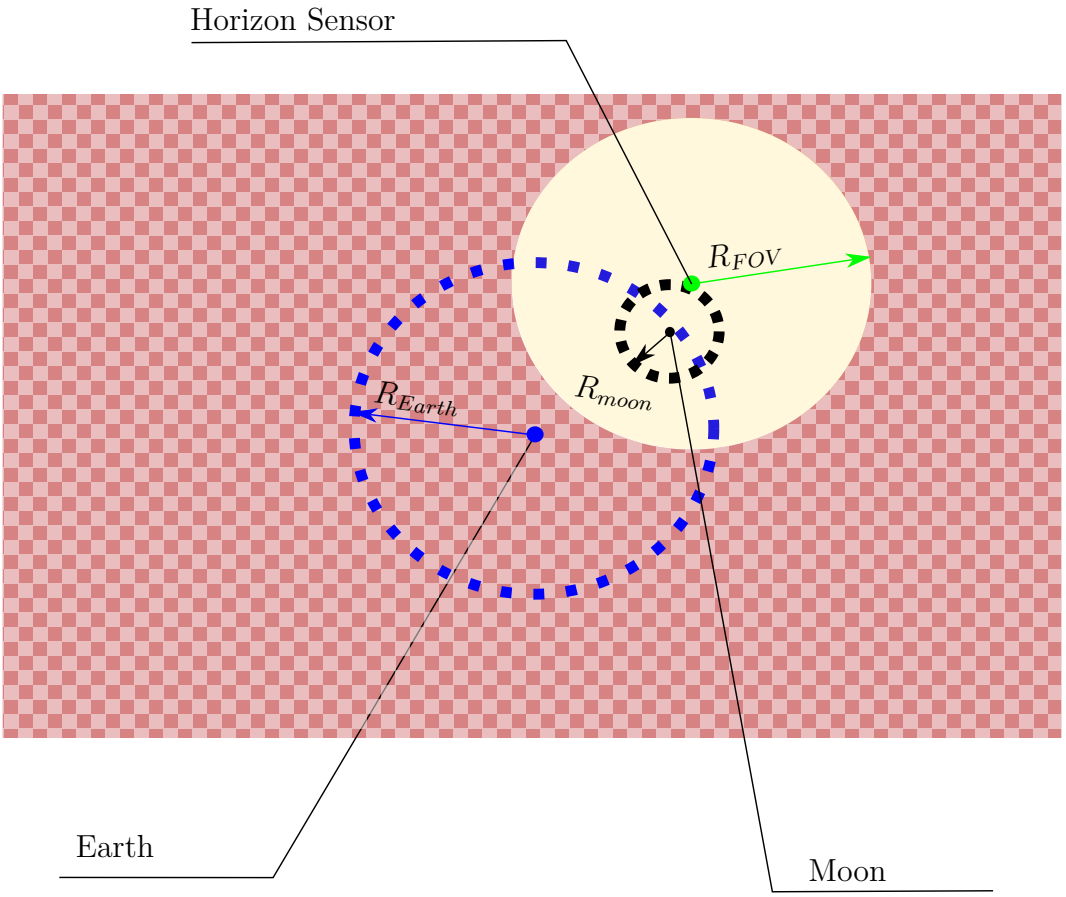

Figure 5.6: Projection of the Moon, the Earth and the horizon sensor's field of view on the geometric plane.

5.2.2. Calculating the Centre of the Earth

The edges of the Earth are detected based on a gradient between the lowest temperature and the highest temperature within the IR horizon sensor's FoV. This will not be implemented in this case, since it is possible to determine discrete points of both the Earth and the Moon from the simulation environment. Furthermore the visible phases of the Moon will not be accounted for. The reasoning for this is due to the coldest side of the Moon being 140K and the warmest part, 400K. The temperature of space is 2.7K and the coldest part on the Earth is 180K. The IR horizon sensor must therefore be calibrated to always use the minimum value for edge detection as 180K or it must use the smallest value in the image, which will most likely be 2.7K. It can therefore be assumed that the Moon will not have any detectable phases for the IR horizon sensor and it will always be seen as a full moon, due to it's lowest temperature being warmer than that of space.

With this assumption the circular fit algorithm as shown in Figure 5.7 can now be used to determine the centre of the Earth on the plane [23]. For this calculation the 3D plane is transformed to a 2D plane and all the coordinates is also transformed. The centre of the Earth on the 2D plane is therefore given as (x_c, y_c). The goal of the algorithm is to

calculate the centre of the Earth (x_c, y_c) and use it to transform the point to the 3D plane and thereafter calculate the measured vector $\mathbf{r}_{SE}$.

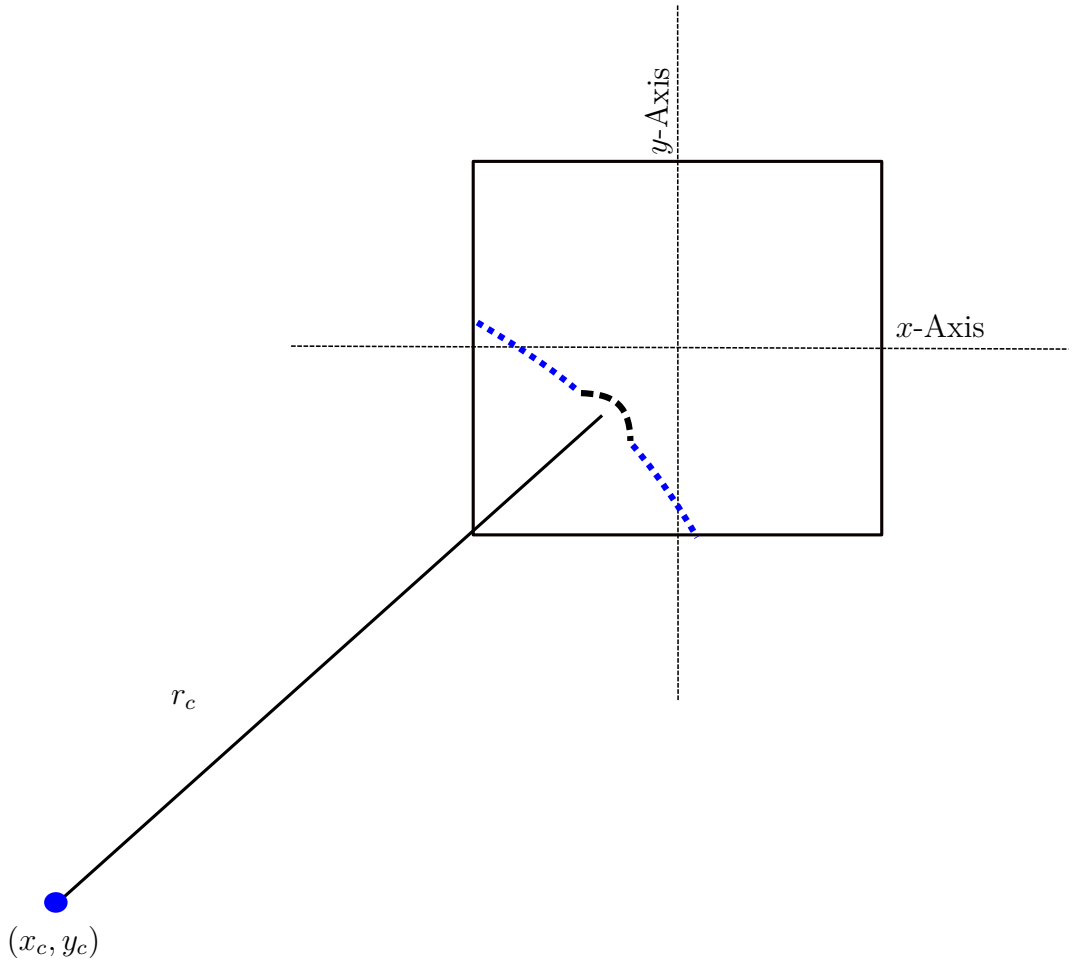

Figure 5.7: Visual representation of the circular fit algorithm for the horizon sensor. The black dotted line represents the discrete dots of the Moon, while the blue dotted line represents the discrete dots of the Earth within the horizon sensor's field of FoV.

Firstly the curvature of a circle is described as

$$ax + by + c = x^2 + y^2, \tag{5.10}$$

where

$$\begin{aligned} a &= 2x_c \\ b &= 2y_c \\ c &= r_c^2 - \sqrt{x_c^2 + y_c^2}. \end{aligned} \tag{5.11}$$

Therefore using all the coordinates of the discrete edges (x_n, y_n) within the horizon sensor's

FoV the values of a, b and c can be calculated as

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x_0^2 + y_0^2 \\ x_1^2 + y_1^2 \\ \vdots \\ x_n^2 + y_n^2 \end{bmatrix}. \tag{5.12}$$

It is thus evident that when the Moon overlaps the horizon of the Earth from the horizon sensor's perspective the centre of Earth will be incorrectly calculated. The error of the calculation depends on the number of discrete points from the moon within the discrete edges (x_n, y_n). A similar anomaly where the Sun is in the FoV of the horizon sensor will not provide a measurement, since the Sun will saturate the IR horizon sensor [23]. The anomaly will therefore not be modelled since it will only provide a sensor vector of $\mathbf{0}$ and will be ignored by default.

5.2.3. Influence of the Anomaly on Estimation

To determine the effect of the discrete the Moon edges on the circular fit algorithm, the measured Earth vector with the Moon on horizon anomaly is shown in Figure 5.8. From Figure 5.8 the effect of the Moon on the horizon anomaly seems to be small and there is no notable change at the sections when the anomaly occurs. The duration in which the Moon is on the Earth's horizon and within the FoV of the horizon sensor is short as is evident by the thin red lines in Figure 5.8. Where the red background indicates the time steps when the anomaly occurs.

Figure 5.9 furthermore provides the estimation metric due to the Moon on horizon anomaly and comparing the estimation metric with that of Figure 4.12 it can be noted that the estimation metric is slightly affected by this anomaly. The effect on the estimation metric is, however, small. This anomaly is therefore not included in the FDIR development. This is because including this anomaly in the FDIR developments might have a negative influence on the detection and isolation algorithms, since the data from this anomaly will be too similar to the normal data might decrease the accuracy of the anomaly detection and isolation. In Figure 5.8 it can be seen that the anomaly's influence falls within the noise profile of the horizon sensor. This anomaly can, however, have a more significant influence with other parameters for the field of view of the horizon sensor and the orbit parameters.

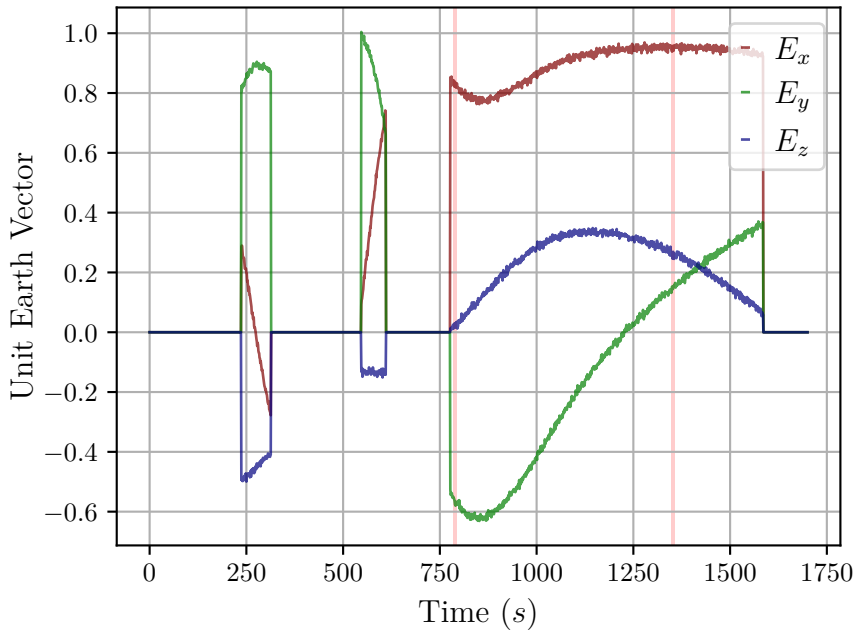

Figure 5.8: The Earth unit vector in SBC due to the Moon on the Earth's horizon anomaly. The red lines of the graph are the periods when the anomaly occurs. The legend provides the axis of the coordinate frame for which the magnitude of the difference in measurement is given.

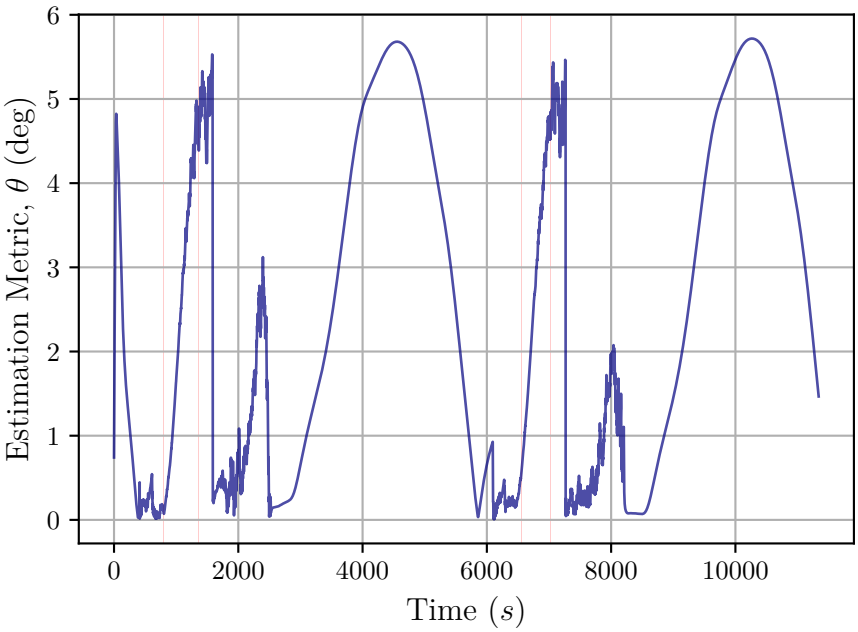

Figure 5.9: Estimation metric in degrees during the first two orbits due to the Moon on the Earth's horizon anomaly. The red lines of the graph are the periods when the anomaly occurs. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

5.3. Magnetic Moment Disturbance

Magnetic moments produced by a coil in solar panels on a CubeSat can create a disturbance torque and influence the magnetometer measurements due to the induced magnetic field in the coil of the solar panel [30, 52]. According to C. Jéger [30] the current I in each individual cell of the solar panel can be modelled as a cumulative current for the entire solar panel, since the normal vector to each cell and the solar panel is the same. This magnetic moment is modelled for the specific size of the CubeSat model in Figure 3.3. The coil in the solar panel and the resulting magnetic field $\mathbf{b}_r$ as well as the resulting dipole moment $\mathbf{m}$ is shown in Figure 5.10. The inner vector area of the coil $\mathbf{a}$ is assumed to be the same as the surface area of the solar panel.

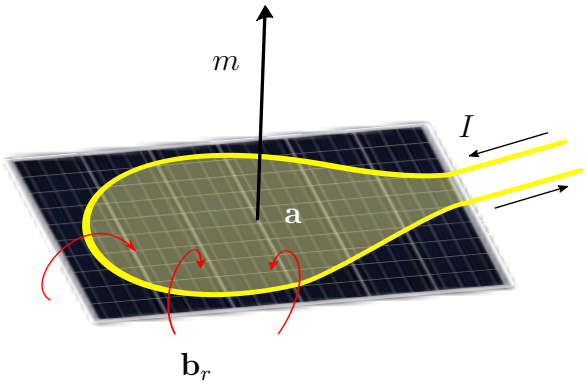

Figure 5.10: Dipole moment due to current in circular loop of circuit in solar panel. The yellow line represents the circuit coil. The dipole moment $\mathbf{m}$ is perpendicular to the surface of the solar panel

The dipole moment $\mathbf{m}$ is calculated as

$$\mathbf{m} = I\mathbf{a}. \tag{5.13}$$

The current is generated by the solar panel and thus depends on the incoming sun vector as well as the area of the solar panel illuminated by the sun. A shadow of the satellite body can cover areas of the solar panels as demonstrated in Figure 5.11. This decreases the current in these solar panels and also the induced dipole moment from these solar panels.

The current I can therefore be calculated as

$$I = I_{max}\frac{A_{total}}{A_{illuminated}}\cos(\theta), \tag{5.14}$$

where θ is the angle between the normal vector to the solar panel and the incoming sun vector $\mathbf{r}_{\text{sun}}$ and I_{max} depends on the solar panel model. I_{max} for this anomaly is given as 0.45A. The areas A_{total} and $A_{illuminated}$ are scalar values and respectively represent the total area of the solar panel and the illuminated area of the solar panel, where the

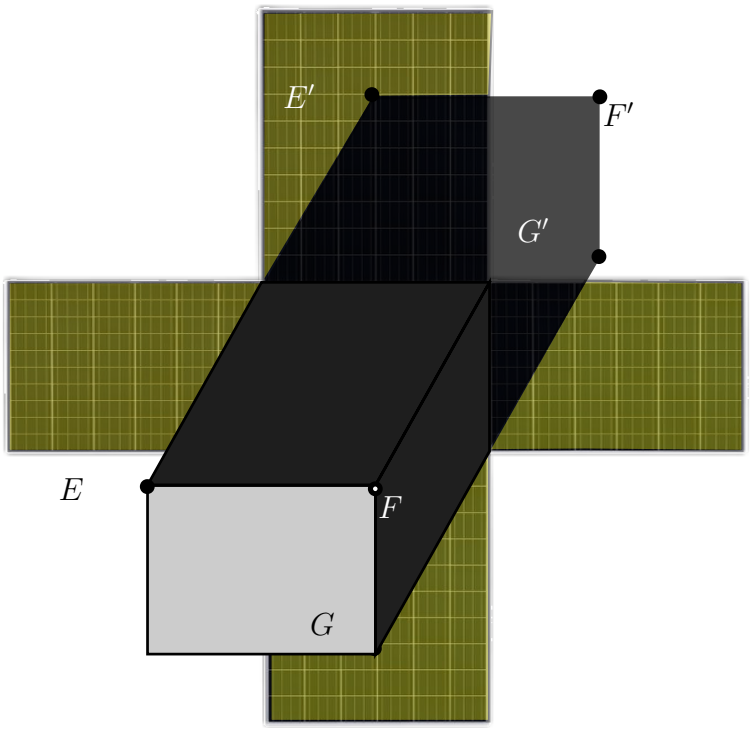

Figure 5.11: Shadow created by CubeSat body on solar panels.

dimensions of the solar panels to calculate A_{total} are provided in Section 3.2.

The dipole moment produces a disturbance torque on the CubeSat. With the resulting torque $\boldsymbol{\tau}_{dm}$, from the dipole moment, expressed as

$$\boldsymbol{\tau}_{dm} = \mathbf{m} \times \mathbf{b}_{\mathcal{B}}, \tag{5.15}$$

where $\mathbf{b}_{\mathcal{B}}$ is the geomagnetic field of the Earth. The only external magnetic field that can create a considerable resulting torque is that of the Earth. The resulting torque for two orbits are shown in Figure 5.12. It is evident that $\boldsymbol{\tau}_{dm}$ is $\mathbf{0}$ during eclipse, since there is no current from the solar panels and therefore no induced dipole moment. The magnetic moment disturbance torque is similar in magnitude to that of the aerodynamic disturbance torque $\boldsymbol{\tau}_{\text{aero}}$ shown in Figure 4.7.

The magnetometer measurement influenced by the magnetic field produced by the coil in the solar panel can be calculated with

$$\mathbf{b}_{r_m} = \frac{\mu_0}{4\pi} \frac{3\mathbf{r}_m(\mathbf{r}_m \cdot \mathbf{m}) - \mathbf{m}}{\|\mathbf{r}_m\|^3}, \tag{5.16}$$

where μ_0 is the vacuum permeability constant and $\mathbf{r}_m$ is the vector from the centre of a solar panel to the centre of the magnetometer. The vector $\mathbf{r}_m$ will therefore change depending on each solar panel. The magnetometer measurement is then a summation of

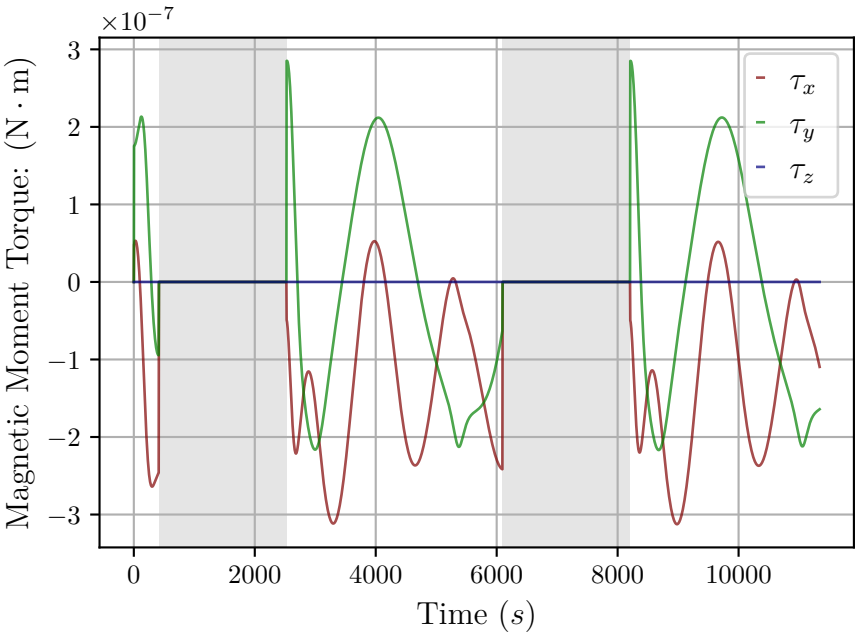

Figure 5.12: Magnetic moment disturbance torque in SBC. The grey background sections of the graph are the eclipse periods, while the sections with the white background is the sunlit phase of the orbit. The legend provides the axis of the coordinate frame for which the magnitude of the torque is given.

each solar panel's resultant magnetic field $\mathbf{b}_{r_m}$ and the geomagnetic field $\mathbf{b}_{\mathcal{B}}$.

5.3.1. Influence of Magnetic Moment Disturbance Anomaly on Estimation

The vector $\mathbf{r}_m$ between the position of the magnetometer and the solar panel influences the magnetic field significantly. The experienced magnetic field by the magnetometer will be different for each solar panel. The resulting measured vector by the magnetometer is the summation of the Earth's geomagnetic field $\mathbf{b}_{\mathcal{B}}$ and the magnetic field produced by the coils in the solar panels $\mathbf{b}_r$. The magnetometer measurement due to the induced dipole moment is shown in Figure 5.13. The changes from the anomalous period to non-anomalous period is significant especially for M_z as is evident in the jumps of the blue line from the white sections to the red sections in Figure 5.13.

From Figure 5.14 it is evident that this anomaly has a significant effect on the estimation, but not as large as the Sun reflection on the sun sensor. The estimation metric increases during the sunlit phase, since this is when the magnetic moment disturbance occurs. However, it is not yet clear whether the estimation error increases due to the difference in the magnetometer measurement or the disturbance torque $\boldsymbol{\tau}_{dm}$. This, however, will be evident when the anomaly recovery results are discussed.

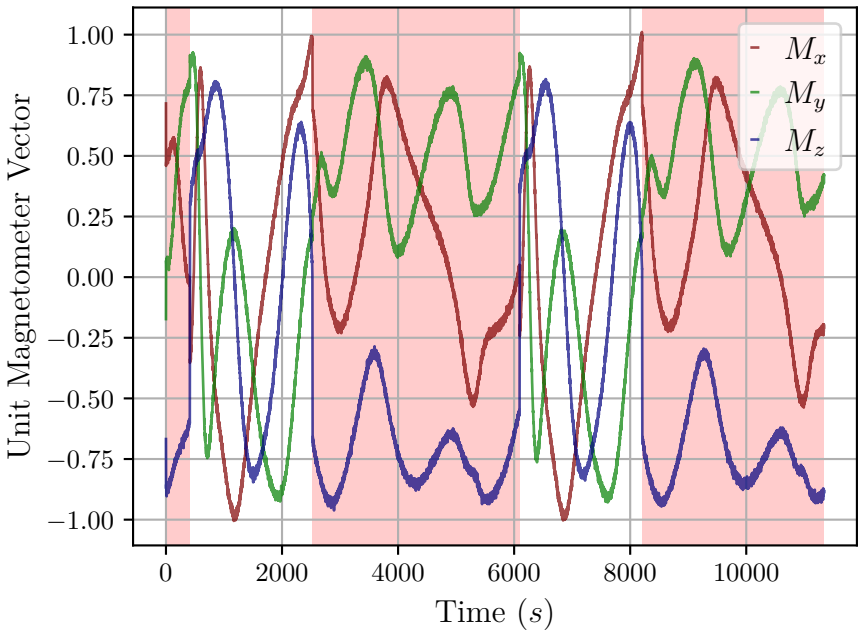

Figure 5.13: Magnetometer unit vector in SBC due to the magnetic moment disturbance. The red background sections of the graph are the periods when the anomaly occurs. The legend provides the axis of the coordinate frame for which the magnitude in the unit vector is given.

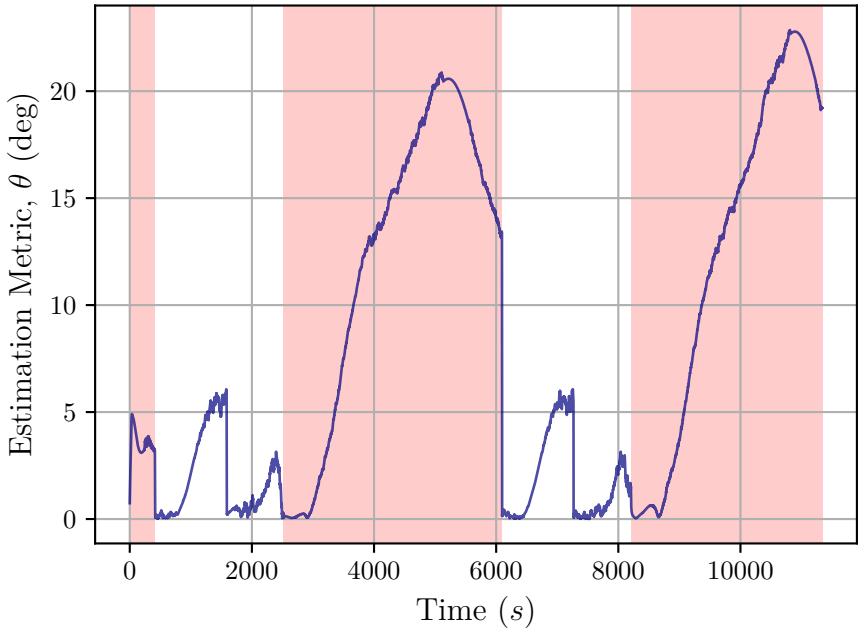

Figure 5.14: Estimation metric in degrees during the first two orbits due to magnetic moment disturbance. The red background sections of the graph are the periods when the anomaly occurs. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

5.4. Higher Noise on Sensor Measurements

As discussed in Chapter 1 and Chapter 2 many research that is conducted in the field of FDIR of sensor anomalies base the research on general sensor anomalies. Increasing the noise of the sensor is a good example thereof. The general anomaly of high noise on a sensor is simulated with an increased standard deviation of 0.1 from the first orbit. An example of the increased noise on the sun sensor is given in Figure 5.15.

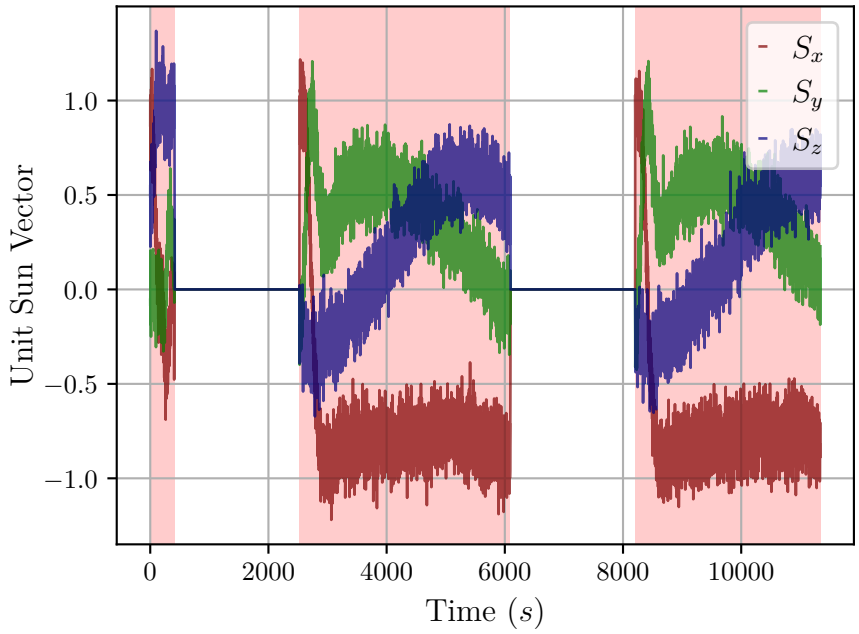

Figure 5.15: Sun sensor measurement with high noise as anomaly. The red background sections of the graph are the periods when the anomaly occurs. The legend provides the axis of the coordinate frame for which the magnitude in the unit vector is given.

To demonstrate the effect of the increase noise on the sun sensor on the estimation metric, the first two orbits are provided in Figure 5.16, where the red background is when the sun sensor is experiencing the anomaly. It is therefore evident that this is anomaly influences the estimation enough to require recovery and must therefore be dwelt with. The fault detection of general anomalies such as high noise can be done through waveform analysis methods such as the Fourier transform. By determining the magnitude of the various harmonics and frequencies within the data set, it can be determined whether the noise has drastically increased. This, however, is not analysed as a possible detection method, since the focus of the thesis is the implementation of learning methods for anomaly detection and isolation as discussed in Chapter 2.

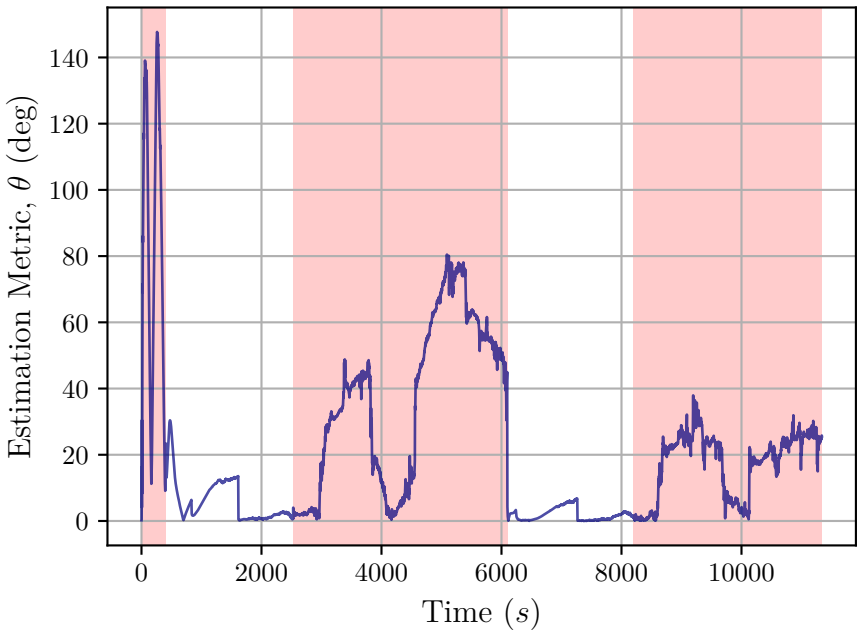

Figure 5.16: Estimation metric for the first two orbits due to high sun sensor noise without any recovery. The red background sections of the graph are the periods when the anomaly occurs. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

5.4.1. Reaction Wheel Failure

Failures of actuators may effect both the estimation and the control of the satellite. When an actuator fails it therefore influences all the sensor measurements. The anomaly will be modelled as a sudden failure in the actuator when it does not react to inputs. The reaction wheel will thus continue to spin, but the control input of the failed reaction wheel will be equal to 0. The momentum dumping will decrease the failed reaction wheel's momentum over time. This will influence the EKF, since the model update will be inaccurate. This anomaly is therefore included even though it is not a sensor anomaly, because this anomaly often occurs and influences the estimation accuracy.

The recovery of the control torque $\boldsymbol{\tau}_w$ within the controller is, however, not within the scope of this thesis, but the model update for the estimation will be adjusted based on a modified torque vector $\boldsymbol{\tau}'_w$. The torque of the failed reaction wheel is set to 0 for a possible recovery method on only the EKF. The anomaly is implemented from the first orbit. The resulting estimation metric for this anomaly is shown in Figure 5.17 and it is evident that this anomaly has a large negative effect on the EKF.

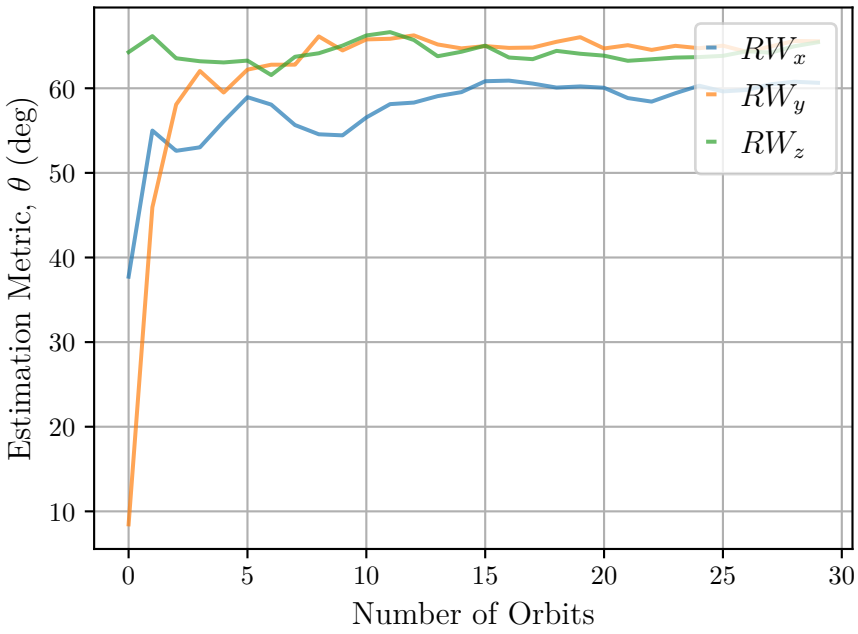

Figure 5.17: Average estimation metric per orbit for 30 orbits of reaction wheel failure without any recovery. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

5.5. Summary

This chapter provides anomalies for both orbit sensitive anomalies (practical) and general anomalies. An anomaly can be classified based on the influence that it has on either the sensors, the actuators or both the sensors and the actuators. The orbit sensitive anomalies are modelled and discussed since it might be more difficult to classify than general anomalies and many literature studies only focusses on FDIR for general sensor anomalies.

It is evident that the reflection of solar panels unto the sun sensor, reaction wheel failure and the magnetic moment disturbance have considerable increases in the estimation metric. These anomalies must therefore be recovered from, to ensure autonomous fault tolerant control. The Moon in the FoV of the horizon sensor has a much smaller effect on the estimation metric given the current conceptual design of the satellite. A FDIR technique is required to ensure that these anomalies do not negatively effect the estimation of the EKF. The anomaly should be detected and then classified based on the sensor responsible for the anomaly (isolation). An appropriate action can be taken to minimise the anomaly's effect on the mission's pointing performance. This chapter therefore provides the detail of the anomaly block in Figure 1.2.

Chapter 6

Anomaly Detection and Isolation

Detecting an anomaly is a binary classification problem. The method implemented must be able to distinguish between normal data and anomalies. There are many different classes of algorithms which can be implemented to detect anomalies. This ranges from simple *If Statement* logic to neural networks. Learning methods are very popular within the field of FDIR and this is consequently chosen as the method of implementation. This is chosen especially due to the lack of testing of developed learning methods for FDIR on practical satellite sensor anomalies. Although there are various learning methods which can be implemented for this use case, only three supervised learning methodologies and one unsupervised learning method are implemented. These methods are chosen due to the literature study as discussed in Chapter 2.

The major difference between the supervised learning and the unsupervised learning methods for anomaly detection is that the data must be labelled for supervised learning. Moreover, the number of anomalous samples and normal samples should be more or less the same for supervised learning, while in the case of unsupervised learning, the anomalous samples should be sparse. This will be accounted for and will be discussed in the application of the supervised learning algorithms.

The feature extraction methods are also discussed in this chapter. The feature extraction methods are implemented to enhance the prediction accuracy of the detection and isolation methods. As shown in Figure 1.2, the feature extraction method provides additional inputs to the detection and isolation algorithms. This chapter will discuss the feature extraction block, as well as the detection and isolation block in Figure 1.2.

The methods discussed in this chapter is shown in Figure 6.1. They are grouped according to the functionality of the method and are discussed, starting with the moving average of the estimated sensor measurements (MAESM) and ending with support vector machines (SVMs). The feature extraction methods discussed in this chapter are the MAESM and the local outlier factor (LOF) algorithms. The LOF can also be implemented as an unsupervised learning detection algorithm and is therefore classified as both a feature extraction and a detection method. The detection methods are divided into supervised learning and unsupervised learning methods. The unsupervised learning method LOF can only be implemented for detection (binary classification), while the supervised learning methods like decision tree (DT), random forest (RF) and support vector machines (SVMs)

can also be implemented for isolation (classifying the sensor responsible for the anomaly).

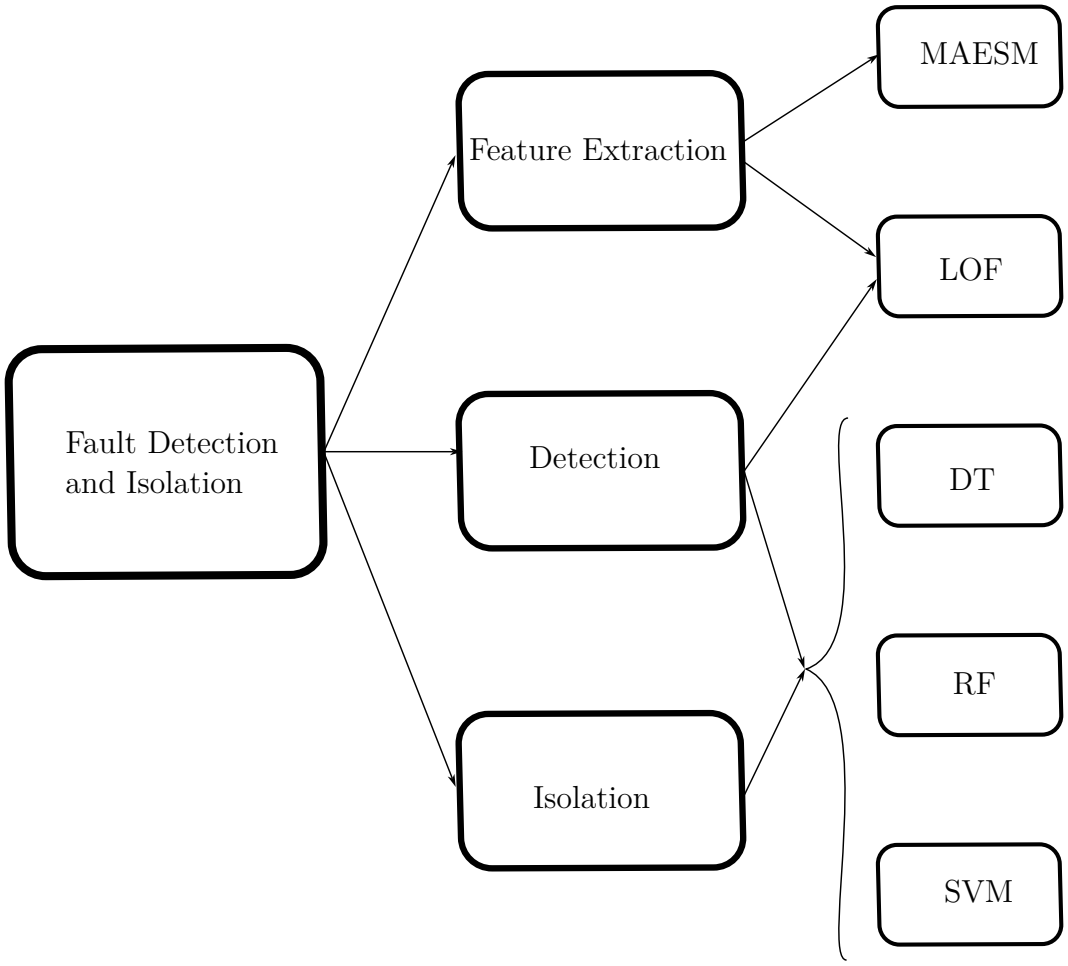

Figure 6.1: The grouping of the various algorithms and methods into the subgroups of feature extraction, detection and isolation. The abbreviations of the methods are provided in the figure. The methods in the figure are the moving average of the estimated sensor measurements (MAESM), local outlier factor (LOF), decision tree (DT) and random forest (RF).

Based on the discussions in Chapter 5 the algorithms discussed here will be analysed on practical anomalies. This is since these anomalies are not the general anomalies and are specific to a satellite's orbit. These anomalies are difficult to classify and since other research papers do not test the proposed methods on these type of anomalies, these anomalies will be the focus of the analysis in this chapter.

6.1. Moving Average of the Estimated Sensor Measurements

The moving average of the estimated sensor measurements (MAESM) method is adapted from a method proposed by de Silva et al. [21]. The method proposed by de Silva et al. [21] implements Dynamic Mode Decomposition (DMD), to reduce the dimensionality of

the data set. The method aims to develop a correlation between the sensor measurements at time step k and the sensor measurements at time step $k+1$. This can then be used to predict the next sensor measurement and detect anomalies based on the difference between the predicted sensor measurements and the actual sensor measurements. The method was tested on various different systems by de Silva et al. [21] governed by underlying physics and the results thereof is promising as it increases the prediction accuracy of a decision tree for detecting anomalies of many systems.

MAESM is implemented with slight variations of the method proposed by de Silva et al. [21]. These variations are discussed in the next sections and the reasoning for the changes are also discussed.

6.1.1. Theoretical Background

Linear regression is a method of predicting the value of one variable based on the value of another variable. Linear regression is implemented to predict the next time step's $k+1$ measurements from the current time step's k measurements. This is a slight adaptation from work done by de Silva et al. [21].

The proposed method by de Silva et al. [21] uses DMD, to provide an estimation of the next time step sensor vector based on the current measurement of the sensor as well as the measurements of the other sensors in the system. This is implemented to detect when the estimated sensor vector deviates a lot from the actual sensor measurement. DMD constructs a matrix $\mathbf{A}$ to relate the sensor vector $\mathbf{x}$ with the following time step of the sensor vector $\mathbf{x}_{k+1}$. The next sensor vector can be calculated as

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k, \tag{6.1}$$

where $\mathbf{x}_k$ during a specified number n of time steps will be denoted as

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{n-1} \end{bmatrix}^{\mathrm{T}} \tag{6.2}$$

and similarly $\mathbf{x}_{k+1}$ during a specified number n of time steps will be denoted as

$$\mathbf{X}' = \begin{bmatrix} x_2 & x_3 & \cdots & x_n \end{bmatrix}^{\mathrm{T}}. \tag{6.3}$$

The method of DMD is useful for high order systems where the calculation of $\mathbf{A}$ is computationally intensive. The main aspect of DMD is that it reduces the dimensionality of the data. This reduction in the dimensionality is not required for the data from the sensor measurements of the satellite simulation, and using DMD is not justifiable. Consequently, a linear regression model is implemented.

The pseudo-inverse of $\mathbf{X}'$, denoted as $\mathbf{X}^{\dagger}$, can be used to approximate the matrix $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{X}\mathbf{X}^{\dagger}. \tag{6.4}$$

The matrix $\mathbf{A}$ is implemented to estimate the next time step sensor vector $\mathbf{x}_{k+1}$ from the current sensor vector $\mathbf{x}_k$. This, however, does not include the other sensor measurements, the control torques or any other variable that influences the prediction of the next time step sensor vector.

de Silva et al. [21] includes the matrix $\mathbf{B}$ and the matrix of the sensor measurements $\mathbf{Y}$ to relate the vector measurements of the other sensors to adjust the predicted state $\mathbf{X}'$ of the monitored sensor

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{Y}, \tag{6.5}$$

Both $\mathbf{A}$ and $\mathbf{B}$ can be calculated as

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix} = \mathbf{X}' \begin{bmatrix}\mathbf{X} \\ \mathbf{Y}\end{bmatrix}^{\dagger} \tag{6.6}$$

After creating the database from the simulation environment during normal operation, the matrices $\mathbf{A}$ and $\mathbf{B}$ can be calculated.

The matrix $\mathbf{Y}$ is adjusted for this use case, where the matrix $\mathbf{Y}$ is the control torques for the magnetorquers and reaction wheels and the matrix $\mathbf{X}$ is all of the sensor measurements. Both matrices are for the combination of the respective variables for n number of time steps. The model of Equation 6.5 denotes the prediction of the sensor measurements at time step $k+1$ based on the current sensor measurements and control inputs.

After calculating both matrix $\mathbf{A}$ and $\mathbf{B}$ from the simulation environment, the model for predicting the matrix of sensor measurements $\mathbf{X}_{k+1}$ at the time step $k+1$ can be implemented. The matrix $\mathbf{X}_k$ is defined as the matrix of the sensor measurements of the sun sensor $\mathbf{v}_{\mathcal{B},s}$, magnetometer $\mathbf{v}_{\mathcal{B},m}$ and the horizon sensor $\mathbf{v}_{\mathcal{B},h}$ at a given time step k and is calculated as

$$\mathbf{X}_k = \begin{bmatrix}\mathbf{v}_{\mathcal{B},s} & \mathbf{v}_{\mathcal{B},m} & \mathbf{v}_{\mathcal{B},h}\end{bmatrix}. \tag{6.7}$$

The matrix $\mathbf{Y}_k$ is therefore also the control torquers of the magnetorquers $\boldsymbol{\tau}_m$ and the reaction wheels $\boldsymbol{\tau}_w$ at a given time step k

$$\mathbf{Y}_k = \begin{bmatrix}\boldsymbol{\tau}_w & \boldsymbol{\tau}_m\end{bmatrix}. \tag{6.8}$$

As implemented by de Silva et al. [21] the model for predicting the matrix of sensor measurements $\mathbf{X}_{k+1}$ is adjusted with a Kalman filter. The predicted sensor measurement matrix at a time step k with be denoted as $\hat{\mathbf{X}}_k$ and will be referred to as the estimated sensor measurement matrix. From the matrix $\mathbf{A}$ and the matrix $\mathbf{B}$ the Kalman filter can

be implemented to estimate the next time step measurement matrix $\hat{\mathbf{X}}_{k+1}$ as

$$\hat{\mathbf{X}}_{k+1} \approx \mathbf{A}\hat{\mathbf{X}}_k + \mathbf{B}\mathbf{Y}_k + K(\mathbf{X}_k - \hat{\mathbf{X}}_k), \tag{6.9}$$

where the Kalman gain K is set to 0.001 [21]. After the calculation of the estimated state matrix $\hat{\mathbf{X}}_{k+1}$ de Silva et al. [21] proposes a moving average of the innovation covariance

$$\mathbf{V}_k = \frac{1}{N} \sum_{i=k-N}^{k} (\mathbf{X}_i - \hat{\mathbf{X}}_i)(\mathbf{X}_i - \hat{\mathbf{X}}_i)^{\mathrm{T}}, \tag{6.10}$$

where N is the number of time steps to account for. This is implemented to determine how much the estimated sensor measurements deviate from the actual sensor measurements. The moving average of the estimate sensor measurements $\mathbf{V}_k$ will therefore be the feature extracted from the sensors and be the additional input parameter to the isolation algorithms.

6.1.2. Application for Feature Extraction

In order to implement the MAESM for feature extraction, the aim must be to provide additional features to the detection and isolation algorithms which will enhance the prediction accuracy. The method is implemented by providing the sensor measurements as an input to the MAESM algorithm as shown in Figure 6.2. This also correlates with the Figure 1.2, where the sensor measurements are given as input to the feature extraction block.

The matrices $\mathbf{A}$ and $\mathbf{B}$ in Equation 6.9 are determined after executing 30 orbits of a normal run of the simulation (without any anomalies). The sensor data as matrix $\mathbf{X}$ and the control torques as matrix $\mathbf{Y}$ after running the simulation can be used to calculate matrices $\mathbf{A}$ and $\mathbf{B}$ according to Equation 6.6. Matrices $\mathbf{A}$ and $\mathbf{B}$ can therefore be stored as arrays and be collected from memory during the execution of the simulation when the MAESM algorithm is implemented as the feature extraction method.

During the execution of the simulation, Equation 6.9 and Equation 6.10 are implemented to determine $\mathbf{V}_k$ from the current sensor measurements as the matrix $\mathbf{X}_k$ and the estimated sensor measurements $\hat{\mathbf{X}}_k$. The matrices $\mathbf{X}_k$ and $\hat{\mathbf{X}}_k$ are stored for N number of time steps and the moving average is therefore recalculated every time step k. The matrices $\mathbf{X}_k$ and $\hat{\mathbf{X}}_k$ are initiated as $\mathbf{0}$ matrices (where all the values are equal to 0). And the moving average $\mathbf{V}_k$ is therefore not influenced by the time steps before time step 0.

6.1.3. Analysis for Feature Extraction

The moving average of the estimated sensor measurements provided by the linear regression method is a matrix and is therefore difficult to visualize. To determine whether the method

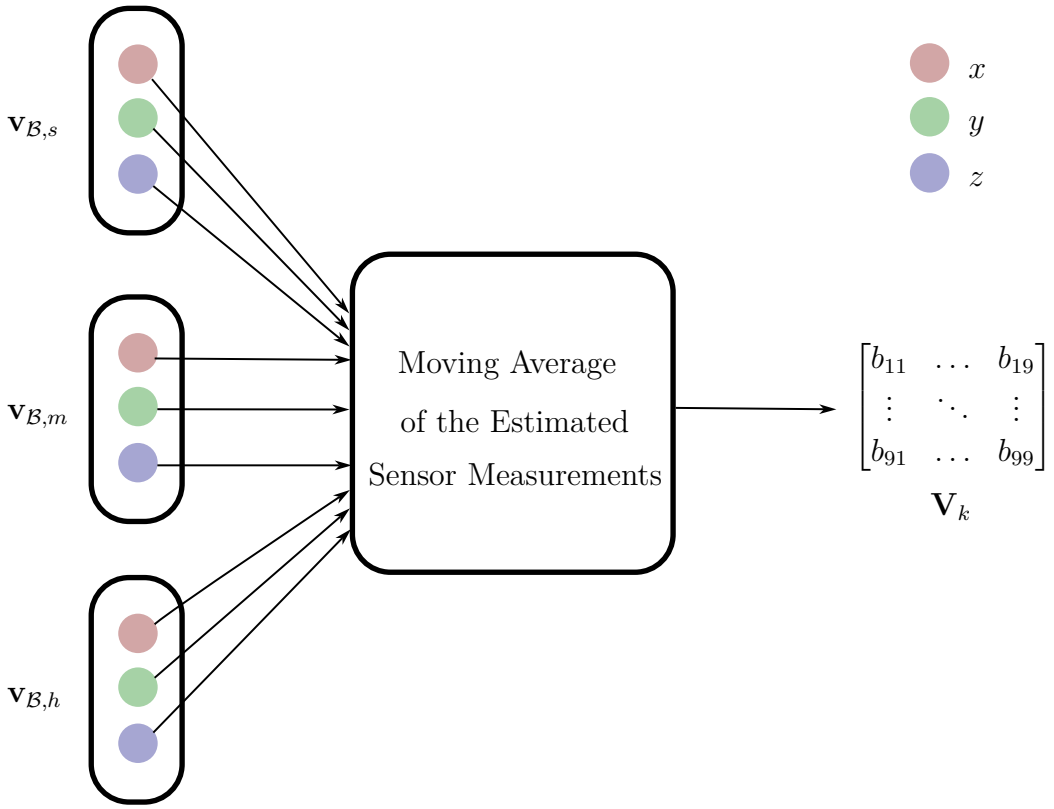

Figure 6.2: The input and output of the moving average of the estimated sensor measurements method. Where the input is the sensor measurements and the output is the moving average $\mathbf{V}_k$.

provides a feature that provides information of whether the current time step contains an anomaly cannot be done by plotting the matrix. The moving average can, however, be simplified to a summation of the values in the moving average (SVMA) or the absolute values thereof (SAVMA). This is implemented to demonstrate the increase of the values within the matrix after an anomaly occurs. Since the moving average has, by definition, a delayed response to the anomaly, it is expected that the SVMA and SAVMA will increase after an anomaly occurs. The SAVMA, however, will only provide insight into whether the estimated sensor measurements deviates from the actual sensor measurements during an anomaly. The SVMA and the SAVMA for the Sun reflection is provided in Figure 6.3 and Figure 6.4 respectively. In the case of the Sun reflection, the SAVMA increases after an anomaly occurs with a delayed response. The SVMA provides even more insight as it creates spikes after an anomaly occurred and then returns to normal.

In Figure 6.5 and Figure 6.6, the SVMA and SAVMA seems to be smaller during a normal period than a period when an anomaly occurs. The moving average of the estimated sensor measurements provides the change in the relationship between the control input and the sensors. Figure 6.6 provides the insight that the magnetic moment disturbance has a smaller effect on the relationship between the sensor and the control input when compared to that of the Sun reflection.

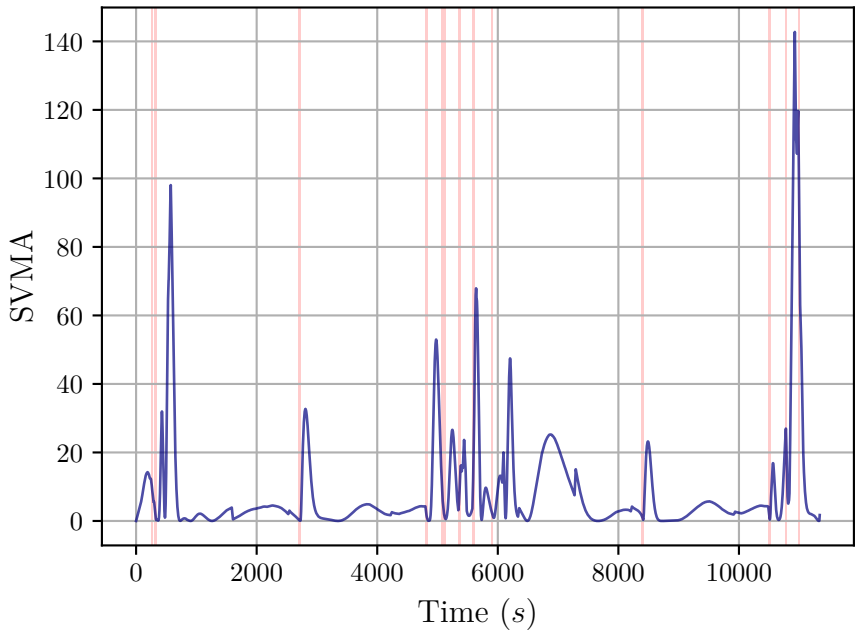

Figure 6.3: Summation of absolute values in the moving average during first two orbits during the Sun reflection anomaly. The red lines indicates when the Sun reflection anomaly occurs.

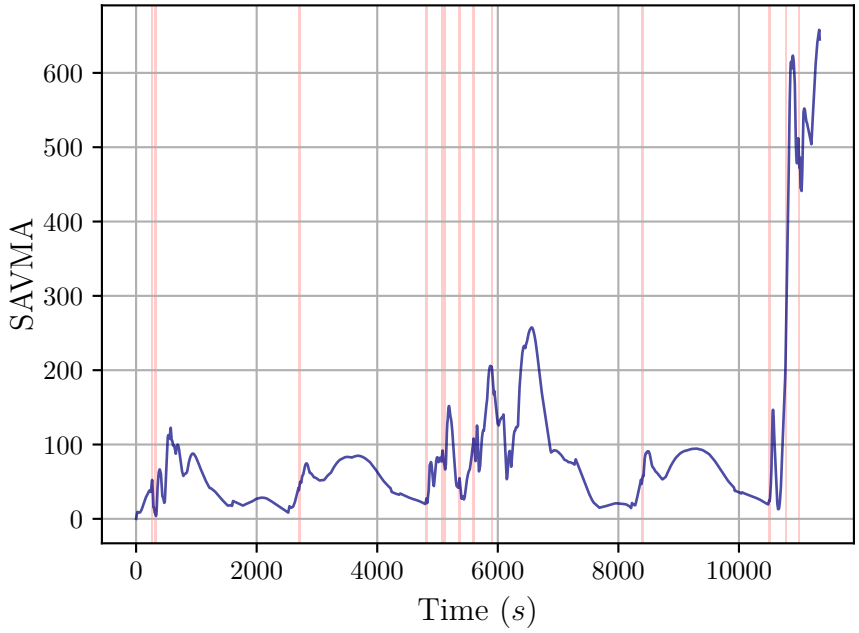

Figure 6.4: Summation of absolute values in the moving average during first two orbits during the Sun reflection anomaly. The red lines indicates when the Sun reflection anomaly occurs.

It can therefore be concluded that the moving average of the estimated sensor measurements might not provide insight into whether a magnetic moment disturbance anomaly occurred. The delayed response in both Figure 6.4 and Figure 6.6 seem to suggest that

the moving average of the estimated sensor measurements might also trigger a detection algorithm too late.

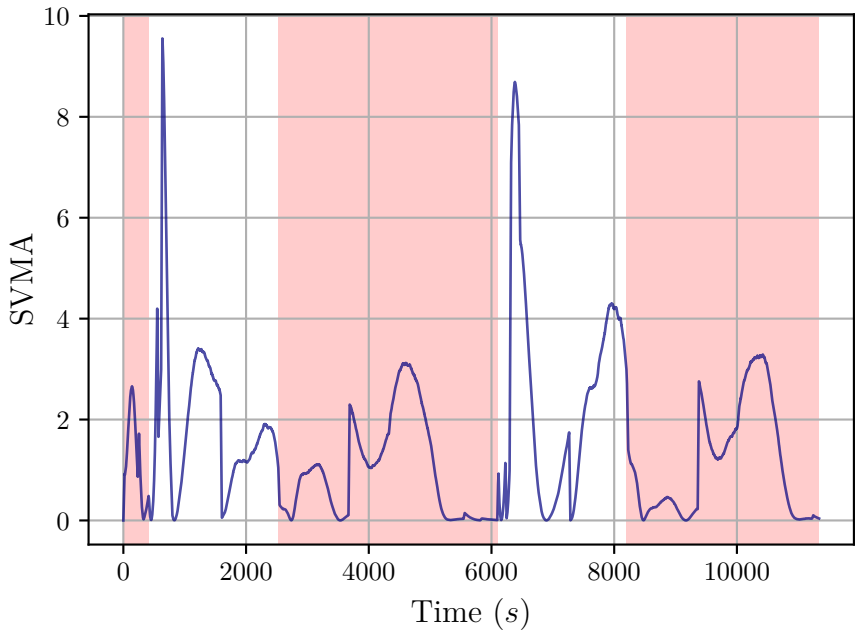

Figure 6.5: Summation of absolute values in the moving average during first two orbits during the magnetic moment disturbance anomaly. The red area indicates when the magnetic moment disturbance occurs.

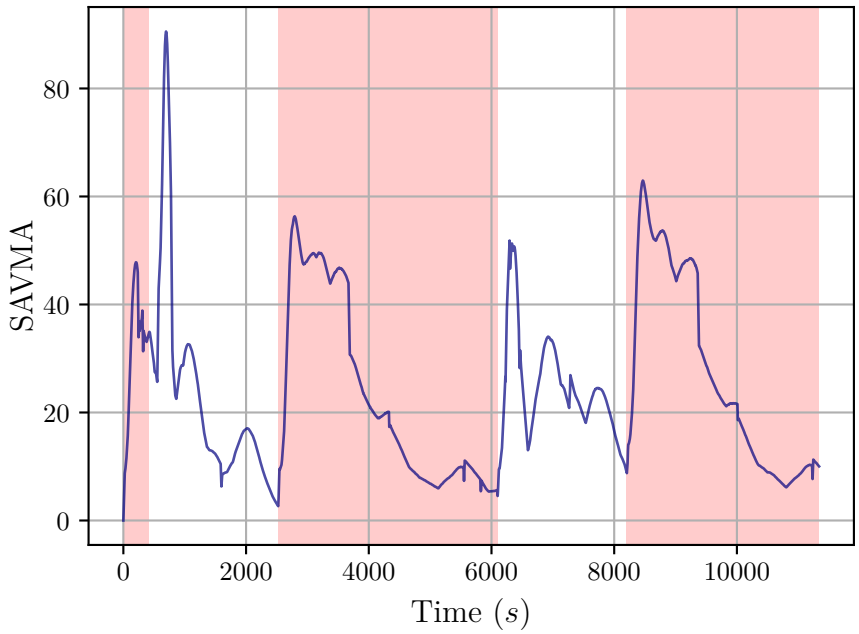

Figure 6.6: Summation of absolute values in the moving average during first two orbits during the magnetic moment disturbance anomaly. The red area indicates when the magnetic moment disturbance occurs.

6.2. Local Outlier Factor Algorithm

Most anomaly detection algorithms are based on a metric which accounts for the entire data set [53]. Many anomalies are, however, identifiable in relation to the local neighbourhood of data points and not the overall data set. Breunig et al. [53] therefore developed the local outlier factor (LOF) algorithm that provides a measure of a data point's "irregularity" within a subset of data points. This implies that a data point is not merely classified as an anomaly or not, but that a local outlier factor is calculated to determine how much a data point is distantiated from its k nearest neighbours. This is a very attractive method for detecting outliers on satellite sensor anomalies, since the satellite data can be classified into two major neighbourhoods, namely the eclipse and sunlit phase. It is therefore necessary to determine whether a data point is an anomaly within the local neighbourhoods where the behaviour of the satellite is different in each instance. The advantage of this algorithm is that the algorithm is only trained on simulated data without any anomalies and can thereafter provide an outlier score of a data point based on the training data. This leads to a method that can potentially detect multiple anomalies without any further training after training on the normal data.

6.2.1. Theoretical Background

Breunig et al. [53] developed the LOF algorithm to provide an outlier score for a data point relative to its k nearest neighbours. The outlier score O as the output of the LOF algorithm is calculated as

$$O_{N_{min}}(p_o) = \frac{\sum_{p_i \in N_{min}(p_o)} \frac{\rho_{N_{min}}(p_i)}{\rho_{N_{min}}(p_o)}}{|N_{min}(p_o)|}, \tag{6.11}$$

and a summary for the derivation thereof will be given in the next few paragraphs. The outlier score for a random data set is demonstrated in Figure 6.7 where the data points that are clustered together have smaller outlier scores compared to the data points that are further away from the highly dense areas.

The LOF algorithm requires an understanding of the underlying concepts and definitions. These definitions are provided by Breunig et al. [53]. Firstly, the k-distance of an object is calculated. This can be demonstrated in Figure 6.8, where, given $k = 5$, the k-distance of the object p_o is provided as the radius of the dashed circle. The k-distance can be determined after a k number of objects are within a given radius from the object p_o. The k-distance is, therefore, simply the distance from the current object to the k^{th} object from it. The k-distance of an object p_o will be denoted as k-distance(p_o).

To reduce fluctuations in the distance $d(p_o, p_i)$ between the object p_o and an object p_i

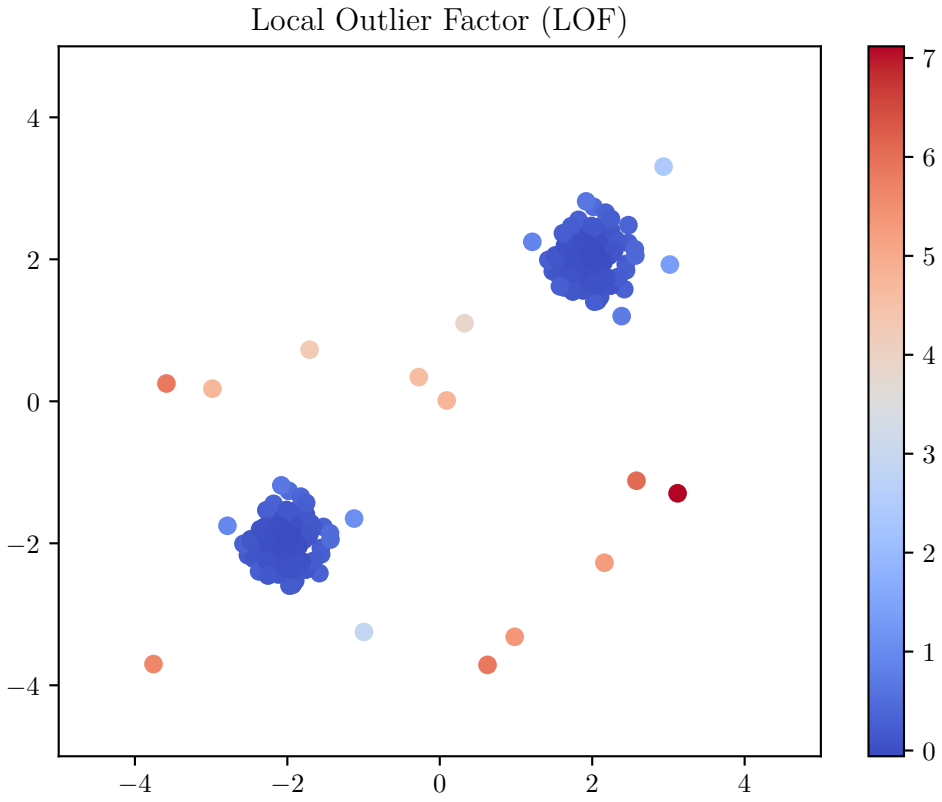

Figure 6.7: Local outlier factor of a random data set to demonstrate the outlier score produced by the local outlier factor algorithm [54]. The colour map on the right provides the value of the outlier score O.

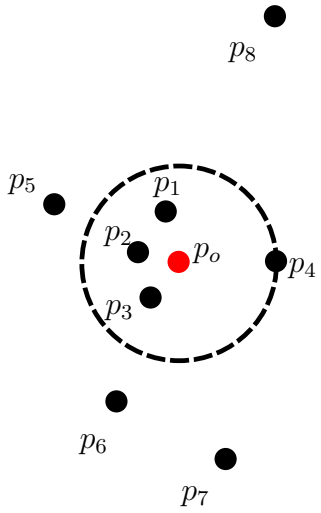

Figure 6.8: Reachability distance of the object p_o with $k = 5$.

the distance between p_o and p_i is replaced with the reachability distance [53]

$$d_k = \mathbf{max}\left(d(p_o, p_i), k\text{-distance}(p_o)\right). \tag{6.12}$$

The reachability distance d_k between the object p_o and the objects p_1 to p_4 are equal given that $k = 5$. The reachability distance d_k between the object p_o and the objects p_5 to p_8 are the distance from p_o to the respective object p_i since it is larger than the k-distance from the object p_o.

The parameter determining the number of points required to define a local neighbourhood is denoted as N_{min}. The k-distance will therefore be replaced by the N_{min}-distance to determine the local neighbourhood. The parameter N_{min} is fine tuned based on the specific application and Breunig et al. [53] provides a method of determining N_{min} for the most accurate results. The equations and parameters that are influenced by this parameter is given the subscript of N_{min}.

To determine the local outlier factor the density of the local neighbourhood is required. The inverse of the average reachability distance $d_{N_{min}}$ for N_{min}-nearest data points to the data point p_o [53] is the local reachability density

$$\rho_{N_{min}}(p_o) = 1/\left(\frac{\sum_{p_i \in N_{min}(p_o)} d_{N_{min}}(p_o, p_i)}{|N_{min}(p_o)|} \right). \tag{6.13}$$

This local reachability density $\rho_{N_{min}}(p_o)$ enables the calculation for the outlier score of point p_o in Equation 6.11.

The threshold for detecting an outlier is not fixed and the threshold can be changed depending on the application. The rule of thumb for detecting an outlier is that when the outlier score O is larger than 1, the point is considered an outlier with respect to its neighbourhood. LOF can therefore be an anomaly detection algorithm if a given threshold is implemented, otherwise the O can be given as an additional feature for other anomaly detection algorithms.

6.2.2. Application for Feature Extraction

The LOF algorithm is aimed at producing a measure of the how much a given data point is considered an outsider or anomaly within a local neighbourhood and not for all the data points. This measure of "irregularity" is given as the outlier score O from Equation 6.11. The implementation of the LOF algorithm for feature extraction is similar to that of MAESM. The LOF algorithm is trained on normal data with the Scikit-learn model for Python [54]. The fitted model produced by the algorithm is stored in memory and used during simulation to provide an outlier score for a data sample at any given time step k. This outlier score is then provided to a detection or isolation algorithm as an additional input parameter.

6.2.3. Analysis for Feature Extraction

The LOF algorithm is trained on the normal operation of a satellite simulation (without any anomalies) using the simulation described in Chapter 4. Thereafter, the outlier score O for both the Sun reflection anomaly, as well as the magnetic moment disturbance as modelled in Chapter 5 is provided in Figure 6.9 and Figure 6.10, respectively. The time steps when the anomaly occurs are given a red vertical line.

In Figure 6.9, the outlier score O increases after the Sun reflection anomaly occurs. This is due to effect of the anomaly on the ADCS. The ADCS control is anomalous for a duration after the anomaly occurred.

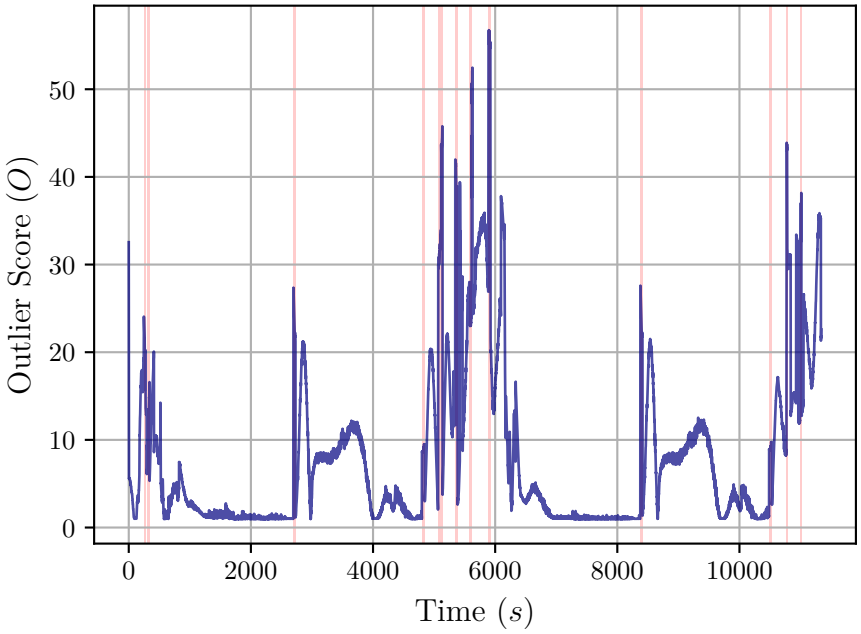

Figure 6.9: Local outlier factor during the first two orbits of the Sun reflection anomaly. The red lines indicate when the Sun reflection anomaly occurs.

In Figure 6.10, the same pattern as Figure 6.9 occurs, where the stable value of the LOF without any anomalies is 0. Figure 6.10 demonstrates this, since the value significantly increases after an anomaly occurs and decreases to a value of 0 after the anomaly period is over. From both Figure 6.9 and Figure 6.10 it is clear that an anomaly will be detected shortly after the anomaly occurs, if using the threshold for outlier score O of 1. The outlier score O does, however, decrease during the magnetic moment disturbance anomaly. The only purpose of the feature extraction method is to provide a feature that increases the information gain of whether a data sample is an anomaly or not and it is clear that the outlier score O does this.

It is important to note that the LOF algorithm provides a severity of the anomaly. This is evident in that the outlier score O for the Sun reflection reaches a maximum of 55, while the maximum for the magnetic moment disturbance is 33. This is as expected

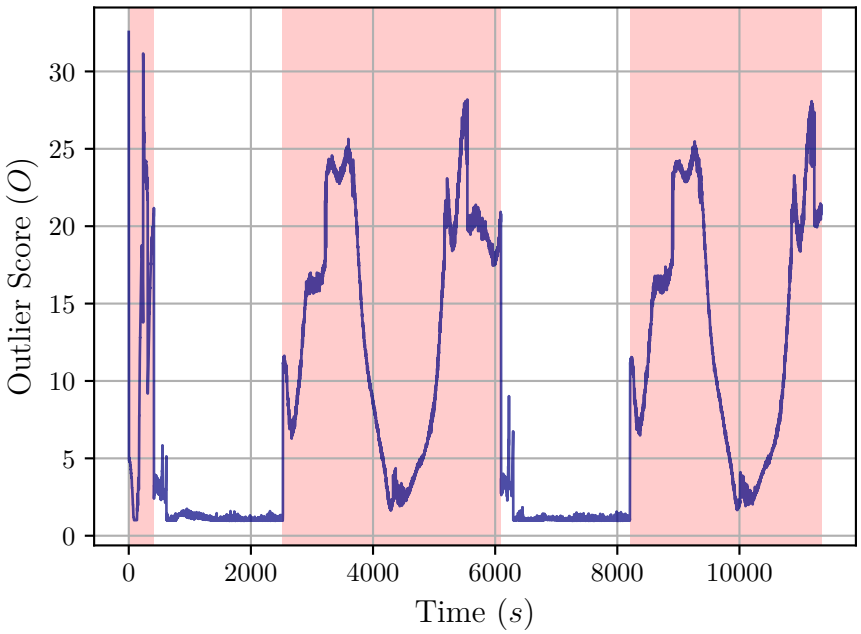

Figure 6.10: Local outlier factor during first two orbits of the magnetic moment disturbance anomaly. The red area indicates when the magnetic moment disturbance occurs.

when comparing the estimation metric during the Sun reflection and the estimation metric during the magnetic moment disturbance in Figure 5.3 and Figure 5.14.

6.2.4. Application for Anomaly Detection

The LOF algorithm can be implemented to detect anomalies through determining a threshold for when an outlier score is considered an anomaly. When the outlier score of a data sample is above the given threshold, the data sample is classified as an anomaly, and otherwise classified as normal. During the implementation of LOF for anomaly detection, the threshold value is kept as the suggested threshold of 1 [53]. The implementation of LOF for anomaly detection is demonstrated in Figure 6.11. The LOF algorithm is the exact same trained model as discussed in Section 6.2.2. The only difference is that the outlier score (as the output of the trained model) is not used as an additional input to another detection algorithm, but is used in isolation as the sole means of detecting an anomaly using a basic *If Statement* to determine whether the outlier score is larger than 1.

6.2.5. Analysis for Anomaly Detection

In order to determine the performance of the LOF algorithm on the detection of anomalies, a few key metrics are required. These metrics will be consistent throughout the analysis of all the detection methods. The metrics will be a confusion matrix as well as a plot of the average accuracy per orbit of the classification during 30 orbits.

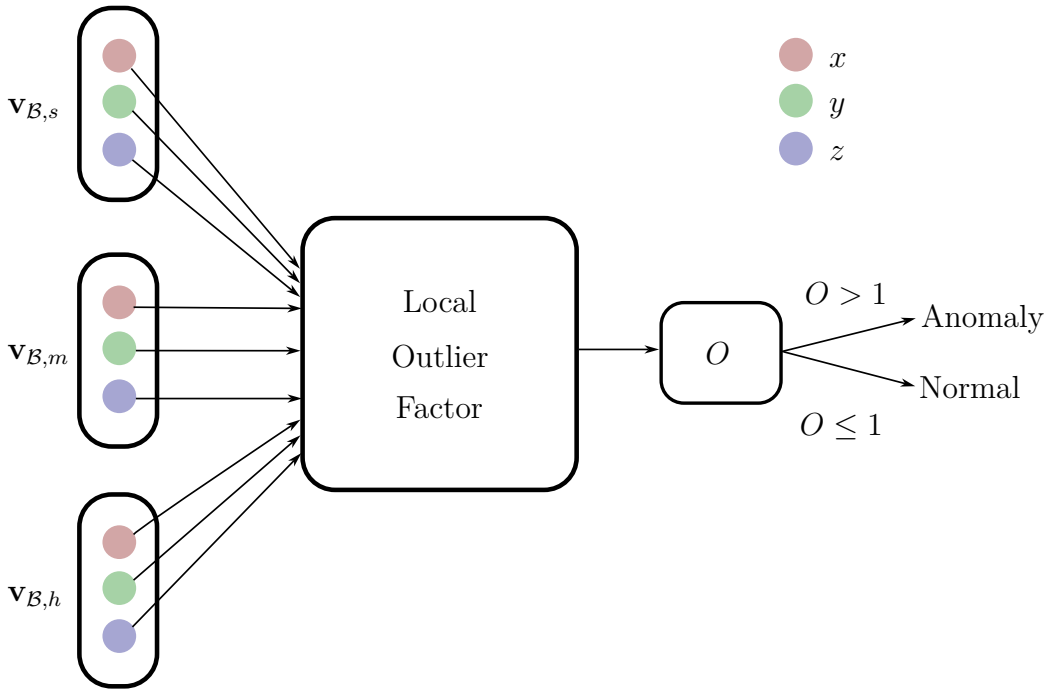

Figure 6.11: The input and output of the local outlier factor algorithm for anomaly detection. The outlier score is passed through a simple *If Statement* to determine whether the current data sample is anomalous or not. When the outlier score is larger than 1, the data sample is anomalous.

A confusion matrix provides a summary of both the true/false positives and the true/false negatives of the LOF prediction algorithm on a test data set. A positive represents the prediction of that a condition is present and a negative represents the prediction that a condition is not present, in this case an anomaly. A false and true of either of these refers to whether the prediction is accurate. For instance, a false negative is when the prediction algorithm classifies a data sample as normal (no anomaly), whereas a false positive is when the prediction algorithm classifies a data sample as being anomalous, when in fact the data sample is normal. The acronyms for false and true positives and negatives are given as FP, TP, FN, TN. An example of the construction of a confusion matrix for detection is provided with Table 6.1.

	Predicted	
	Normal	Anomaly
Normal	**TN**	FP
Anomaly	FN	**TP**

Table 6.1: An example of a confusion matrix. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples.

A parameter within the Sci-kit learn implementation requires the portion of anomalies within the data set. This portion is typically referred to as the contamination parameter. For two largely different contamination parameters 10^{-5} and 0.1, the confusion matrices

of the prediction accuracy of the trained LOF algorithm on the test data set is given in Table 6.2. The test data set is an uniformly selected 40% of 30 orbits of both the Sun reflection anomaly, the magnetic moment disturbance, and normal data.

	LOF	
	Normal	Anomaly
Normal	**144495**	14460
Anomaly	19398	**26085**

(a) Contamination of 10^{-5}.

	LOF	
	Normal	Anomaly
Normal	**90961**	65463
Anomaly	20	**15845**

(b) Contamination of 0.1.

Table 6.2: Comparison of confusion matrices of the local outlier factor algorithm with 10^{-5} and 0.1 contamination values. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples.

The prediction accuracy of the contamination at 10^{-5} is 80.8% overall, 57.35% for the prediction of anomalies and 90.9% for the prediction of normal data samples. The accuracy of the contamination at 0.1, on the other hand, is 62% overall, 99.87% for anomalous data samples and 58.15% for the prediction of normal data samples. None of these results can be viewed in isolation. It cannot simply be concluded that the lowest contamination portion is the best, since it does not perform as well at predicting anomalies in comparison with the higher contamination portion. The weight of importance in predicting anomalies and in prediction normal data samples are not the same. It is of much more importance to accurately predict anomalies. This, however, does not conclude that 0.1 provides the best results, since it is not good at predicting normal data. A detection algorithm which always predicts a data sample as an anomaly will produce a 100% prediction accuracy for anomaly, but will produce a 0% accuracy in predicting normal data samples. It must therefore be tested with the recovery and isolation method in place in order to determine the best value for this parameter. This will be discussed further in Chapter 8.

It can be noted beforehand that the contamination parameter produces high anomaly accuracy at a large contamination percentage and a lower anomaly percentage at a lower contamination percentage, while the vice versa for the prediction of normal data samples. It can, however, be predicted that the prediction of anomalous data samples is of higher importance and the percentage accuracy of this will carry more weight in the optimisation of the parameter compared to the prediction of normal data samples.

6.3. Decision Tree Algorithm

The decision tree algorithm is a method of splitting data samples into different groups based on the thresholds of a given feature. These splits can be done numerous times and depends on the application. The method of MAESM implemented by de Silva et al. [21] provides the moving average feature as additional input to a decision tree, which classifies

the data as anomalous or not. Since decision tree are a supervised learning method, the data of both the simulation orbits with and without anomalies are used to train the decision tree. The decision tree algorithm can also perform multi-class classification, and can therefore also be implemented for anomaly isolation. The implementation of the decision tree algorithm for anomaly detection and isolation in this thesis is due to the application thereof by de Silva et al. [21].

6.3.1. Theoretical Background

The decision tree and random forest algorithms can be implemented to perform binary classification on regular data samples and anomalous data samples. The decision tree and random forest algorithms are supervised learning algorithms that classify data based on threshold splitting [55–58]. Data samples are split based on a threshold of a specific input parameter. The decision tree determines this split with the classification and regression tree (CART) algorithm.

To split the data for the anomalies, however, the input parameter which will be used to make the first split - the root node - must be decided. The Gini index GI measures the probability of a data sample being wrongly classified at a given node. This can be calculated by

$$GI = 1 - \sum_{i=1}^{n} (P_i)^2, \tag{6.14}$$

where P_i is the probability that a data sample is classified for a specific class and n is the number of classes. The split that produces the lowest proportional Gini index will be used as the root node. Thereafter the decision tree will train new nodes after the initial split at the root node and once again the split with the lowest Gini index will be inserted as the next split. The proportional Gini index will therefore decrease after each split (accounting for the percentage of data samples within a node). For this use case, the CART algorithm will be used to optimize the decision tree [59].

The depth of a decision tree determines how many splits occur from the root node to the leaf node, which is the furthest from the first split. If the depth is unspecified, the decision tree will split until all the data samples are perfectly split into anomalous and normal data samples. The larger the depth, however, the more biased the decision tree is to the training data. This depth can be altered to optimize the efficiency and accuracy of the decision tree.

6.3.2. Application for Anomaly Detection and Isolation

The decision tree is trained on both the anomalous and normal data sets from the simulation environment, where the anomalies are the Sun reflection on the sun sensor and the magnetic moment disturbance. The data samples are labelled during the simulation environment as

either anomalous or not. The decision tree algorithm is implemented through the Sci-kit learn package in Python [54]. Figure 6.12 provides a graphical illustration of the inputs and outputs of the decision tree. The only difference between the implementation of the decision tree algorithm for detection and the implementation thereof for isolation is that the training data provided is labelled for the specific anomaly. It can therefore have multiple classes depending on the number of anomalies.

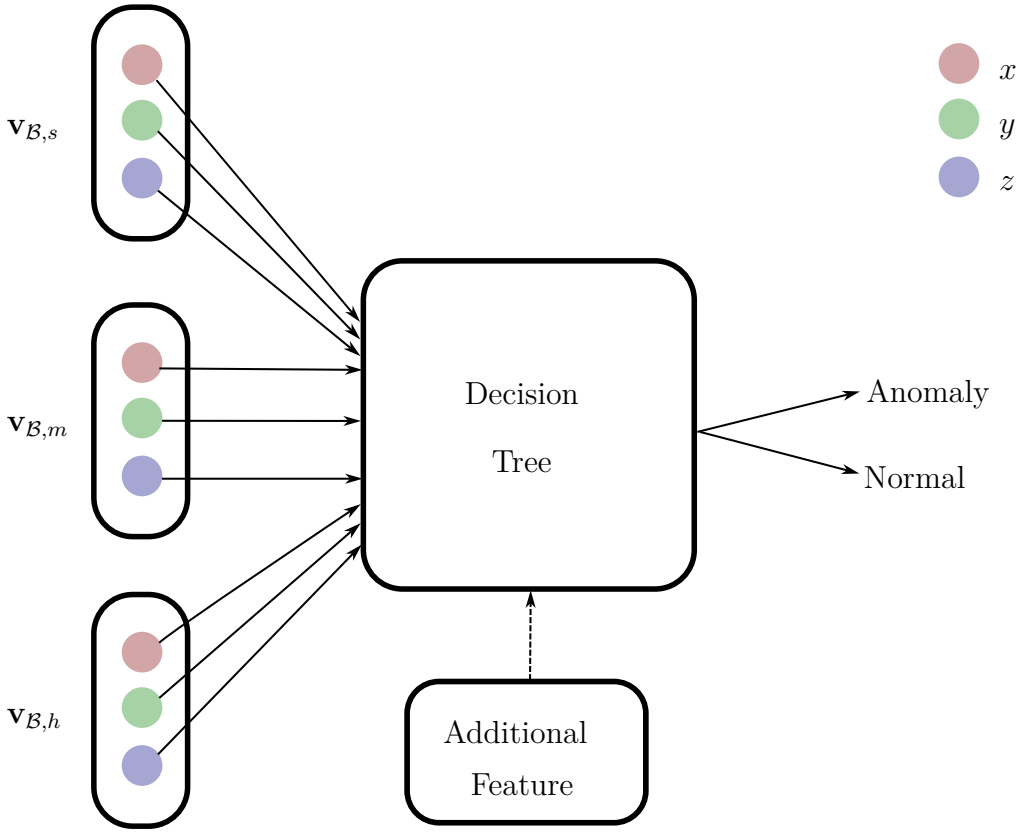

Figure 6.12: The input and output of the decision tree algorithm for anomaly detection. The decision tree algorithm is provided with the sensor measurements and (potentially) with an additional feature from a feature extraction method as input. The decision tree performs binary classification to determine whether a data sample is anomalous or not. The *samples* label provides the percentage of samples within the node with respect to the total training data samples, while the *value* label provides the portion of each class within the node *i.e.* normal samples or anomalous samples.

To demonstrate the splits of a trained decision tree on normal data as well as data from the reflection anomaly, a visual representation of the first three layers of the decision tree is given in Figure 6.13. This is implemented without any additional feature as input, only the sensor measurements as input. From Figure 6.13 it can be derived that the most significant splits are performed on the sun sensor as the root node as well as the second split nodes are both performed on the sun sensor. This is as expected, considering that the anomaly of the Sun reflection is most notable on the sun sensor.

Furthermore, it is clear from the Gini index in every node, that the Gini index decreases after every split, demonstrating that the data samples are getting "purer" after each split.

This excludes the nodes with a small percentage of data samples as these will not necessary have a smaller Gini index as the node above it. The purpose of the decision tree is to determine the split of a given node that will decrease the Gini index the most. Even though the split of the $Sun_z \leq -0.374$ increase the Gini index if **True**, it decreases the Gini index if **False**. It must therefore be noted that the sample size of the **False** split is much larger than the **True** split. The **False** split therefore carries more weight. A better comparison of the decrease in the Gini index is to multiply the samples percentage with the Gini index. The node where the $Sun_z \leq -0.374$ split is performed provides a value of 0.036 after multiplication of the samples percentage with the Gini value, while the **True** split has a value of 0.009 and the **False** node has a value of 0.024 The cumulative value of the Gini index with the samples percentage (taken into consideration after the split) is therefore smaller than before the split.

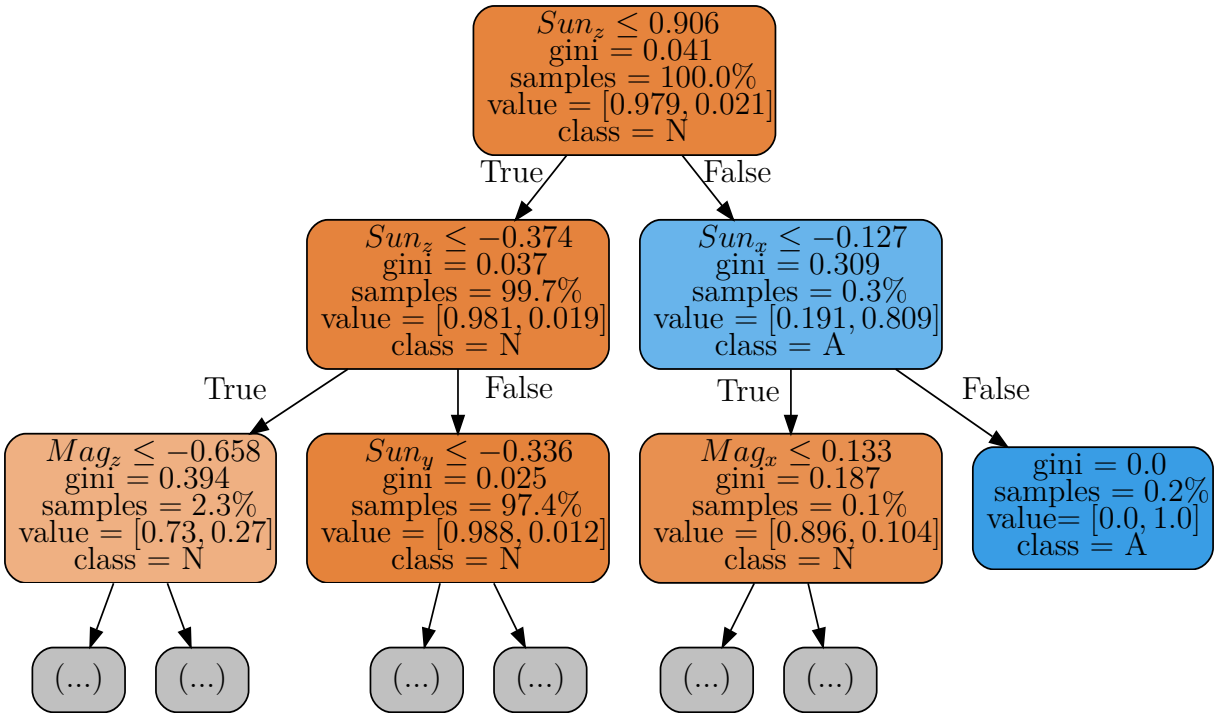

Figure 6.13: The decision tree algorithm as a binary classification of the Sun reflection anomaly. The orange nodes have more than 50% normal samples, while blue nodes have more than 50% anomalous samples. The darker the shade of orange or blue, the larger the percentage of the respective class. The class "A" refers to an anomaly and the class "N" refers to a normal data sample.

The decision tree model is trained with a random 60% of the data set and evaluated with the other 40% of the data set. This 40% of the data set is the data used to construct the confusion matrices in the analysis of the decision tree for anomaly detection section. The random 60% of the data set will be implemented the same for each supervised learning algorithm. The decision tree is trained with an adjusted class weight to take into account the unequal distribution between the number of normal data samples and the number of anomalous data samples. This weight is the parameter that provides additional weight to

the prediction accuracy of a particular class. If an algorithm always predicts that a data sample is normal, the prediction accuracy of it will be very high since most of the data samples are normal. This, however, is not an accurate model. To account for this, the prediction accuracy of the anomalies require additional emphasis. This is implemented with the class weight parameter calculated as

$$weight = \frac{n_{samples}}{n_{classes} \times n_y}, \tag{6.15}$$

where the parameter $n_{samples}$ is the total number of training data samples and the parameter $n_{classes}$ is the number of classes which is equal to two for detection and three for isolation. The parameter n_y is the number of occurrences of the value y in the data set, where y can be equal to any of the input classes [54]. This calculation then adjusts for the unequal number of normal and anomalous occurrences in the data set during the training of the supervised learning algorithms.

6.3.3. Analysis for Anomaly Detection

To determine the performance of the decision tree for detection and isolation confusion matrices as well as the average prediction accuracy per orbit of the decision tree are provided for 30 orbits. The confusion matrix provided in Table 6.3 is the combined prediction of both the normal data set, the Sun reflection anomaly, and the magnetic moment disturbance. This test data set is an uniformly distributed 40% of the provided data sets. The prediction accuracy for binary classification is equal to 99.87%. Even though this is a very high prediction accuracy, the influence of the recovery methods can change the prediction accuracy of the detection methods, since the data changes due to the influence of the recovery method on the attitude of the satellite.

	Decision tree	
	Normal	Anomaly
Normal	**158837**	118
Anomaly	139	**45344**

Table 6.3: Confusion matrix of the decision tree for detection on a test data set. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples.

To determine what the detection accuracy of the decision tree are on each anomaly, the performance thereof is plotted on Figure 6.14. It can be derived from Figure 6.14 that the decision tree is much more accurate in predicting the magnetometer anomaly of magnetic moment disturbance than detecting the sun sensor anomaly.

The most plausible explanation for the higher accuracy in the prediction of the magnetic moment disturbance is that during the training and prediction of anomaly detection,

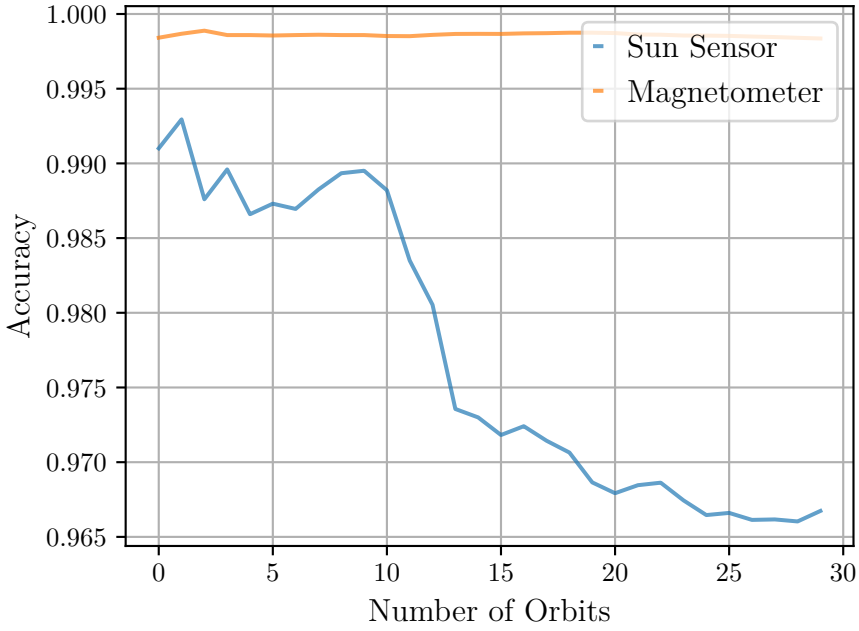

Figure 6.14: Average estimation metric per orbit of decision tree for both the sun sensor anomaly and the magnetic moment disturbance anomaly. The legend is the sensor that experiences the anomaly.

the magnetometer and sun sensor anomalies are grouped together as one class, namely anomalies. This therefore causes the decision tree to be biased towards the magnetic moment disturbance. Since only one decision tree is trained and there is no averaging of the data selected due to bootstrapping, it is expected that the decision tree will split on the most significant Gini index splits, which favours the magnetic moment disturbance, since this occurs more often. Even though the class weight is calculated with Equation 6.15, this still provides a single class for the magnetic moment disturbance and the Sun reflection. This is most likely the reason for the higher prediction accuracy of the magnetic moment disturbance anomaly in comparison with the Sun reflection by the decision tree model.

6.3.4. Analysis for Anomaly Isolation

The implementation of the decision tree for anomaly isolation can be done in two ways. The one is to train the decision tree on only the anomalous data samples, while the other is to train the decision tree on both the anomalous data and the normal data samples. The reasoning behind training the decision tree on only the anomalous data is that the isolation is, typically, only implemented to classify the anomaly after it has been detected that the data sample is anomalous by the anomaly detection algorithm. The isolation and detection can also be a single step where a single algorithm classifies a data sample as either “normal” or as the sensor causing the anomaly.

There are potentially positive and negative aspects to both these methods. Firstly, it

is assumed that a detection algorithm is more accurate in classifying anomalies, since it is only a binary classification problem compared to the isolation algorithm which is usually a multi-class classification problem.

In Table 6.4 it can be noted that the decision tree classifies normal and anomalous data samples better if the sun sensor and magnetometer are in two separate classes. This is most evident when columns labelled as "Sun" and "Mag" of Table 6.4a and the rows labelled as "Sun" and "Mag" are merged into one. This is essentially a binary classification and a detection algorithm. A data sample is therefore detected as an anomaly when it is not classified as "Normal" and therefore can be classified as either "Sun" or "Mag". Providing the false positives and the false negatives of both models demonstrate this. The decision tree trained for detection — "Sun" and "Mag" are in a single class — have 139 false negatives and 118 false positives. The decision tree trained for isolation and viewing the "Sun" and "Mag" class as a two separate classes have 125 false negatives and 103 false positives on the test data set.

Table 6.4a is be the result for implementing the decision tree either as both detection and isolation simultaneously or as an isolation algorithm that can reclassify a data sample as "normal" after the detection algorithm classified the data sample as anomalous. Table 6.4b is the results of the decision tree when only implemented for isolation and thus classifies which sensor is responsible for the anomaly.

	Decision tree		
	Normal	Sun	Mag
Normal	**158852**	3	100
Sun	13	**2909**	0
Mag	112	0	**42449**

(a) Including normal data.

	Decision tree	
	Sun	Mag
Sun	**2893**	1
Mag	0	**42751**

(b) Excluding normal data.

Table 6.4: Confusion matrix of isolation for the decision tree algorithm. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples. The magnetometer is abbreviated to "Mag" and the sun sensor is abbreviated to "Sun".

6.4. Random Forest

The random forest algorithm trains a number of decision trees and provides a classification output from a majority vote of the decision trees. The decision trees are trained on subgroups of the dataset and thus each decision tree will differ. This leads to higher classification accuracy in general [60].

6.4.1. Theoretical Background

The random forest algorithm is an extension of the decision tree. It generates various decision trees from the data set and provides a classification output p, which can be any of the labelled classes, for a data sample $\mathbf{D}^*$, by utilizing a majority vote **mode**() from all the decision trees classification outputs $\mathbf{P}$. Random forest utilizes bootstrap aggregating $ba()$ to ensure that new random data sets $\mathbf{D}'$ with size of n' is created [56–58]. Bootstrap aggregating a data set, simply means that the training data for a model is sampled uniformly and the sampled data is replaced in the original dataset. This means that a single data sample can be utilized numerous times for the training of multiple decision trees. This ensures that every trained decision tree is independent from the training of the other decision trees within the random forest. A number k of individual decision trees is then trained with Algorithm 6.2.

Algorithm 6.2: Training of random forest from individual decision trees.

1: `Input = Sensor measurements (`$v_{\mathcal{B}}$`)`
2: **for** $i \coloneqq 1$ **to** k **do**
3: $\quad \mathbf{D}'_i = ba(\mathbf{D})$
4: $\quad DT_i = CART(\mathbf{D}'_i)$, where DT_i is a single decision tree.
5: **end for**
6: `Output = Trained random forest consisting of a number` k `decision trees`

The classification output p, is then provided with Algorithm 6.3. The classification output p can either be 0 or 1 for binary classification or any number within the range of classes in multi-class classification.

Algorithm 6.3: Prediction of random forest from individual decision trees.

1: **for** $i \coloneqq 1$ **to** k **do**
2: $\quad \mathbf{P}_i = \mathbf{DT_i}(\mathbf{D}^*)$
3: **end for**
4: $p = \mathbf{mode}(\mathbf{P})$

6.4.2. Application for Anomaly Detection and Isolation

The implementation of the random forest algorithm for anomaly detection is very similar to that of the decision tree algorithm. The random forest is also implemented with the Scikit-learn model provided in Python [54]. The random forest is trained with a hyperparameter of a 100 decision trees. The more decision trees are implemented the more accurate the random forest algorithm is, but this increases the training duration and decreases the computationally efficiency of the model. The random forest algorithm is trained on both the normal data and the anomalous data from the simulation environment. Thereafter the model is stored and during implementation of fault detection and isolation,

the model is implemented to classify the current data sample as anomalous or not. The random forest model is trained with a random 60% of the data set and evaluated with the other 40% of the data set. This 40% of this data set is the data used to construct the confusion matrices in the analysis of the random forest section. The class weight parameter is also adjusted according to Equation 6.15 for the training of the random forest model.

The implementation of the random forest algorithm for anomaly isolation distinguishes between each anomaly. Two implementations are possible for anomaly isolation. The one assumes that the detection method can be invalid and can reclassify the data sample as normal during the isolation step, while the other assumes that the isolation can only predict which anomaly is responsible for the nature of the current data sample. Both of these methods will be analysed and the positive and negative aspects thereof will be discussed.

6.4.3. Analysis for Anomaly Detection

The tested results of both anomalies after training are provided with both the confusion matrix given in Table 6.5. The average prediction accuracy per orbit for each individual anomaly for 30 orbits is shown in Figure 6.15. The random forest algorithm performs significantly better than that of the decision tree when comparing Table 6.3 with Table 6.5. The prediction accuracy of the random forest is 99.95% on the testing data.

	Random forest	
	Normal	Anomaly
Normal	**158905**	50
Anomaly	42	**45441**

Table 6.5: Confusion matrix of the random forest for detection on a test data set. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples.

To determine the performance of the detection algorithm on each anomaly, the average prediction accuracy per orbit is given for the anomaly of each sensor. The random forest is more accurate in predicting the sun sensor anomalies compared to predicting the magnetometer anomalies. This is different from the performance of the decision tree for anomaly detection. The random forest algorithm trains multiple decision trees based on subsets of the training data based on bootstrapping. This generally increases the average prediction accuracy when taking the majority vote of all the decision trees. Even though some decision trees within the random forest can be biased to a specific anomaly, the majority votes hopefully reduces these biased decision trees. The prediction accuracy is therefore not just based on the anomaly with the most occurrences, but based on all anomalies. Both anomalies are predicted with a very high accuracy. The plausible explanation for the lower prediction accuracy of the magnetometer anomaly is that it is

less distinguishable from the normal data than that of the sun sensor anomaly. The sun sensor anomaly has a larger effect on the control than that of the magnetometer anomaly as is evident when comparing Figure 5.3 with Figure 5.14.

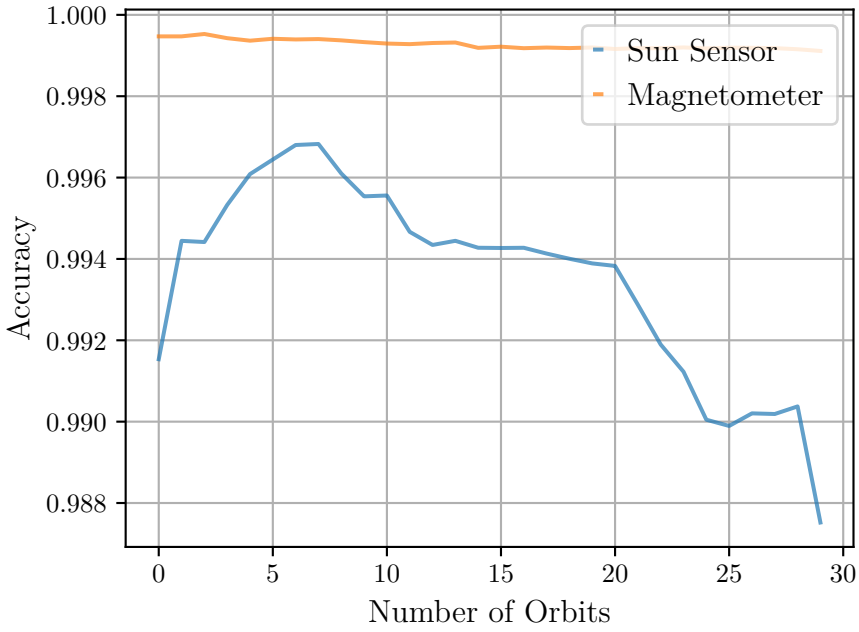

Figure 6.15: Average estimation metric per orbit of random forest for both the sun sensor anomaly and the magnetic moment disturbance anomaly. The legend is the sensor that experiences the anomaly.

6.4.4. Analysis for Anomaly Isolation

The analysis of the random forest for anomaly isolation is done with the comparison of the confusion matrices of the random forest trained on both normal data and anomalous data samples with the random forest trained on only the anomalies. In Table 6.6a it can be noted that if the "Sun" and "Mag" columns and rows were merged, it can be renamed as a single class, namely Anomaly. Table 6.6a can therefore be compared with Table 6.5. It can be noted that the differences in the false positives and false negatives can be due to randomness in the training during the bootstrap method. This difference, however, is very small. Both Table 6.6b and Table 6.6a have a 100% prediction accuracy in distinguishing Sun reflections (Sun) from magnetic moment disturbance torques (Mag). It cannot therefore be determined whether implementing detection (Table 6.5) and then isolation (Table 6.6b) separately have better results than only implementing isolation directly (Table 6.6a). This will depend on the time duration for each prediction as well as the influence of the recovery on both accuracies. If the recovery decreases the prediction accuracy of both the detection and isolation accuracies, it will probably be better to only implement isolation directly. The two events, prediction and isolation, are dependent

events and not independent. The prediction accuracy will therefore be the multiplication of the accuracy of the detection method and the accuracy of the isolation method.

	Random forest		
	Normal	Sun	Mag
Normal	**158902**	0	53
Sun	1	**2921**	0
Mag	39	0	**42522**

(a) Anomaly Isolation including normal data.

	Random forest	
	Sun	Mag
Sun	**2894**	0
Mag	0	**42751**

(b) Anomaly Isolation excluding normal data.

Table 6.6: Confusion matrix of isolation for the random forest algorithm. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples. The magnetometer is abbreviated to "Mag" and the sun sensor is abbreviated to "Sun".

6.5. Support Vector Machines Algorithm

A study conducted by Colagrossi et al. [61] implemented support vector machines (SVMs) for sensor and actuator failure detection on small satellites systems. This example, as well as many other fault detection research in other fields, implement SVMs for anomaly detection. The major advantage of SVMs is that it is able to increase the feature space to determine splits between groups of data. SVMs is a supervised learning algorithm and through training the algorithm on labelled data the algorithm is able to perform classification on many complex data sets.

6.5.1. Theoretical Background

The feature space refers to the dimensions of the inputs to the prediction model. SVMs create a hyperplane between the feature space to split the classes from each other. This hyperplane is created to increase the distance between the data samples nearest to the hyperplane [62, 63]. The hyperplane can be defined as

$$\mathbf{w}^{\mathrm{T}}(\mathbf{f}) + b = 0 \tag{6.16}$$

where $\mathbf{w}$ is the normal vector to the hyperplane. It is not, however, necessarily a unit vector. This is also constrained to ensure that all vectors $\mathbf{f}_n$ are on the correct side of the hyperplane with

$$\mathbf{w}^{\mathrm{T}}(\mathbf{f}_i) + b \geq 1, \qquad \text{for all} 1 \leq i \leq n. \tag{6.17}$$

The hyperplane in its most simplified form on a binary classification problem is linear and divides the data into two halves as shown in Figure 6.16. The distance between the

hyperplane and any vector $\mathbf{f}_i$ can be given as

$$d(\mathbf{f}_i) = \frac{\mathbf{w}^{\mathrm{T}}(\mathbf{f}_i) + b}{\|\mathbf{w}\|^2} \tag{6.18}$$

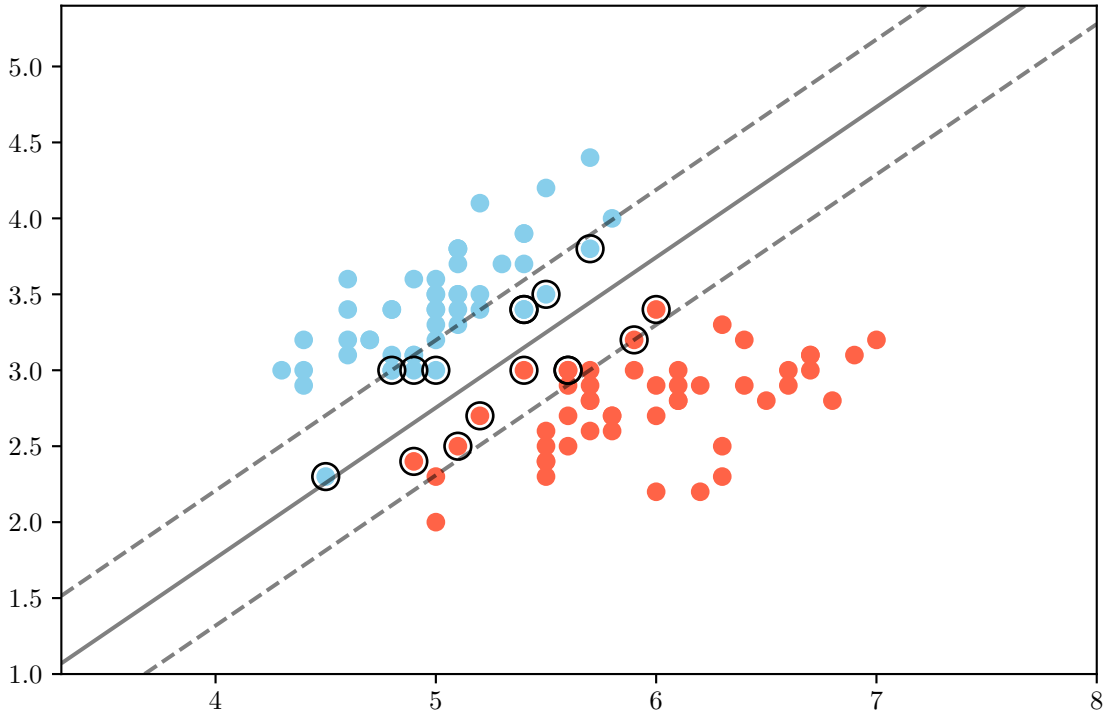

Figure 6.16: Support vector machine example of splitting data into two groups based on a hyperplane. The blue and red circles are the data samples from two different classes. The black circles emphasize the support vectors. The solid black line is the hyperplane used to classify data samples and the data samples between the solid line and the dashed lines are the support vectors. The dashed lines are calculated based on the soft-margin algorithm.

For binary classification, all vectors $\mathbf{f}_n$ will be classified into a group by determining on which side of the hyperplane the point vector is. To increase the margin between all point vectors $\mathbf{f}_n$ and the hyperplane, the minimum distance of all the point vectors and the hyperplane is required. The objective of the algorithm is therefore to maximise the minimum distance between the hyperplane and any point vector $\mathbf{f}_i$ for all data samples n. The sub-gradient descent algorithm can be implemented to calculated the hyperplane as

$$\boldsymbol{g}(\mathbf{w}, b) = \lambda \|\mathbf{w}\|^2 + \left[\frac{1}{n}\sum_{i=1}^{n} \mathbf{max}\left(0, 1 - \mathbf{w}^{\mathrm{T}}\left(\mathbf{f}_i\right) + b\right)\right], \tag{6.19}$$

where λ increases and decreases the margin size, to increase the number of training vectors $\mathbf{f}_i$ that is on the correct side of the hyperplane. The function $\boldsymbol{g}$ is a convex function and can be solved with an adaptation of classic gradient descent methods.

The linear hyperplane is not always feasible, since certain data samples cannot be

divided with a single line. There are two solutions to this problem. The first is to increase the dimension size m of $\mathbf{R}^m$. For instance, increasing m from 2 to 3 with

$$\mathbf{f}[2] = \sqrt{\mathbf{f}[0]^2 + \mathbf{f}[1]} \tag{6.20}$$

provides a higher dimension for the split of the data. This can be implemented with any equation for increasing m. Another method to divide the data into different classes is by changing the kernel from linear to non-linear kernels, such as a polynomial kernel.

With this overview of the algorithm it can be implemented to perform anomaly detection and isolation. The SVMs algorithm is trained on labelled data to calculate the hyperplane that best splits the training data into the number of classes.

6.5.2. Application for Anomaly Detection and Isolation

The SVMs algorithm is implemented similarly to that of the decision tree and random forest. The SVMs model is trained on the anomalous and normal data from the simulation. It is trained with the Scikit-learn implementation of the SVMs algorithm in Python [54]. The model is trained with the same split and setup of the training data as that of the decision tree and random forest algorithms. The class weight parameter is also adjusted according to Equation 6.15 for the training of the SVMs model.

Two methods for isolation is implemented. The one method assumes that there is an anomaly and that it can only classify the sensor responsible for the current data sample. The other method, however, assumes that the detection algorithm can be incorrect in its prediction and can therefore reclassify the data sample as “normal”. This method can also be implemented without anomaly detection, and will depend on a few factors, such as the computational efficiency of the isolation method. Isolation methods are usually more complex and less computationally efficient than detection methods, unless both the isolation and detection methods are binary classification (there are only two anomalies and the normal data are excluded from the isolation method).

6.5.3. Analysis for Anomaly Detection

The SVMs are implemented for anomaly detection by performing binary classification on the normal data samples as well as the merged group of the magnetic moment disturbance anomaly and the Sun reflection anomaly. In Table 6.7 it can be noted that the SVM is more prone to predicting anomalies. The SVMs overall prediction accuracy is 98.13%.

A plot of the average prediction accuracy per orbit is provided in Figure 6.17 to determine the prediction accuracy of SVMs on each anomaly individually. This demonstrates that the SVMs algorithm performs better on the magnetometer anomaly than on the sun sensor anomaly. This is most possibly due to the way in which the SVMs

	Support Vector Machines	
	Normal	Anomaly
Normal	**155383**	3572
Anomaly	40	**45443**

Table 6.7: Confusion matrix of the support vector machines for detection on a test data set. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples.

algorithm splits the two classes with a single hyperplane. To draw a hyperplane between this merged anomaly group and the normal data group is not the same as the method of the decision tree. The decision tree performs multiple splits, while the SVMs perform a single split as a hyperplane within the feature space. The SVMs algorithm will therefore be biased towards the magnetometer anomaly, since this is the anomaly that occurs most frequently within the anomaly group. To provide the best prediction accuracy, the SVMs will provide a hyperplane that splits most of the data into the correct category and not the individual subgroups within the anomaly class. This bias towards the magnetometer anomaly is emphasized in SVMs due to the single split nature of the algorithm, since different anomalies do not manifest as a single identifiable group within the feature space. This does not create an easy split between the anomaly and normal class.

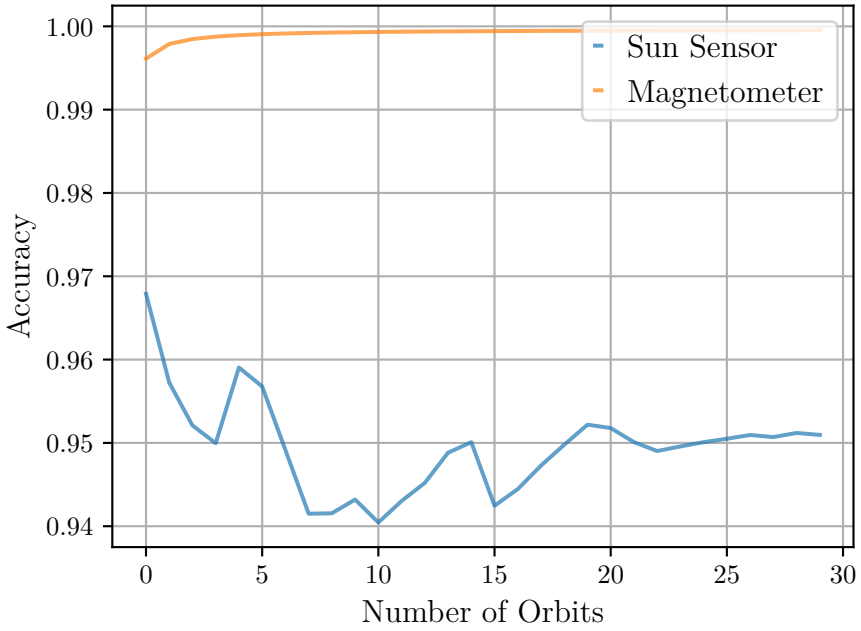

Figure 6.17: Average prediction accuracy per orbit of support vector machines as anomaly detection for both the Sun reflection and magnetic moment disturbance anomaly. The legend is the sensor that experiences the anomaly.

6.5.4. Analysis for Anomaly Isolation

The analysis of the SVMs performance for anomaly isolation is done by comparing Table 6.8a with Table 6.8b. Table 6.8a provides the confusion matrix of the isolation method that includes the normal data as a class. When merging both the rows and the columns of "Sun" and "Mag", the structure of the confusion matrix for only anomaly detection is replicated from Table 6.7. From this it can be derived that the SVMs more accurately predicts anomalous data than it predict anomalous data samples, as only 13 magnetometer data samples are incorrectly classified as normal. This is the desired outcome as the false negatives are more crucial than that of the false positives, since it is the false negatives that will cause the sensor anomalies to be incorporated in the EKF measurement update step. The false positives on the other hand will only implement the recovery method without it being necessary.

The trend of accurately distinguishing between the Sun reflection and magnetic moment disturbance anomaly continues with the SVMs algorithm as it predicts with 100% accuracy the difference between the "Sun" and "Mag" in both confusion matrices given in Table 6.8. This emphasizes once again that the two anomalies are very different and grouping them into a single class for anomaly detection might not always be the best solution. To determine the final performance of all the methods requires the recovery step. This will determine how bias the trained model is to the training data and whether it can accurately classify new data that is different due to the implementation of the recovery method.

	Support Vector Machines		
	Normal	Sun	Mag
Normal	**155054**	90	3811
Sun	0	**2922**	0
Mag	13	0	**42548**

(a) Anomaly Isolation including normal data.

	Support Vector Machines	
	Sun	Mag
Sun	**2894**	0
Mag	0	**42751**

(b) Anomaly Isolation excluding normal data.

Table 6.8: Confusion matrix of isolation for Support Vector Machines. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples.

6.6. Computational Efficiency

The computational efficiency aspect is important when developing the prediction methods for on board the satellite, since the satellite is constraint by on-board computation. To finally decide which method is preferred for a system depends on the systems capabilities. The exact time of the prediction can not be determined for on-board the satellite, only the relative time difference can be determined, since the prediction time will differ depending on the capabilities of the computer as well as what processes are running simultaneously.

From Table 6.9a it can be calculate that the random forest algorithm for detection is 45 times slower in predicting an anomaly than that of the decision tree algorithm. This makes sense as the random forest is a combination of a 100 optimised decision trees from which the mean is used to classify a data sample. This is a substantial difference and is the reason why less computation intensive algorithms are implemented for anomaly detection, since anomaly detection runs continuously along with all the rest of the processes. The classical approach of anomaly isolation is only required after an anomaly is detected. This therefore means that the more computationally intensive algorithms are only implemented once in a while. The anomaly detection algorithms should rather predict a false positive and cause the anomaly isolation method to be invoked to accurately classify the data sample. The SVMs prediction time lowers significantly from detection to isolation. This is due to the implementation change to "one vs rest" of the SVM, which is an optimised implementation.

This decision depends on the specific application and the constraints of the satellite. If the satellite has the capabilities of running the more computationally intensive algorithm continually with the rest of the processes, then this might be desirable by the engineers.

	Time μs	
	Sun	Mag
LOF	3798.61	3894.5
DT	**166.91**	**177.36**
RF	**7856.43**	**7832.84**
SVMs	1367.23	1371.46

(a) Average time per prediction for anomaly detection methods. These are the anomaly detection methods that are trained on the anomalies grouped as a single class.

	Time μs	
	Sun	Mag
DT	**174**	**159.95**
RF	**7705.08**	**7865.67**
SVMs	230	198.68

(b) Average time per prediction for anomaly isolation methods. These are the anomaly isolation methods that are trained on both anomalous and normal data.

Table 6.9: Average time per prediction method for both anomaly detection and anomaly isolation. The average times are given in μs and are sub divided depending on the anomaly, where "Sun" refers to the Sun reflection anomaly and "Mag" refers to the magnetic moment disturbance anomaly.

6.7. Summary

The feature extraction methods discussed in this chapter focuses on extracting features that provide additional information on whether a data point is an anomaly or not. For the LOF algorithm, this is done by providing a value for the "irregularity" of that data point relative to its nearest neighbourhood. The moving average of the estimated sensor measurements is an adaptation of work by de Silva et al. [21] to provide an estimate of the sensor measurements based on the previous sensor measurements, as well as the control inputs. This can be extended to calculate a moving average for the difference between the

estimated sensor measurements and the actual sensor measurements. The outlier score O and the moving average $\mathbf{V}_k$ can be provided as additional inputs to the detection and isolation algorithms to possibly enhance the accuracy by improving the information gain of the features.

For the detection of anomalies, different binary classification methods are discussed. This includes supervised learning methods such as decision tree, random forest and support vector machines (SVMs), as well as an unsupervised learning method namely the local outlier factor algorithm. Both supervised and unsupervised learning methods for detection provide a binary split between anomalous and normal data points, while supervised learning methods can also be implemented for anomaly isolation and can classify which anomaly is responsible for the nature of the current data sample.

The random forest algorithm is the most accurate in detecting and isolating anomalies. It is, however, also the most computationally intensive algorithm. LOF is the algorithm with the lowest false positive value and the decision tree algorithm is the most computationally efficient algorithm. Both the positive and negative aspects regarding the prediction accuracy and the computational efficiency will have to be weighed by the engineers implementing the FDIR on the satellite. This chapter provides detailed analysis and discussion of the feature extraction block as well as the detection and isolation block of Figure 1.2.

Chapter 7

Anomaly Recovery

The main purpose of the FDIR is to recover from the sensor anomalies. The fault detection and isolation methods are a means to predict the anomaly, but that still requires a solution to the anomaly. The recovery methods are evaluated based on the simulation of perfect detection and isolation to demonstrate the effectiveness thereof. The effectiveness of the recovery methods will be evaluated based on the estimation metric, which demonstrates whether the EKF remains accurate and robust during the simulation.

The estimation metric is used for analysis since this ensures that the entire FDIR technique solves the problem induced by the anomalies. The maximum estimation metric that is considered acceptable is determined as 10° since this is considered a reasonable error for the satellite to still be able to meet it's mission requirements. This does, however, depend on the accuracy required for the pointing of the payload towards the Earth. The anomalies also have different influences on the EKF and an estimation metric of 10° is reasonable given the conceptual design of the satellite.

The recovery of the actuator failure or recovery that requires changes of the control law is not within the scope of this thesis, since this thesis focusses on recovery of the EKF and not of the controller. These are two separate blocks in Figure 1.2 and function as two separate subsystems within the ADCS. The recovery methods will therefore only be developed with the focus on the EKF, although the anomaly detection and isolation are analysed for actuator failures. This is since the anomaly detection and isolation analysis thereof provides insight into whether the anomaly detection and isolation techniques can be used for future work during the development of recovery methods for the controller.

7.1. Recovery Methods

The recovery methods are implemented after the detection of an anomaly and the isolation and classification of the error. Three different methods of recovery are proposed and compared. These methods aim to mitigate the effect of the anomaly on the reliability and stability of the EKF. If no recovery method is implemented the anomalous sensor measurement will be used in the EKF measurement update. This influences the estimation as discussed in Chapter 5.

A method that ignores the sensor measurement from the EKF measurement update of the detected anomalous sensor is referred to as the *EKF-ignore*. This is the most straightforward approach, since the anomalous sensor measurement does not enhance the accuracy of the EKF and can therefore be ignored during the time steps that the sensor is anomalous. This, however, will highly depend on the accuracy of the anomaly detection and isolation methods. A detection method with low accuracy in predicting the anomaly will create instability of the EKF, since many anomalous measurements will still be included in the measurement update of the EKF.

To provide a method that is not that dependent on the accuracy of the detection and isolation methods another technique is considered that recalculates the EKF up until the time step when the anomaly is detected without the anomalous sensor. This method is triggered every single time the prediction shifts from anomalous to normal or from normal to anomalous. During the anomalous periods the *EKF-reset* method ignores the sensor measurement. This method is introduced based on the assumption that multiple anomalous measurements are not detected. This method is referred to as the *EKF-reset* method. The parameters required to update the EKF such as the control torques, sensor measurements need to be in a buffer. By implementing a buffer $\mathbf{E}_n$ of size n of the parameters $\mathbf{e}_k$ that are used to update the EKF as

$$\mathbf{E}_n = \begin{bmatrix} \mathbf{e}_{k-n} & \mathbf{e}_{k+1-n} & \cdots & \mathbf{e}_k \end{bmatrix}, \tag{7.1}$$

the EKF can be recalculated up until time step k. The gain matrix $\mathbf{K}_k$ and the state covariance matrix $\mathbf{P}_k$ are *reset* to the initial values of the matrices. This is because of the uncertainty of when the matrices within the buffer were trustworthy, since multiple sensor anomalies could've occurred before the buffer time step $k-n$ which are not accounted for. If a sensor failure is detected, the measurements of the anomalous sensor is removed from the buffer $\mathbf{E}_n$ for all the time steps and the EKF is updated with the data in the buffer. The EKF is, therefore, *reset* with the EKF matrix parameters at time step $k-n$ and updated from time step $k-n$ to k. The number of time steps n must, however, be optimized based on the computational time used to *reset* the EKF and the size of the memory buffer. If the sensor which was detected to have anomalous behaviour changes back to normal operation, the sensor will be included in the measurement update of the time step k, since it was anomalous for the time steps up until k.

A method that combines the *EKF-ignore* and the *EKF-reset* method is referred to as *EKF-combination*. The *EKF-combination* method ignores the sensor measurement for a N number of time steps just as the *EKF-ignore* method. After N number of detected sensor anomalies, the EKF is *reset* according to the *EKF-reset* methodology. This method is introduced due to the constant *reset* of the *EKF-reset* strategy. The constant *reset* of the EKF will be very time consuming and might not perform well with high prediction

accuracies. The EKF will continuously *reset* the EKF every time an anomaly is predicted and might influence the convergence of the EKF. The *EKF-combination* is implemented to increase the efficiency and hopefully attaining the result where the *resetting* of the EKF is not detrimental and the multiple incorrectly classified anomalies do not drift the EKF from an accurate measurement over time. The number N of predicted anomalies before *resetting* is chosen as 20 after fine tuning the parameter.

7.2. Analysis of Recovery Methods

To determine the best recovery method, a comparison is done on the estimation metric of each recovery method with a 100% prediction accuracy for both the detection and isolation. This is done to determine what the best possible outcome is for each of the recovery methods given that the other subsystems of the FDIR work perfectly. In order to determine whether the recovery method provides a robust EKF, the results are given as an average of the estimation metric for each orbit for a duration of 30 orbits.

7.2.1. Analysis for the Sun Reflection Anomaly

The simulation environment is implemented with the Sun reflection anomaly and each of the recovery methods are individually implemented during 30 orbits. The detection and isolation algorithms provide perfect results during this simulation. The estimation metric is averaged for each orbit for ease of analysis. The comparison of the different recovery methods are provided in Figure 7.1. It can be seen that the *EKF-ignore* method outperforms both the *EKF-combination* and *EKF-reset* methods. The estimation metric without any recovery is also given as "None" and it is clear that the EKF performs better without any recovery compared to the *EKF-reset* method at a 100% prediction accuracy. This is due to the *EKF-reset* method not ensuring convergence of the EKF and increasing the buffer size might solve this issue, but this also decreases the computational efficiency. The *EKF-ignore* method reduces the estimation metric to a result similar to that of the EKF without any anomaly occurring.

7.2.2. Analysis for Magnetic Moment Disturbance Anomaly

The simulation environment is implemented with the magnetic moment disturbance anomaly and each of the recovery methods are individually implemented during 30 orbits. The detection and isolation algorithms provide perfect results during this simulation. The estimation metric is averaged for each orbit for ease of analysis. The comparison of the different recovery methods are provided in Figure 7.1. It is clear that the *EKF-ignore* method once again outperforms the other methods at 100% prediction accuracy.

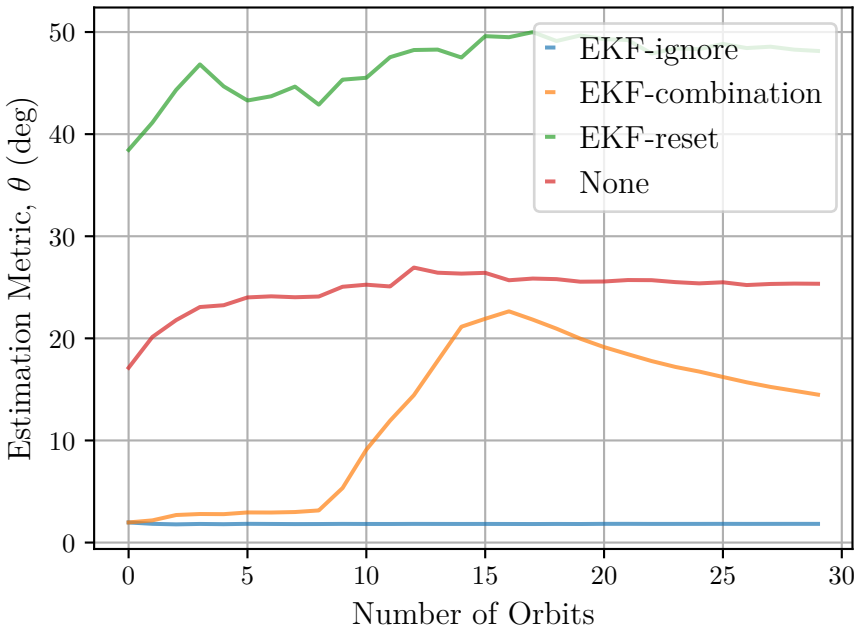

Figure 7.1: Comparison of the average estimation metric θ per orbit for 30 orbits for each recovery method during the Sun reflection anomaly. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

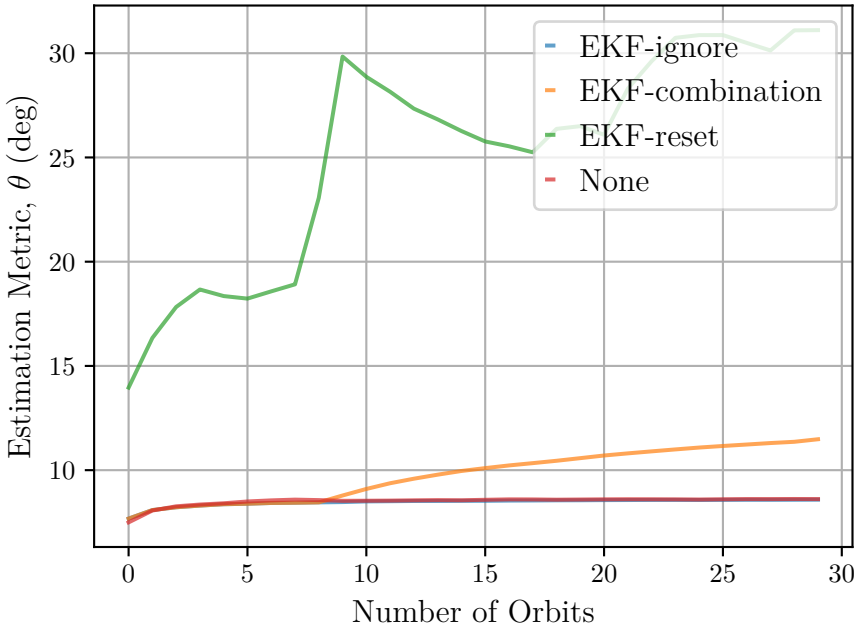

Figure 7.2: Comparison of the average estimation metric per orbit for 30 orbits for each recovery method during the magnetic moment disturbance anomaly. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

It can be seen that only ignoring the magnetometer measurement during the measurement update of the EKF does not significantly enhance the estimation performance. This

can be derived from the fact that "None" and *EKF-ignore* follows almost the same trend in Figure 7.2. The *EKF-ignore* is slightly below the line without any recovery ("None"), but this is a small difference as is more evident in Figure 7.3.

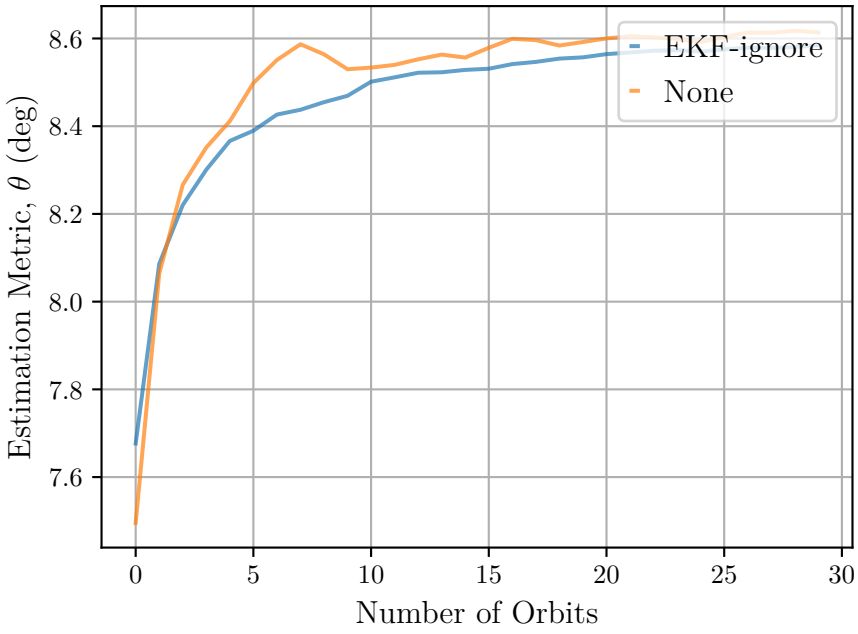

Figure 7.3: Comparison of the average estimation metric per orbit for 30 orbits for *EKF-ignore* recovery method and with no recovery (None) during the magnetic moment disturbance anomaly. The sun sensor and magnetometer are alternated so that the sun sensor is implemented first and the magnetometer is implemented last in the EKF measurement update. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

This can be due to the order of the measurement update where the magnetometer is the first sensor in the measurement update and the sun sensor is last. This is confirmed by reordering the sensors and implementing the same test as seen in Figure 7.4, where the magnetometer is now last in the measurement update sequence of the EKF. The sun sensor will therefore have a more prominent impact on the EKF compared to the magnetometer due to the sequence of the sensors in the EKF measurement update.

It can also be noted that the *EKF-ignore* method shown in Figure 7.4 does not reduce the estimation metric to the same result as without any anomaly. This is due to the magnetic disturbance torque influencing the EKF. The magnetic moment disturbance is still evaluated even though the recovery does not always enhance the performance of the EKF by the same margin as in the case of the Sun reflection anomaly. This is because it still provides an evaluation for the prediction accuracy of the detection and isolation algorithms and is an example of the group of anomalies that can influence both the sensor and create a disturbance torque on the satellite.

From both Figure 7.1 and Figure 7.2 it is evident that the *EKF-ignore* method outperforms the other recovery methods. This, however, is implemented with a perfect

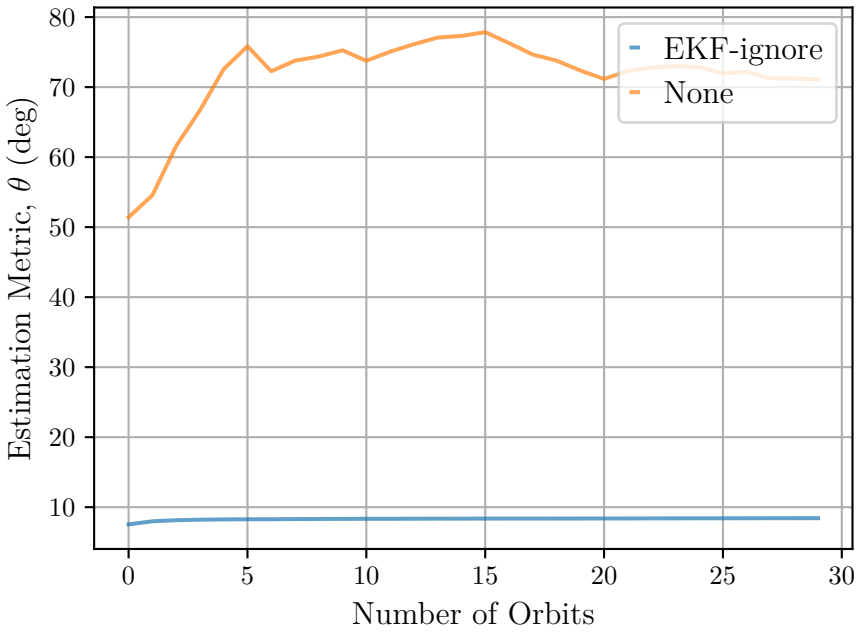

Figure 7.4: Comparison of the average estimation metric per orbit for 30 orbits for *EKF-ignore* recovery method with no recovery (None) during the magnetic moment disturbance anomaly when the sequence for the measurement update is changed. The magnetometer is implemented last in the EKF measurement update and the other sensors are shifted. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

detection and isolation. If none of the detection and isolation algorithms are able to provide results for the *EKF-ignore* that are better than the *EKF-combination* at perfect detection and isolation, then the recovery methods must be re-evaluated with the best prediction and isolation methods.

The result of Figure 7.4 when the sequence of the sensors during the measurement update are rearranged provides the possibility of another recovery method. A possibility is to change the order of the measurement updates depending on whether a sensor anomaly occurs. For instance if a sensor is predicted as anomalous, the sensor can be implemented first in the measurement update. This can evolve into an anomaly detection and isolation architecture where instead of predicting which sensor is anomalous a probability of anomaly for each sensor is provided. The sensor with the highest probability of being anomalous is implemented first in the measurement update and the sensor with the lowest probability is implemented last. This is, however, added to future work since this significantly changes the current anomaly detection and isolation architecture and design.

7.3. Robustness of Recovery Methods

The performance of the recovery methods are highly dependent on the prediction accuracy of the detection and isolation methods. The *EKF-ignore* will most probably perform worse the less accurate the prediction is. If the prediction is inaccurate the *EKF-ignore* method will include the anomalous sensor measurement in the EKF measurement update or exclude an accurate sensor measurement from the update. There is no way of "resetting" the EKF after this if the *EKF-ignore* method fails. The *EKF-reset* on the other hand will most likely perform better the less accurate the prediction is (up to a point). This is because the *EKF-reset* method will basically re-implement the EKF based on a buffer of sensor measurements and states and exclude the incorrect sensor during that *reset*. The *EKF-combination* will most probably perform in between these two methods as it is combination of the two. To analyse the robustness of each recovery method the estimation metric of each method is provided for 30 orbits with specific prediction accuracies varying from 50% to 100%. The analysis will also be done on the Sun reflection anomaly. It must be noted that at very low prediction accuracies (near to 0%) the implementation of the *EKF-reset* will be as if the sun sensor is almost never included in the measurement update of the EKF. This is because the Sun reflection anomaly does not occur frequently.

7.3.1. *EKF-ignore*

The analysis of the robustness of the *EKF-ignore* method is provided for varying prediction accuracies. The result in Figure 7.5 are as expected and the *EKF-ignore* method performs reasonably from 100% prediction accuracy up until 90%. This demonstrates that the *EKF-ignore* method requires accurate detection and isolation methods to perform well and recover the EKF from anomalous sensor measurements. This is because every data sample that is inaccurately predicted influences the EKF and the *EKF-ignore* method does not account for these inaccurate predictions.

At a prediction accuracy of less than 60% the estimation metric of the *EKF-ignore* method is performing worse than the EKF would've performed without any recovery method, when compared to "None" in Figure 7.1 after 30 orbits. This is because the *EKF-ignore* method not only incorporates the anomalous sensor measurement during incorrect predictions, but also excludes the normal sensor measurements if the prediction is inaccurate.

7.3.2. *EKF-reset*

The *EKF-reset* method does not ignore sensor measurements, but "resets" the EKF every single time the prediction shifts from normal to anomalous or the other way around. The results for the robustness of the *EKF-reset* method at varying prediction accuracies are

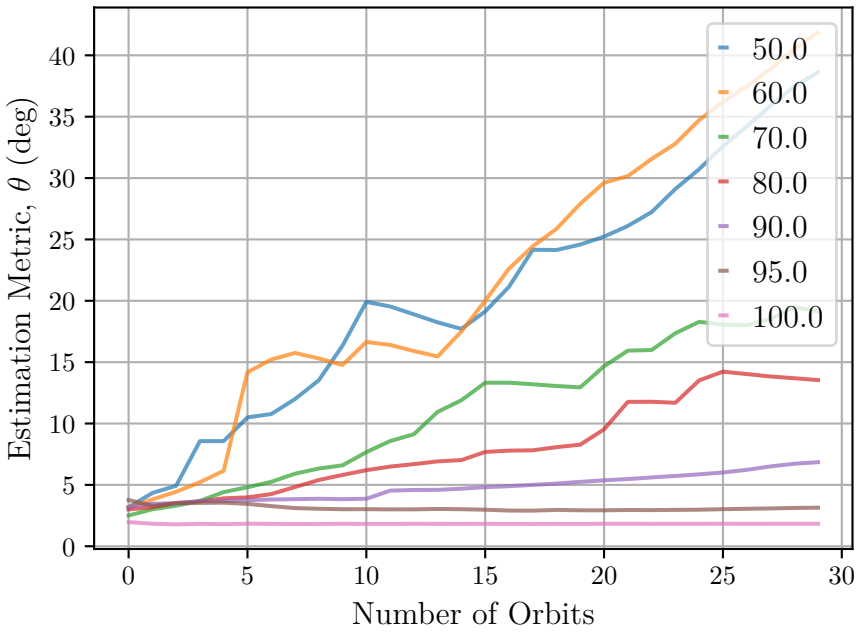

Figure 7.5: Comparison of the average estimation metric per orbit for 30 orbits for the *EKF-ignore* method with varying prediction accuracies during the Sun reflection anomaly. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

provided in Figure 7.6. As expected the *EKF-reset* reduces the estimation metric the lower the prediction accuracy. This, however, will only be up until a certain point and the *EKF-reset* method is time consuming since it must reiterate through the entire buffer every single time an anomaly is predicted.

If the *EKF-reset* method is continually implemented with a 0% prediction accuracy it will be equivalent to ignoring the sun sensor for every time step. This, however, does not take into account that this analysis is implemented for varying detection accuracies, but not varying isolation accuracies. Varying isolation accuracies will ignore different sensors at different time steps and the isolation accuracy will therefore also have a significant impact on the *EKF-reset* method. The *EKF-reset* method does not, however, perform with an estimation metric of less than 10° at a 50% detection accuracy. The analysis of varying isolation accuracies is therefore not required since that will only increase the estimation metric which is already not acceptable.

7.3.3. *EKF-combination*

The *EKF-combination* method is developed to try and get the best of both the *EKF-reset* and *EKF-ignore* method. The results for the robustness of the *EKF-combination* method is provided in Figure 7.7. Comparing the results with that of the *EKF-reset* and the *EKF-ignore* methods it can be concluded that the *EKF-combination* method (without varying isolation accuracies) outperforms both methods at prediction accuracies between

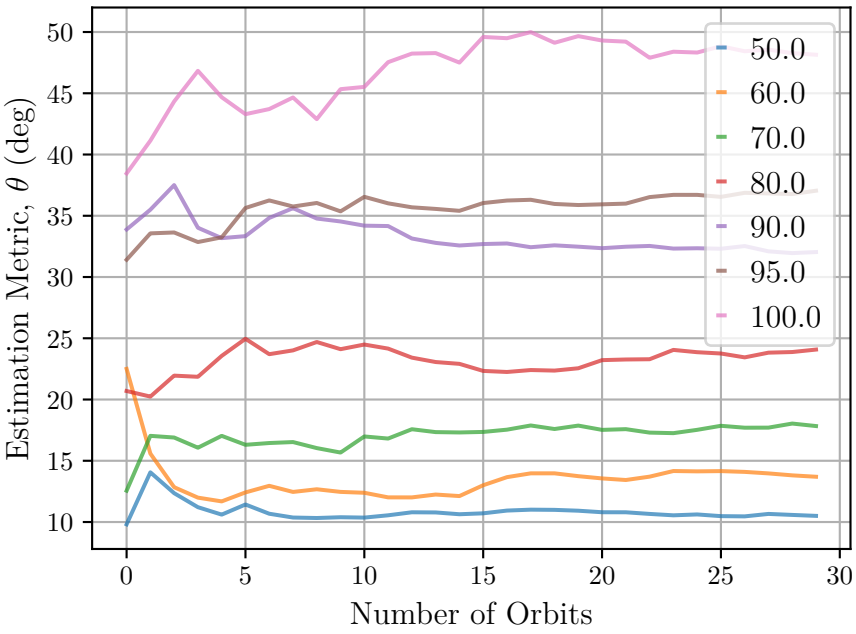

Figure 7.6: Comparison of the average estimation metric per orbit for 30 orbits for *EKF-reset* method with varying prediction accuracies during the Sun reflection anomaly. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

80% and 50%. The *EKF-combination* reduces the estimation metric to less than 8° after 30 orbits within that range of prediction accuracies. At high prediction accuracies the *EKF-combination* method performs better than the *EKF-reset* method since it does not "reset" the EKF as much. It does not, however, perform as well as the *EKF-ignore* method at prediction accuracies between 80% and 100%.

The recovery method implemented therefore depends on the anomaly detection and isolation accuracies. The *EKF-combination* will be implemented with prediction accuracies less than 80%. It is, however, concluded that with high prediction accuracies the *EKF-ignore* method will be implemented, which seems to be the case given the analysis done in Chapter 6. If the prediction accuracies are lower during the final analysis, the analysis might require implemented the *EKF-combination* method.

7.4. Summary

From both Figure 7.1 and Figure 7.2 it is clear that the *EKF-ignore* method outperforms the other recovery methods. This is implemented by only excluding the sensor measurement with an anomaly from the EKF measurement update. It is also derived from this chapter that the sequence in which the sensor updates occur influences the EKF and also the effect of the recovery method. The *EKF-ignore* method will probably be the best recovery method given the prediction accuracies of the anomaly detection and isolation methods in

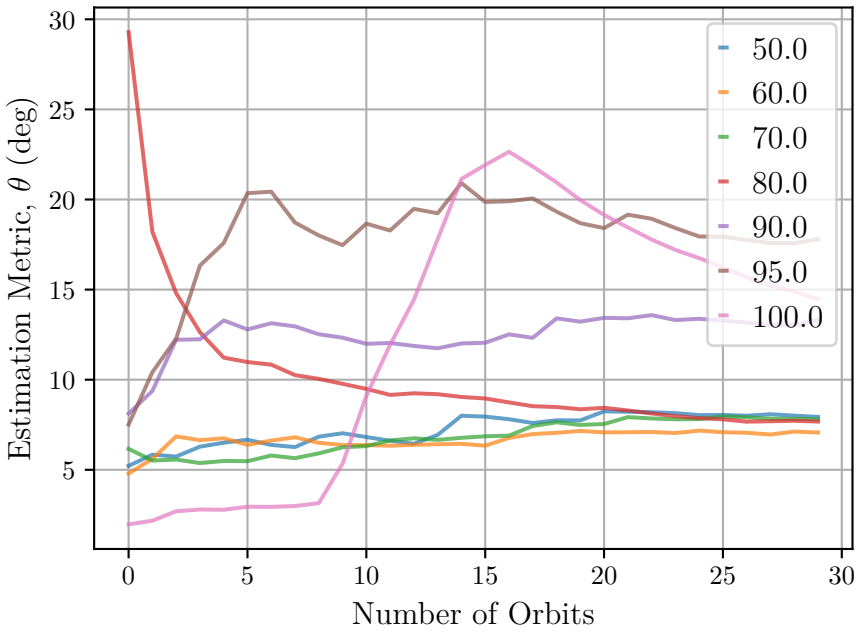

Figure 7.7: Comparison of the average estimation metric per orbit for 30 orbits for *EKF-combination* method with varying prediction accuracies during the Sun reflection anomaly. The estimation metric is the absolute difference between the attitude in quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$ in degrees.

Chapter 6. This is concluded from the analysis of the robustness of the recovery methods at varying prediction accuracies. This chapter provides the detailed analysis of the recovery block in Figure 1.2. Each block in Figure 1.2 was discussed and analysed in the previous chapters and the entire system can now be analysed as a whole.

Chapter 8

Results of Fault Detection, Isolation and Recovery Techniques

Each category within FDIR can be chosen based on the sequential analysis of each category independently. Combining the methods and closing the loop can change the results as the recovery method might change the response of the control output which in turn might change the behaviour of both the detection and isolation algorithms. This chapter provides the results for the entire system block diagram as shown in Figure 1.2. The different anomaly detection and isolation methods were previously analysed in Chapter 6 with the specific focus on the Sun reflection and magnetic moment disturbance anomalies. The integrated system of recovery, detection, isolation as well as feature extraction is further analysed in this chapter to determine the most suitable combination of methods.

In this chapter the results of the prediction accuracy for the isolation methods is provided and analysed with the addition of the feature extraction methods. The best method according to the analysis is thereafter also compared with small variations such as with and without detection and the time efficiency of the method. The methods are then compared to the traditional technique of training the FDIR method on only a general anomaly, in this case high sensor noise. The method will then be implemented and tested on the all the anomalies as discussed in Chapter 5 excluding the Moon on the Earth horizon anomaly.

8.1. Comparison of Feature Extraction Methods

To determine the best performance of each detection and isolation method the outputs of the feature extraction methods are provided as additional input to the detection and isolation methods. This excludes the LOF algorithm for detection since it is already considered as a feature extraction method and will not be tested with additional features as input. The prediction methods — decision tree, random forest and SVMs — are trained with either the outlier score O, as the output of the LOF algorithm, or the moving average $\mathbf{V}_k$, as the output of the MAESM algorithm, as additional inputs. The feature extraction methods are trained beforehand on the normal operation of the satellite (without

anomalies) and thereafter the output of the feature extraction methods for both anomalous and normal data is provided as input to the prediction methods.

The confusion matrices of the detection methods with the moving average $\mathbf{V}_k$ as additional input are provided in Table 8.1. If compared to Table 6.3, Table 6.5 and Table 6.7 it can be noted that the prediction accuracy of both the decision tree and the random forest have increased, while the prediction accuracy of the SVMs algorithm has decreased. The prediction accuracy of the random forest algorithm has increased from 99.95% to 99.997% and the prediction accuracy of the decision tree algorithm has increased from 99.87% to 99.95%. This demonstrates the enhancement of the the moving average in the prediction accuracy of the decision tree and random forest algorithm as is evident in the research done by de Silva et al. [21]. The prediction accuracy of the SVMs algorithm has, however, dramatically decreased from 98.13% to 93.54%.

	Random forest	
	Normal	Anomaly
Normal	**158955**	0
Anomaly	6	**45477**

(a) Random forest as detection.

	Decision tree	
	Normal	Anomaly
Normal	**158938**	17
Anomaly	40	**45443**

(b) Decision tree as detection.

	SVMs	
	Normal	Anomaly
Normal	**146944**	12011
Anomaly	1182	**44301**

(c) Support vector machines as detection.

Table 8.1: Confusion matrices for anomaly detection of the decision tree, random forest and SVMs trained with the moving average $\mathbf{V}_k$ as additional input on a test data set. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples.

The detection algorithms are now expanded to also perform isolation and classify the most probable cause. The confusion matrices are provided in Table 8.2 to determine the performance of the isolation methods with the moving average as additional input. The decision tree and random forest algorithms prediction accuracy increases as in the previous case. The prediction accuracy of the decision tree increases from 99.88% to 99.97% and the prediction accuracy of the random forest increases from 99.95% to 99.99%. The prediction accuracy of the SVMs algorithm decreases as with the anomaly detection. The prediction accuracy decreases from 98.09% to 94.49% with the moving average as additional input. This is most probably due to the SVMs algorithm increasing the dimensionality of an input feature space and since the moving average $\mathbf{V}_k$ is an additional 9×9 matrix, the feature space might be too large and the SVM provides a hyperplane that is not as accurate due to the additional feature.

The confusion matrices of the detection methods with the additional input of the

	Random forest		
	Normal	Sun	Mag
Normal	**158955**	0	0
Sun	18	**2904**	0
Mag	3	0	**42558**

(a) Random forest as isolation.

	Decision tree		
	Normal	Sun	Mag
Normal	**158938**	9	8
Sun	32	**2890**	0
Mag	15	0	**42546**

(b) Decision tree as isolation.

	SVMs		
	Normal	Sun	Mag
Normal	**147764**	3883	7308
Sun	51	**2861**	10
Mag	0	19	**42542**

(c) SVMs as isolation.

Table 8.2: Confusion matrix for anomaly isolation of the decision tree, random forest and SVMs algorithm trained with the moving average $\mathbf{V}_k$ as additional input on a test data set. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples. The magnetometer is abbreviated to "Mag" and the sun sensor is abbreviated to "Sun".

outlier score O, as output from the LOF algorithm, are provided in Table 8.3. This is the results based on the LOF algorithm with the contamination hyper parameter set to 0.1. Although all three algorithms have an increased prediction accuracy, the increase is quite small for both the decision tree and random forest with an increase of 0.01% for both methods. The SVMs algorithm on the other hand has an increased prediction accuracy of 98.13% to 98.87%. The outlier score definitely provides enhancement to the prediction accuracy of the detection method.

	Random forest	
	Normal	Anomaly
Normal	**158904**	51
Anomaly	26	**45457**

(a) Random forest as detection.

	Decision tree	
	Normal	Anomaly
Normal	**158829**	126
Anomaly	118	**45365**

(b) Decision tree as detection.

	SVMs	
	Normal	Anomaly
Normal	**156691**	2264
Anomaly	40	**45443**

(c) SVMs as detection.

Table 8.3: Confusion matrix for anomaly detection of the decision tree, random forest and SVMs trained with the outlier score O as additional input on a test data set. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples.

The confusion matrices for the isolation methods with the outlier score as additional input is given in Table 8.4. The change in performance of the isolation methods are similar

to that of the detection methods due to the additional input of the outlier score as feature. The prediction accuracy increases from 99.88% to 99.885% and the prediction accuracy of the random forest increases from 99.95% to 99.96%. This a very small difference compared to the increased prediction accuracy of the decision tree and random forest algorithm due to the moving average as additional input. The prediction accuracy of the SVMs algorithm for anomaly isolation increases from 98.09% to 98.81% with the outlier score as additional input. This demonstrates the different nature of the SVMs algorithm and the decision tree and random forest algorithms. The SVMs performs better with a single additional feature which increases the accuracy of the classification slice from the hyperplane, while the decision tree and random forest perform multiple splits on the data set and can make accurate predictions with a larger feature space.

	Random forest		
	Normal	Sun	Mag
Normal	**158902**	0	53
Sun	1	**2921**	0
Mag	27	0	**42434**

(a) Random forest as isolation.

	Decision tree		
	Normal	Sun	Mag
Normal	**158854**	13	88
Sun	22	**2900**	0
Mag	112	0	**42449**

(b) Decision tree as isolation.

	SVMs		
	Normal	Sun	Mag
Normal	**156515**	151	2289
Sun	1	**2921**	0
Mag	0	0	**42561**

(c) SVMs as isolation.

Table 8.4: Confusion matrix for anomaly isolation of the decision tree, random forest and SVMs algorithm trained with the outlier score O as additional input on a test data set. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples. The magnetometer is abbreviated to "Mag" and the sun sensor is abbreviated to "Sun".

From this section it can be concluded that the best results for the random forest and decision tree algorithms are with the moving average $\mathbf{V}_k$ as additional input and the best results for the SVMs algorithm is with the outlier score as additional input. This will therefore be the implementation of these methods with the incorporation of the recovery method. The method with the best isolation accuracy and detection accuracy is the random forest algorithm which will be implemented for further analysis with recovery.

8.2. Integration of the Recovery Method

Based on the analysis of all the methods, the prediction accuracy seems to suggest that the best method is to only have a single random forest performing anomaly detection and isolation with the additional feature $\mathbf{V}_k$ from the MAESM output. The recovery method

that will be implemented is that of the *EKF-ignore* method, since it outperforms the other recovery methods with prediction accuracies above 80% as seen in Chapter 7.

The random forest can be implemented with either the moving average $\mathbf{V}_k$ or the outlier score as additional input. From Section 8.1 it is clear that the best prediction accuracy of the random forest is with the additional moving average as input. To compare the random forest with and without the moving average as additional output the architecture is as shown in Figure 8.1. The *EKF-ignore* as recovery method is implemented if the final classification is not "normal".

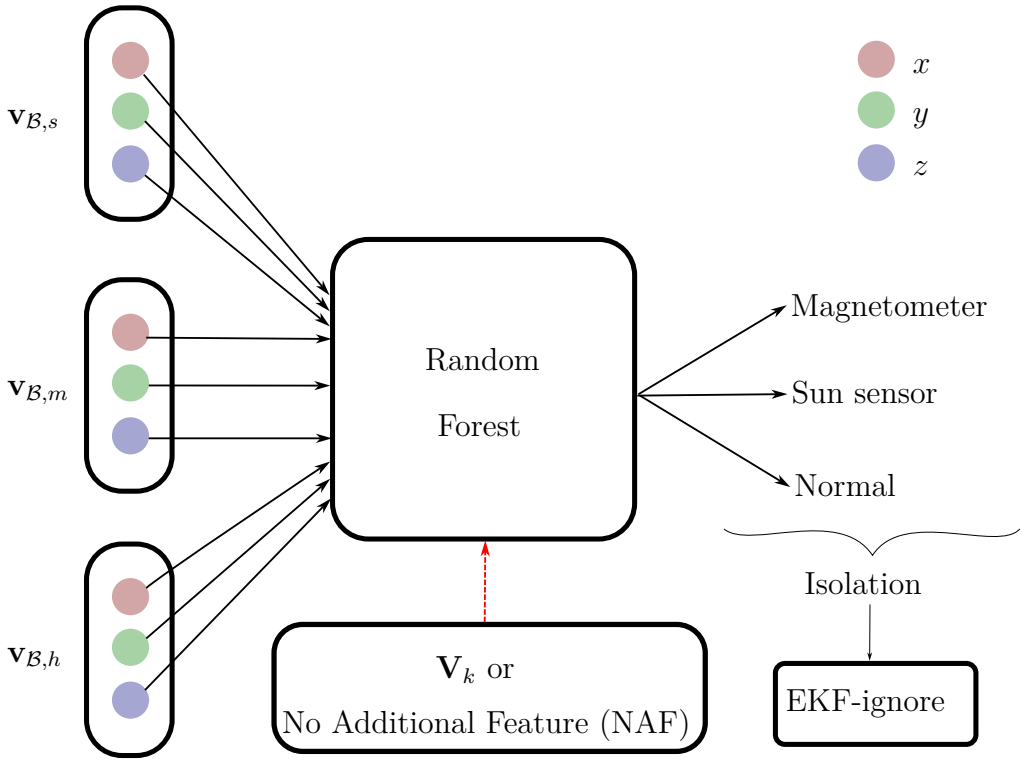

Figure 8.1: The random forest algorithm with an additional feature $\mathbf{V}_k$ as input or with no additional feature (NAF) as input. The random forest can classify the cause of the anomaly or reclassify the data sample as normal. The isolation triggers the *EKF-ignore* recovery method if the classification is not normal.

From both Figure 8.2 and Figure 8.3 it is clear that the random forest with no additional feature (NAF) performs well with the *EKF-ignore* method implemented. This demonstrates that the current FDIR method is able to detect, isolate and recover from the Sun reflection anomaly. It lowers the estimation metric to an average of less than 4° per orbit, which is acceptable. This means that no re-iteration of the analysis for the recovery methods and detection methods are required since there is a combination of methods that does perform well.

The moving average as additional input, however, does not perform as well as no additional input if the recovery method is also implemented. This is evident from both the estimation metric in Figure 8.2 and the isolation accuracy in Figure 8.3. The most probable reason for this is the influence that the recovery method has on the input features

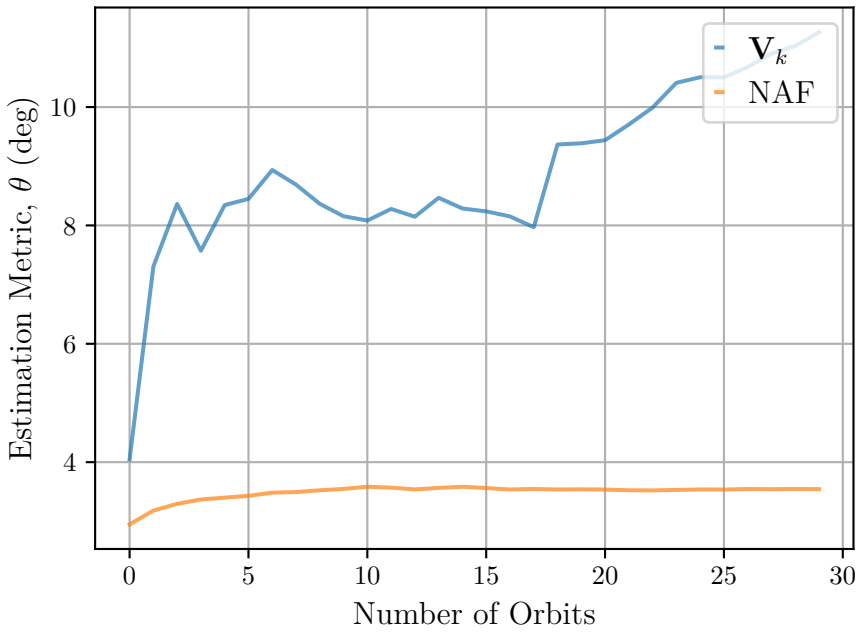

Figure 8.2: Comparison of average estimation metric per orbit of moving average $\mathbf{V}_k$ as additional input and no additional feature (NAF) to the random forest isolation method. This is implemented with the *EKF-ignore* method for recovery on the Sun reflection anomaly.

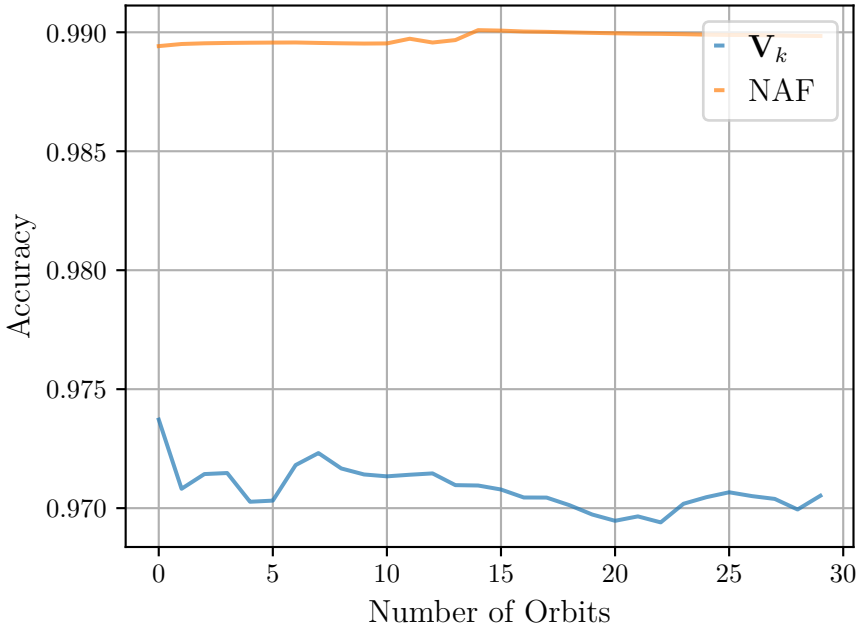

Figure 8.3: Comparison of average isolation accuracy per orbit of moving average $\mathbf{V}_k$ as additional input and no additional feature (NAF) to the random forest isolation method. This is implemented with the *EKF-ignore* method for recovery on the Sun reflection anomaly.

of the MAESM. The recovery method changes the control output torques and therefore influence the moving average $\mathbf{V}_k$ since the MAESM algorithm incorporates the control

torques with the **B** matrix in Equation 6.9, where $\mathbf{Y}_k$ is the control torques. The MAESM algorithm is a linear approximation of a non-linear system. When this system changes from the data that the linearisation was performed on, the results will differ. The random forest that performs prediction only on the sensor measurements as input is not affected by the changes in the moving average. Since the random forest predetermined the splits of the features, any changes from the training data might not be correctly predicted and the more the data deviates from the training data, the less accurate the predictions will be. It is therefore concluded that the changes of the system influences the linear approximation of the MAESM algorithm which in turn influences the accuracy of the trained random forest model.

The random forest without any additional features as input can now be compared to the scenario with and without an additional detection method. An additional detection method can be implemented to increase the computational efficiency of the FDIR method. This is demonstrated in Figure 8.4 where the data sample is either pass directly to the random forest model or it is passed to the anomaly detection model which triggers the random forest isolation model when it detects an anomaly. This trained random forest model for isolation includes the classification of normal data. LOF will be implemented as the anomaly detection method, since it has a very low false negative value as seen in Table 6.2 and has a faster execution time than the random forest algorithm. These two classification architectures will therefore be analysed.

The LOF algorithm is only implemented to reduce the average time spent on prediction, since the random forest can reclassify any data sample as normal. The estimation metric of the two methods is provided in Figure 8.5. The additional LOF algorithm for detection performs just as well as the random forest without any additional detection algorithm. Both architectures perform do not perform as well as 100% prediction in Figure 7.5 since the average estimation metric for all 30 orbits are 1.83°. This is due to the low false negative value of the trained LOF model. Most of the data samples that are anomalous are passed to the random forest model and a large percentage of normal samples as well. The normal samples that are passed to the random forest can be reclassified as normal even though the LOF algorithm predicted that the data sample is anomalous.

Comparing the timing of the two methods demonstrated that the LOF as detection method reduces the prediction time by 38.5% after 30 orbits. This, however, is still slower than either the decision tree's or SVMs' isolation time as given in Table 6.9b. The time, however,will decrease with a smaller contamination parameter for LOF algorithm, since this will decrease the number of false positives and increase the number of false negatives. Depending on the time constraint the contamination parameter can be fined tuned by taking into account the various design factors.

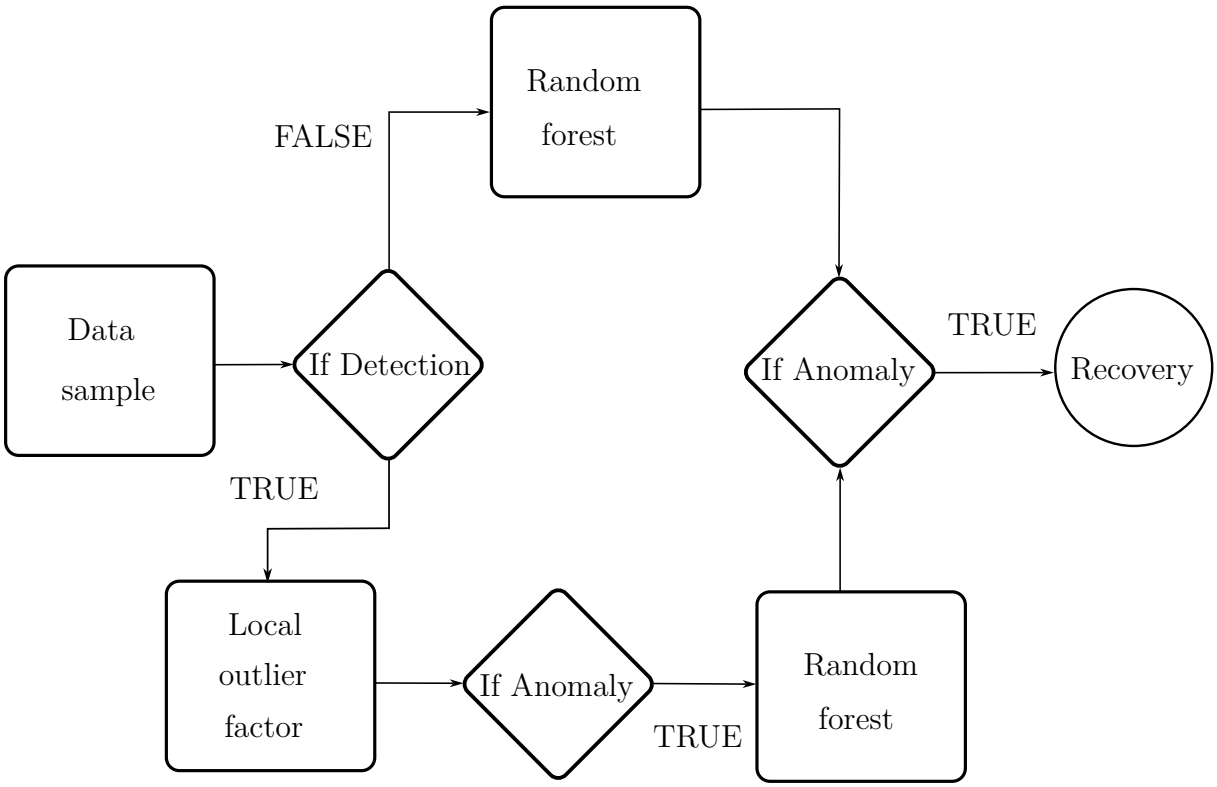

Figure 8.4: The two different architectures for classifying a data sample. Either the isolation is triggered with a detection algorithm or the isolation method performs both the detection and the isolation continuously.

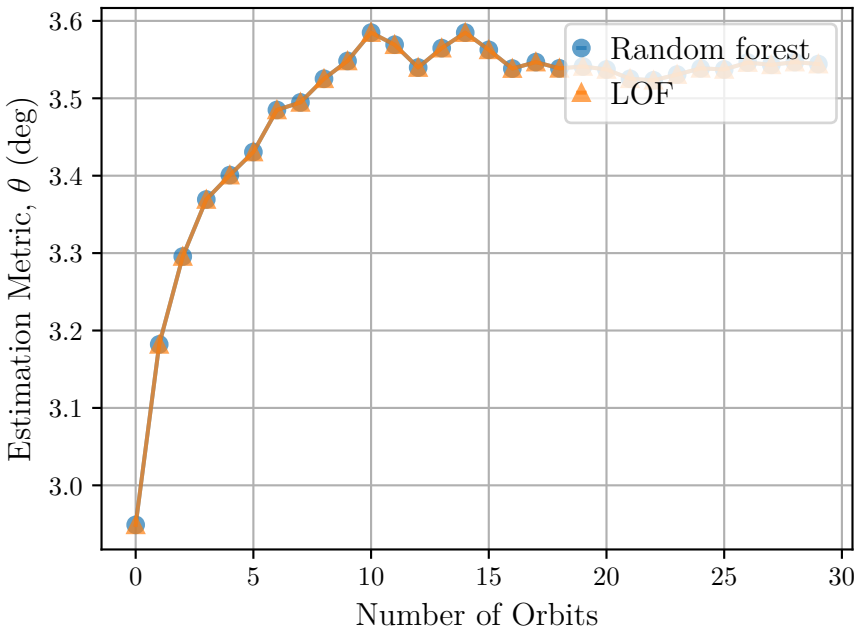

Figure 8.5: Comparison of average estimation metric per orbit of LOF as detection method or without any additional detection method to the random forest isolation method. This is implemented with the *EKF-ignore* method for recovery.

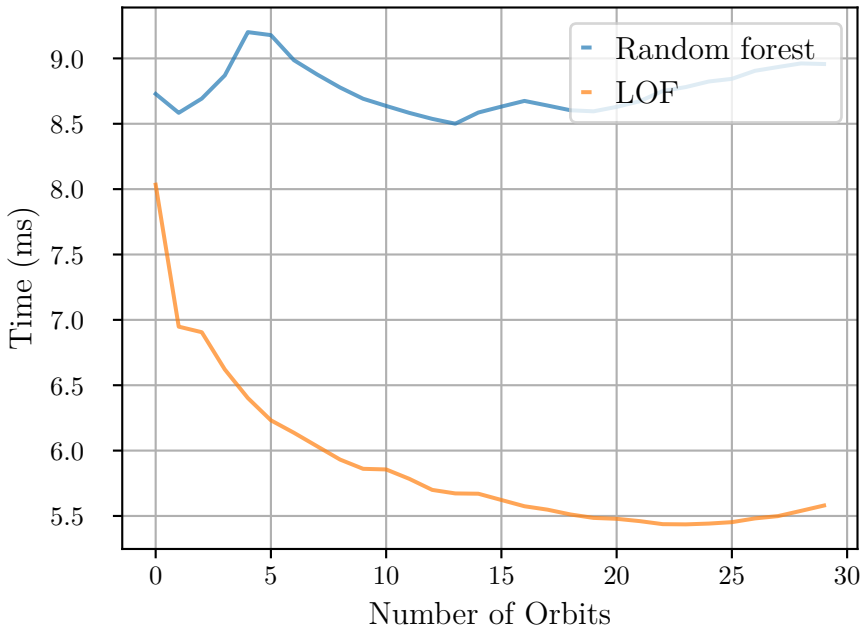

Figure 8.6: Comparison of average prediction time for each time step per orbit of LOF as detection method and the random forest isolation method without any additional detection method. This is implemented with the *EKF-ignore* method for recovery.

8.3. Fault Detection and Isolation on General Actuator Anomalies

As discussed in Chapter 2 the actuators can influence the model update of the EKF, since the control torques and the actual torques can differ. The actuator anomalies also require fault detection, isolation and recovery. The design decision is to implement two separate isolation models for actuator and sensor failures. The reason for this is due to the different inputs that might be required for isolating which actuator is experiencing an anomaly, compared to that of sensor anomalies. These inputs can for instance include the measured angular momentum of the reaction wheels. The actuator anomaly will also keep the correlation between the sensors the same. Including the control torques as input to the random forest might aid in the isolation of the anomaly, since the correlation between the control torques and the sensor measurements will change if the actual torques differ from the control torques. This also simplifies the implementation. The recovery method for actuator failure will be different to that of sensor failure. This design decision is made to ensure that the FDIR techniques of the sensor and actuators can be developed independently. This is because these two anomaly groups might require different techniques due to the nature of the anomalies.

For the isolation of the reaction wheels a random forest model is trained on the sensor and control torques as input to classify which reaction wheel is experiencing the anomaly as

well as classifying it as normal. The LOF algorithm is implemented differently for actuator anomalies and has the control torques as additional input. The FDIR for actuators and the FDIR for sensors can therefore operate separately. The random forest for actuator anomalies has four output classes, where three classes are the reaction wheel responsible for the anomaly and one is a class for normal data samples. The reaction wheel anomalies are implemented from the first orbit without any recovery. Three separate tests are conducted, one for each reaction wheel failure. The results for the anomaly detection accuracy of each wheel is provided in Figure 8.7 and the isolation accuracy is provided in Figure 8.8.

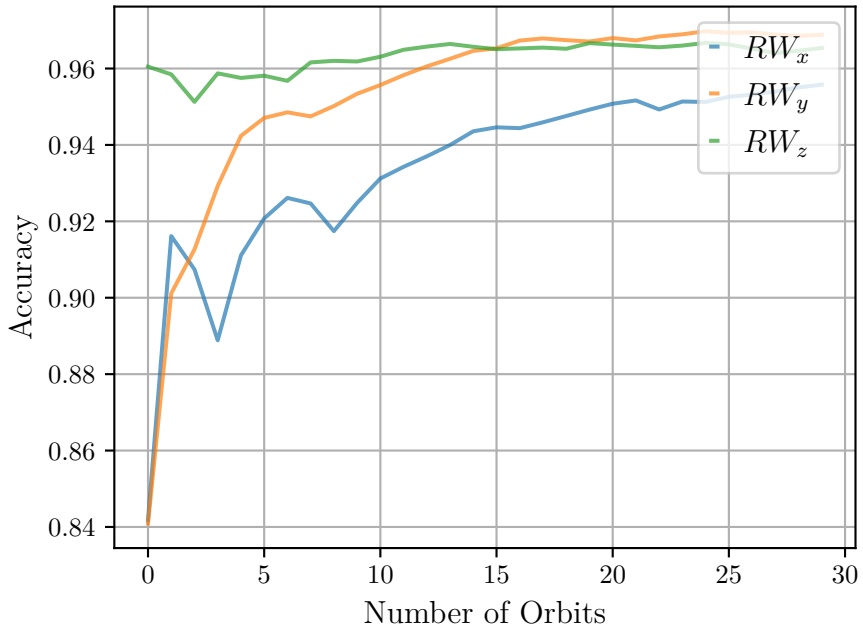

Figure 8.7: Average prediction accuracy per orbit of LOF as detection method and random forest as isolation method for reaction wheel failure.

The prediction accuracy and isolation accuracy for each reaction wheel is above 95% and 98.8% respectively after 30 orbits. This demonstrates that the proposed method for fault detection and isolation performs well on the simulated reaction wheel anomalies. The estimation metric, however, does not improve with the proposed recovery method of only adapting the input torque to the EKF model update based on the predicted actuator failure. This demonstrates that if the actuators fail that the control law should also be adapted to ensure robustness of the EKF.

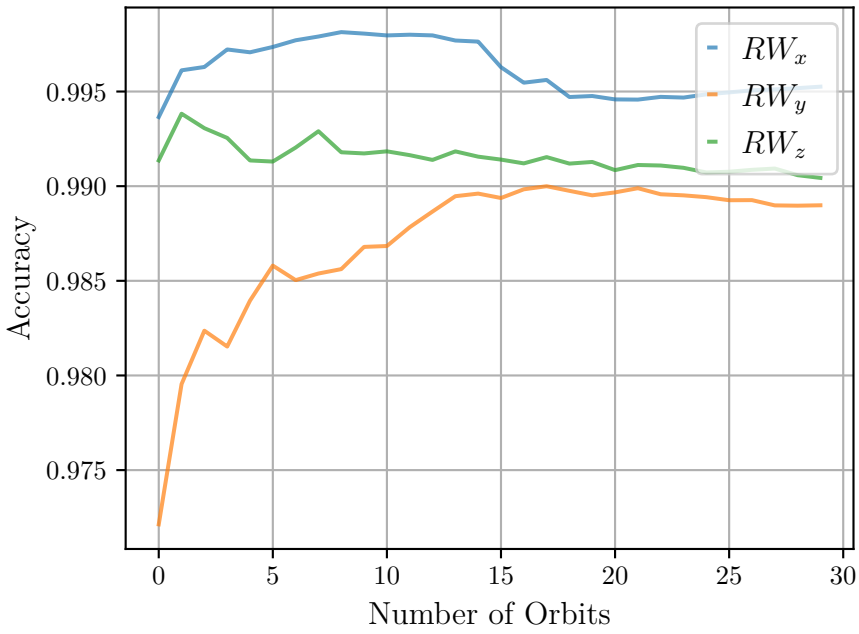

Figure 8.8: Average isolation accuracy per orbit of LOF as detection method and random forest as isolation method for reaction wheel failure.

8.4. Fault Detection, Isolation and Recovery on General Sensor Anomalies

This section provides the performance of the random forest algorithm for isolation of general anomalies. This will be the high noise of both the magnetometer and sun sensor as discussed in Section 5.4. The random forest algorithm can be trained and implemented in three ways as demonstrated in Figure 8.9. The one is to group all anomalies of a sensor into one group. This would mean that instead of classifying an anomaly, the algorithm would only classify the sensor that is providing anomalous measurements. The other would be to classify each anomaly individually and to determine the sensor based on the anomaly. The last method is to train the random forest only on general sensor anomalies, which in this case is high noise.

To abbreviate the anomalies and summarise the results, the orbit sensitive practical modelled anomalies such as the Sun reflection and magnetic moment disturbance are abbreviate with (P), while the general anomalies of high noise are abbreviated with (G). The confusion matrix for the random forest trained on the general and practical anomalies are provided in Table 8.5. The overall prediction accuracy of the model is 97.43%. This is not the prediction accuracy of 99.9%+ with which the random forest performed on only the orbit sensitive practical anomalies. It can also be noted that the random forest's prediction accuracy is the lowest when predicting either the high noise of the sun (Sun G) or the high noise of the magnetometer (Mag G). This therefore demonstrates how a specific

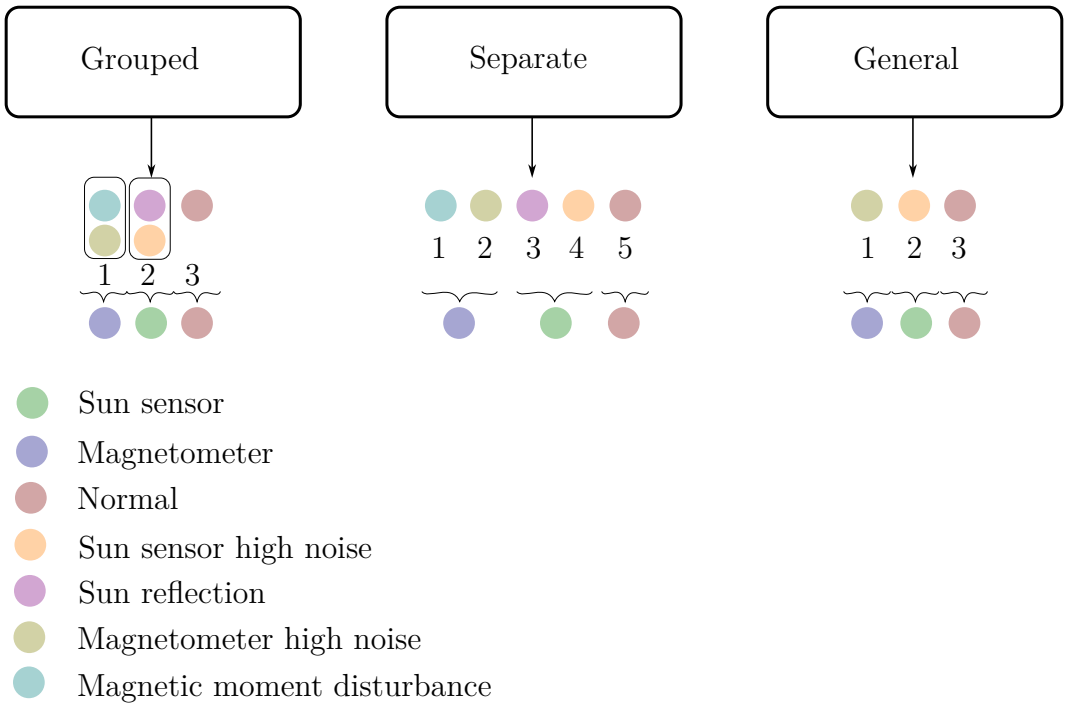

Figure 8.9: The three different ways of training and classification of the random forest algorithm for isolation. The random forest algorithm can classify data sample either as Grouped, Separate or General. The output is always the anomalies on which the random forest is trained and the anomalous sensor can be determined depending on the classified anomaly.

method does not perform as well with the classification of different types of anomalies. A more appropriate method for detecting high noise could be to perform waveform analysis with methods like applying the Fourier transform. These waveform analysis methods, however, might perform badly due to the orbital nature of the satellite with regards to eclipse and sunlit phases.

	Random forest				
	Normal	Sun P	Mag P	Sun G	Mag G
Normal	**182859**	0	53	119	1388
Sun P	0	**2917**	0	5	0
Mag P	38	0	**42518**	0	5
Sun G	385	6	11	**41910**	216
Mag G	6391	0	36	94	**61760**

Table 8.5: Confusion matrix for random forest algorithm trained on both general (G) and orbit sensitive practical modelled (P) anomalies. The anomalies are separated into a class per anomaly. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples.

The other method of isolation is grouping the anomalies into a single group based on the sensor responsible for the independent anomalous measurement. This leads to an overall prediction accuracy of 97.47%. This is slightly better than the prediction accuracy

of each anomaly in its own class. Both methods will be implemented with the *EKF-ignore* as the recovery method to determine the performance thereof later in this section.

	Random forest		
	Normal	Sun	Mag
Normal	**182704**	132	1583
Sun	388	**44800**	262
Mag	6175	96	**104571**

Table 8.6: Confusion matrix for random forest algorithm trained on both general and orbit sensitive practical modelled anomalies. The anomalies are grouped into a class per sensor for the sun sensor (Sun) or the magnetometer (Mag). The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples.

To determine the prediction accuracy of the random forest algorithm on only high noise the confusion matrix thereof is provided in Table 8.7a with an overall prediction accuracy of 97.02%. In the field of FDIR there is many research that provide results of developed methods based on general anomalies only. Table 8.7b provides the confusion matrix of a model trained on general anomalies only and implemented on orbit sensitive practical modelled anomalies. The prediction accuracy thereof is 70.62% which demonstrates how this specific model does not perform well on the orbit sensitive practical modelled anomalies and this could be the same for other proposed methods in the FDIR field that are only tested on general anomalies.

	Random forest		
	Normal	Sun G	Mag G
Normal	**92290**	26	1197
Sun G	140	**42406**	309
Mag G	4329	99	**63642**

(a) Random forest on general anomalies.

	Random forest		
	Normal	Sun P	Mag P
Normal	**117144**	18979	22832
Sun P	0	**2815**	107
Mag P	4	18145	**24412**

(b) Random forest on practical anomalies.

Table 8.7: Confusion matrix for random forest algorithm trained on only the general (G) anomalies. The same trained model is implemented on both the general (G) and orbit sensitive practical modelled (P) anomalies. The row labels are the actual conditions of the data samples, while the column labels are the classified condition of the data samples.

It must be noted that even though the methods perform well in the analysis, that when the methods are implemented along with the recovery, that the estimation metric is high as seen in Figure 8.10. This is due to the sensitivity of the recovery to the prediction accuracy of the isolation method. The prediction accuracy of 97%+, means that 3% of the time either an anomalous data sample is predicted as normal or even worse when the isolation method predicts that another sensor is responsible for the anomalous measurement. This then not only incorporates the anomalous measurement in the EKF measurement update,

but also excludes correct sensor measurements from the sensor that is performing as expected.

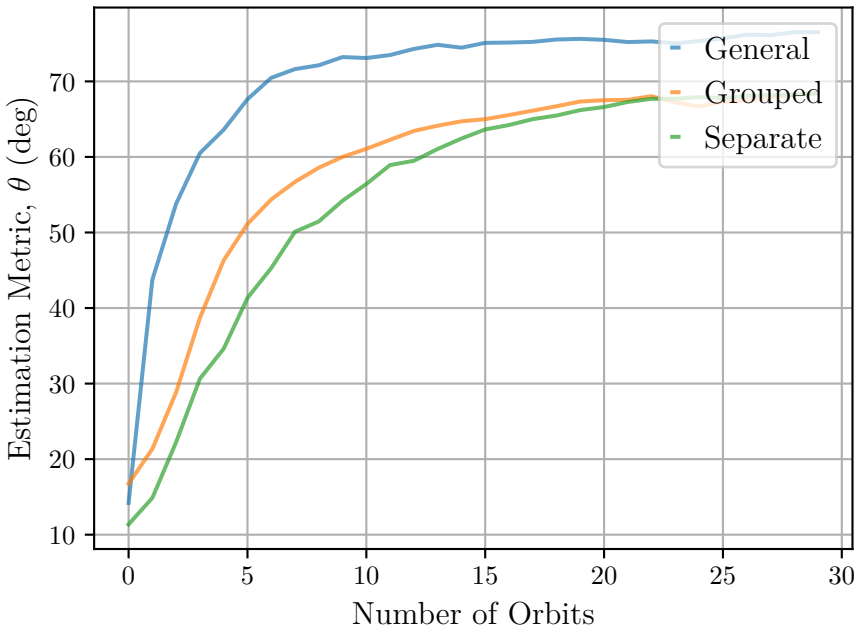

Figure 8.10: Comparison of average estimation metric per orbit of the different training methods of the random forest algorithm as isolation. The general class refers to the random forest trained only on high noise anomalies, the grouped class is that of training on all the anomalies grouped per sensor responsible for the anomalous measurement. The separate class is the random forest trained on each anomaly as an individual class. The LOF algorithm is implemented as the detection method and *EKF-ignore* is implemented as the recovery method.

This leads to the conclusion that although the random forest performs well on the orbit sensitive practical modelled anomalies it is not the best method to classify high noise. A combination of isolation methods could be implemented to isolate various types of anomalies instead of relying on one method that is more applicable for certain anomaly types. It is also clear that training a model on a specific type of anomaly and then implementing that trained model on another type of anomaly does also not guarantee good performance. The random forest algorithm which is trained on high noise and then implemented on the Sun reflection anomaly provided bad results as is seen in Figure 8.10. This demonstrates that FDIR testing should be conducted on both general anomalies as well as practical anomalies, since the methods that perform well on one group will not necessarily perform well on the other.

8.5. Summary

In this chapter the results for variations of the FDIR methods and algorithms were discussed and analysed. Based on the analysis it was derived that random forest has the

highest prediction accuracy for anomaly isolation. A comparison was given for LOF as detection, random forest as isolation and *EKF-ignore* as recovery and variations of this. It is clear that the estimation metric decreases with the integration of the recovery method in Figure 8.5 where the estimation metric remains below 4°. This demonstrates that the FDIR technique is successful for the application of the Sun reflection anomaly.

The time efficiency of the random forest with and without LOF as detection was also discussed and it was clear that LOF decreases the time per prediction by 38.5%. The results for the fault detection and isolation method for the reaction wheel anomaly was briefly given and was only analysed on the prediction accuracy, since the controller also requires recovery, which is not within the scope of this thesis. The method was also compared to the training of the random forest on the specific anomalies as well as the high noise on the sensors. From the results it was concluded that the random forest performs significantly better when trained on only the orbit sensitive practical anomalies compared to the training on the high noise of the sensors. The prediction accuracy of the random forest for isolation, however, decreases and highly influences the estimation metric when the model was trained on both general and orbit sensitive practical anomalies. It was therefore concluded that a combination of isolation methods could be the best solution for FDIR in a system with diverse anomaly types.

Chapter 9

Conclusion

The ADCS of satellites is a crucial subsystem for the success of a satellite mission. The detection of faults and the recovery thereof is therefore required. While many research has been conducted on the actuator or control law failure of the ADCS, little research has been done on the attitude determination aspect of the ADCS. The sensor anomalies used in previous research, however, are general sensor anomalies and consists of bias drift, high noise, sudden failure or any drastic change in the sensor measurement. This does not necessarily ensure that the proposed methods provide accurate estimation during specific practical anomalies, such as the Sun reflection from the solar panels of the satellite unto the sun sensor.

9.1. Summary and Evaluation

There are two different categories within fault tolerant control — the one category is the recovery of the control law and actuator failure, while the other is the attitude determination of the ADCS. It was determined from previous research on FDIR for satellites, that the most attention was given to control law and actuator failure. The objective of this thesis was to develop a FDIR method for sensor anomalies to ensure robust EKF state estimation. Previous research on developing methods to ensure robust attitude determination are based on general sensor anomalies. The objective of this thesis was, however, based on specific practical sensor anomalies, that are orbit sensitive.

To fulfil this objective, a realistic simulation environment was developed in which the effect of the FDIR techniques on the EKF could be evaluated. This simulation environment was used to simulate a number of practical anomalies. The orbit sensitive anomalies are specific to each sensor and are due to natural occurrences and not due to a failure in the sensor itself. This makes the sensor inadequate at providing an accurate measurement from which the attitude can be determined. The Sun reflection from the solar panels unto the sun sensor is the most prominent anomaly due to the standard sequence of the sensor fusion algorithm. The Moon on the horizon of the Earth and in the FoV of the horizon sensor is accurately modelled, but the influence thereof is small due to the current orbit parameters. The magnetometer can be influenced by a dipole moment induced by the circuitry in the

solar panels. This will also create a disturbance torque. The influence of this anomaly is mostly due to the disturbance torque and not due to the magnetometer measurements. This is mostly due to the standard sequence of the sensor fusion algorithm, where the magnetometer is normally implemented as the first measurement in the measurement update of the EKF.

To enhance the performance of anomaly detection and isolation methods, extraction of features that provide additional information was investigated. The feature extraction was specifically designed to provide a feature that provides information on whether the current data sample was anomalous and which sensor was anomalous. The local outlier factor (LOF) algorithm provides a feature that offers information on whether the current data sample was anomalous within a local neighbourhood. The linear regression method, on the other hand, provided a moving average to increase the information gain on both the abnormality of a data sample and which sensor was anomalous.

The simulation environment provides realistic data for the training of anomaly detection and isolation algorithms. It was concluded, by comparing LOF, decision tree, random forest and SVM, that the random forest has the highest prediction accuracy, but it is also the least time and computation efficient. After detecting the anomaly, it is required to isolate the anomalous sensor. The supervised learning methods, namely decision trees, random forest and SVM were compared. The comparison was done within a simulation where the detection method always predicts an anomaly. The supervised learning methods are trained on both normal and anomalous data and could therefore classify a data sample as normal. It was also concluded that the random forest algorithm is the most accurate for anomaly isolation.

To develop a recovery method which ensures a robust EKF during an anomaly, different recovery methods were compared. It was concluded that the best recovery method under perfect conditions is the *EKF-ignore* method which only removes an anomalous sensor from the measurement update of the EKF. The *EKF-ignore* provides a result similar to an estimation where no anomaly is present if the prediction accuracy if higher than 90% for the Sun reflection anomaly.

Different detection and isolation methods were analysed with the additional features as input. It was, however, concluded that the random forest algorithm performs the best without any additional feature from the feature extraction methods as input when the recovery method is also implemented. The proposed FDIR method was thus LOF as detection, random forest as isolation, and *EKF-ignore* as the recovery without any feature extraction method as input to the random forest algorithm.

The method was thereafter analysed on the reaction wheel failures where it was concluded that the proposed method can accurately predict reaction wheel failure and isolate the wheel that failed. This was implemented without recovery, since the recovery required changes in the controller which is outside the scope of this thesis. Implementing

the method on the high noise anomalies demonstrated that different methods perform well on different types of anomalies. The random forest can accurately predict high noise, but not accurately enough to ensure a robust Kalman filter. This was demonstrated in the increased estimation metric of the model trained of the high noise data samples. It was therefore proposed that a combination of isolation methods might be better to classify different types of anomalies.

9.2. Observations and Findings

Many observations and findings were made from the results in this thesis. The contribution is therefore summarized and given as:

1. Implementing a random forest model trained on practical sensor anomalies and ignoring a sensor during the EKF measurement update step when a sensor is predicted as anomalous, restores the estimation stability and accuracy of the EKF.

2. The random forest algorithm does not perform well in the isolation of high noise as the anomaly. There is most likely not a single method that will perform well in classifying all anomalies. An architecture where multiple methods are used simultaneously to isolate the anomaly is proposed.

3. Training a model on high noise of a sensor performs badly when implemented on a practical anomaly. Testing proposed FDIR methods on practical anomalies is required, since there is a substantial behavioural difference between practical anomalies and general anomalies.

4. Rearranging the order of sensors in the measurement update can significantly change the impact of an anomaly on the EKF. This leads to another possible recovery method which only changes the order in which the sensors are used in the EKF measurement update. This is, however, suggested for future work due to the change that this has on the architecture of the anomaly detection and isolation.

9.3. Future Work

The observations and findings also lead to many recommendations and improvements for the FDIR of the ADCS of satellites. These recommendations are for future work and are summarized as:

1. The Sun reflections can be modelled for different influences on the sun sensor. Currently the sun sensor latches unto the reflected sun vector, but this will not be the case for all sun sensors.

2. Research and development of models for more anomalies of each sensor can be conducted. The isolation will still have to classify the same number of classes, since it is only required to classify the anomalous sensor and not the specific anomaly. This, however, depends on whether the anomaly requires control law recovery as well. An important evaluation is to determine the performance of the FDIR technique on sensor anomalies that are not part of the training dataset.

3. Develop recovery methods for anomalies that influence the controller as well as the EKF. The magnetic moment disturbance anomaly and the reaction wheel failure are both examples of anomalies that require recovery of the controller.

4. Develop anomalies for the magnetorquer to demonstrate the recovery law implementation for more than one actuator type.

5. High noise anomalies can be analysed with a Fourier transform and a combination of isolation methods can be developed to predict different types of anomalies.

6. Multiple sensors and actuators can experience anomalies simultaneously and it is necessary to develop FDIR techniques that are robust against this.

Implementation of this list and evaluating the results thereof can provide valuable insight in the FDIR of the ADCS. The fault tolerant control of satellites can be improved based on further research and determining the effect of multiple anomalies per sensor on the FDIR method provided. This thesis provides the first steps to gaining insight in the influence of training the FDIR techniques on orbit sensitive anomalies.

Bibliography

[1] M. Tafazoli, "A study of on-orbit spacecraft failures," *Acta Astronautica*, vol. 64, no. 2-3, pp. 195–205, 2009.

[2] S. A. Jacklin, "Small-satellite mission failure rates," 2019.

[3] M. Swartwout, "Cubesat database," *URL: https://sites. google. com/a/slu. edu/swartwout/home/cubesat-database (visitado 12-06-2022)*, 2015.

[4] H. E. Soken, C. Hajiyev, and S.-i. Sakai, "Robust kalman filtering for small satellite attitude estimation in the presence of measurement faults," *European Journal of Control*, vol. 20, no. 2, pp. 64–72, 2014.

[5] M. Wang and T. Liang, "Adaptive kalman filtering for sensor fault estimation and isolation of satellite attitude control based on descriptor systems," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 6, pp. 1686–1698, 2019.

[6] K. Xiong, C. Chan, and H. Zhang, "Detection of satellite attitude sensor faults using the ukf," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 2, pp. 480–491, 2007.

[7] J. Zhou, X. Li, R. Liu, and Y. Liu, "A scheme of satellite multi-sensor fault-tolerant attitude estimation," *Transactions of the Institute of Measurement and Control*, vol. 38, no. 9, pp. 1053–1063, 2016.

[8] S. S. Nasrolahi and F. Abdollahi, "Sensor fault detection and recovery in satellite attitude control," *Acta Astronautica*, vol. 145, pp. 275–283, 2018.

[9] J. Carvajal-Godinez, J. Guo, and E. Gill, "Agent-based algorithm for fault detection and recovery of gyroscope's drift in small satellite missions," *Acta Astronautica*, vol. 139, pp. 181–188, 2017.

[10] L. Van Eykeren, Q. Chu, and J. Mulder, "Sensor fault detection and isolation using adaptive extended kalman filter," *IFAC Proceedings Volumes*, vol. 45, no. 20, pp. 1155–1160, 2012.

[11] G. Zhiwei, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part ii: Fault diagnosis with knowledge-based and hybrid/active approaches," 2015.

[12] B. Pilastre, L. Boussouf, S. d'Escrivan, and J.-Y. Tourneret, "Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning," *Signal Processing*, vol. 168, p. 107320, 2020.

[13] U. Feucht, M. Saliya, and M. Subramanian, "Knowledge based analysis of a satellite navigation system," in *Proceedings of IEEE Systems Man and Cybernetics Conference-SMC*, vol. 4. IEEE, 1993, pp. 241–246.

[14] E. Sobhani-Tehrani and K. Khorasani, *Fault diagnosis of nonlinear systems using a hybrid approach*. Springer Science & Business Media, 2009, vol. 383.

[15] D. Liu, H. Zhen, D. Kong, X. Chen, L. Zhang, M. Yuan, and H. Wang, "Sensors anomaly detection of industrial internet of things based on isolated forest algorithm and data compression," *Scientific Programming*, vol. 2021, 2021.

[16] O. Ciftcioglu and E. Turkcan, "Sensor failure detection in dynamical systems by kalman filtering methodology," Netherlands Energy Research Foundation, Tech. Rep., 1991.

[17] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, 2016.

[18] A. Rahimi and A. Saadat, "Fault isolation of reaction wheels onboard three-axis controlled in-orbit satellite using ensemble machine learning," *Aerospace Systems*, pp. 1–8, 2020.

[19] J. Jin, S. Ko, and C.-K. Ryoo, "Fault tolerant control for satellites with four reaction wheels," *Control Engineering Practice*, vol. 16, no. 10, pp. 1250–1258, 2008.

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

[21] B. M. de Silva, J. Callaham, J. Jonker, N. Goebel, J. Klemisch, D. McDonald, N. Hicks, J. N. Kutz, S. L. Brunton, and A. Y. Aravkin, "Physics-informed machine learning for sensor fault detection with flight test data," *arXiv preprint arXiv:2006.13380*, 2020.

[22] G. H. Janse van Vuuren, "The design and simulation analysis of an attitude determination and control system for a small earth observation satellite," Ph.D. dissertation, Stellenbosch: Stellenbosch University, 2015.

[23] J. H. Wessels, "Infrared horizon sensor for cubesat implementation," Ph.D. dissertation, Stellenbosch: Stellenbosch University, 2018.

[24] *Sun sensor*, Sputnix, May 2020, v13. [Online]. Available: https://sputnix.ru/en/equipment/cubesat-devices/sun-sensor-flight-proof-1

[25] *AAC SpaceQuest MAG-3*, AAC Clyde Space, May 2022, three Axis Satellite Magnetometer. [Online]. Available: https://www.aac-clyde.space/what-we-do/space-products-components/adcs/mag-3

[26] *Microsatellite Reaction Wheels (RW3-0.06)*, Sinclair Interplanetary, 2014. [Online]. Available: https://satsearch.co/products/sinclair-interplanetary-rw3-0-060

[27] *Medium CubeTorquer*, CubeSpace, 2014. [Online]. Available: https://www.cubespace.co.za/products/gen-1/actuators/cubetorquer/

[28] J. Auret, "Design of an aerodynamic attitude control system for a cubesat," Ph.D. dissertation, Stellenbosch: Stellenbosch University, 2012.

[29] W. H. Jordaan, "Spinning solar sail: the deployment and control of a spinning solar sail satellite," Ph.D. dissertation, Stellenbosch: Stellenbosch University, 2016.

[30] C. Jéger, "Determination and compensation of magnetic dipole moment inapplication for a scientific nanosatellite mission," 2017.

[31] X. Chen, W. Steyn, and Y. Hashida, "Ground-target tracking control of Earth-pointing satellites," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, ser. AIAA Guidance, Navigation, and Control Conference and Exhibit. American Institute of Aeronautics and Astronautics.

[32] D. Vallado, P. Crawford, R. Hujsak, and T. Kelso, "Revisiting spacetrack report# 3," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2006, p. 6753.

[33] B. Rhodes. Python package index - sgp4. [Online]. Available: https://pypi.org/project/sgp4/

[34] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[35] K. Großekatthöfer and Z. Yoon, "Introduction into quaternions for spacecraft attitude representation," *TU Berlin*, vol. 16, 2012.

[36] J. R. Wertz, *Spacecraft attitude determination and control*. Springer Science & Business Media, 2012, vol. 73.

[37] D. A. Vallado, *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001, vol. 12.

[38] F. E. Ross, "The sun's mean longitude," *The Astronomical Journal*, vol. 29, pp. 152–156, 1916.

[39] W. H. Campbell, *Introduction to geomagnetic fields*. Cambridge University Press, 2003.

[40] P. Alken, E. Thébault, C. D. Beggan, H. Amit, J. Aubert, J. Baerenzung, T. Bondar, W. Brown, S. Califf, A. Chambodut *et al.*, "International geomagnetic reference field: the thirteenth generation," *Earth, Planets and Space*, vol. 73, no. 1, pp. 1–25, 2021.

[41] D. Winch, D. Ivers, J. Turner, and R. Stening, "Geomagnetism and schmidt quasi-normalization," *Geophysical Journal International*, vol. 160, no. 2, pp. 487–504, 2005.

[42] H. E. Koskinen and E. K. Kilpua, "Radiation belts and their environment," in *Physics of Earth's Radiation Belts*. Springer, 2022, pp. 1–25.

[43] K. Pullen and A. Dhand, "Mechanical and electrical flywheel hybrid technology to store energy in vehicles," in *Alternative fuels and advanced vehicle technologies for improved environmental performance*. Elsevier, 2014, pp. 476–504.

[44] W. H. Steyn and M.-A. Kearney, "An attitude control system for za-aerosat subject to significant aerodynamic disturbances," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 7929–7934, 2014.

[45] W. H. Steyn and V. Lappas, "Cubesat solar sail 3-axis stabilization using panel translation and magnetic torquing," *Aerospace Science and Technology*, vol. 15, no. 6, pp. 476–485, 2011.

[46] B. Bialke, "High fidelity mathematical modeling of reaction wheel performance," in *21st Annual American Astronautical Society Guidance and Control Conference*, 1998.

[47] W. Steyn, "Ground-target tracking control of Earth-pointing satellites," in *AODCS EKF Estimation*, ser. AIAA Guidance, Navigation, and Control Conference and Exhibit. American Institute of Aeronautics and Astronautics.

[48] B. Wie, H. Weiss, and A. Arapostathis, "Quarternion feedback regulator for spacecraft eigenaxis rotations," *Journal of Guidance, Control, and Dynamics*, vol. 12, no. 3, pp. 375–380, 1989.

[49] W. H. Steyn, "An attitude control system for sumbandilasat an earth observation satellite," in *ESA 4S Symposium*, 2008, pp. 1–12.

[50] X. Chen, W. Steyn, and Y. Hashida, "Ground-target tracking control of earth-pointing satellites," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000, p. 4547.

[51] H. M. Van Rensburg, "An infrared earth horizon sensor for a leo satellite," Ph.D. dissertation, Stellenbosch: University of Stellenbosch, 2008.

[52] S. Rückerl, D. Meßmann, N. Appel, J. Kiesbye, F. Schummer, M. Fähling, L. Krempel, T. Kale, A. Lill, G. Reina *et al.*, "First flight results of the move-ii cubesat," 2019.

[53] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[55] M. Reif, M. Goldstein, A. Stahl, and T. M. Breuel, "Anomaly detection by combining decision trees and parametric densities," pp. 1–4, 2008.

[56] T. Shi and S. Horvath, "Unsupervised learning with random forest predictors," *Journal of Computational and Graphical Statistics*, vol. 15, no. 1, pp. 118–138, 2006.

[57] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved random forest for classification," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4012–4024, 2018.

[58] R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," pp. 1–6, 2017.

[59] A. Priyam, G. Abhijeeta, A. Rathee, and S. Srivastava, "Comparative analysis of decision tree classification algorithms," *International Journal of current engineering and technology*, vol. 3, no. 2, pp. 334–337, 2013.

[60] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[61] A. Colagrossi and M. Lavagna, "Fault tolerant attitude and orbit determination system for small satellite platforms," *Aerospace*, vol. 9, no. 2, p. 46, 2022.

[62] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[63] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, no. 1-2, pp. 169–186, 2003.

System Perturbation Matrix

The derivation of the continuous system perturbation matrix, $\mathbf{F}_t$, is repeated from G.Janse van Vuuren [22], with slight adaptions in notation. This derivation is required for the execution of the full state EKF, as described in Section 4.3.2.

The continuous system perturbation matrix $\mathbf{F}_t$ can be constructed by determining its individual components, thus

$$\mathbf{F}_t = \left[\begin{array}{cc} \frac{\partial \dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}}{\partial \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}} & \frac{\partial \dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}}{\partial \mathbf{q}} \\ \frac{\partial \dot{\mathbf{q}}}{\partial \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}} & \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} \end{array} \right]_{\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}=\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}, \mathbf{q}=\hat{\mathbf{q}}} \tag{1}$$

Note that the subscript 't' indicating the time domain has been dropped from Equation 4.31 to simplify the derivation. The non-linear function $\boldsymbol{f}(\mathbf{x})$ can be separated into two parts: a non-linear function describing $\dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}$ and a non-linear function describing $\dot{\mathbf{q}}$. The continuous non-linear system equation with regards to $\dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}$ is the Euler dynamic equation, or

$$\dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}} = \mathbf{J}^{-1}\left(\boldsymbol{\tau}_c + \boldsymbol{\tau}_d - \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}} \times \left(\mathbf{J}\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}} + \mathbf{h}_w\right)\right) \tag{2}$$

The individual components of Equation 2 can also be expressed as

$$\begin{aligned} \dot{\omega}_{\bar{\mathbf{x}}_{\mathcal{I}}} &= \frac{1}{I_{xx}}\left(N_{c\bar{\mathbf{x}}_{\mathcal{B}}} + N_{d\bar{\mathbf{x}}_{\mathcal{B}}} - \omega_{\bar{\mathbf{y}}_{\mathcal{I}}}\left(I_{zz}\omega_{\bar{\mathbf{z}}_{\mathcal{I}}} + h_z\right) + \omega_{\bar{\mathbf{z}}_{\mathcal{I}}}\left(I_{yy}\omega_{\bar{\mathbf{y}}_{\mathcal{I}}} + h_y\right)\right) \\ \dot{\omega}_{\bar{\mathbf{y}}_{\mathcal{I}}} &= \frac{1}{I_{yy}}\left(N_{c\bar{\mathbf{y}}_{\mathcal{B}}} + N_{d\bar{\mathbf{y}}_{\mathcal{B}}} - \omega_{\bar{\mathbf{z}}_{\mathcal{I}}}\left(I_{xx}\omega_{\bar{\mathbf{x}}_{\mathcal{I}}} + h_x\right) + \omega_{\bar{\mathbf{x}}_{\mathcal{I}}}\left(I_{zz}\omega_{\bar{\mathbf{z}}_{\mathcal{I}}} + h_z\right)\right) \\ \dot{\omega}_{\bar{\mathbf{z}}_{\mathcal{I}}} &= \frac{1}{I_{zz}}\left(N_{c\bar{\mathbf{z}}_{\mathcal{B}}} + N_{d\bar{\mathbf{z}}_{\mathcal{B}}} - \omega_{\bar{\mathbf{x}}_{\mathcal{I}}}\left(I_{yy}\omega_{\bar{\mathbf{y}}_{\mathcal{I}}} + h_y\right) + \omega_{\bar{\mathbf{y}}_{\mathcal{I}}}\left(I_{xx}\omega_{\bar{\mathbf{x}}_{\mathcal{I}}} + h_x\right)\right). \end{aligned} \tag{3}$$

Using Equation 3, $\frac{\partial \dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}}{\partial \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}}$ can be determined by taking each individual partial derivative, which delivers

$$\frac{\partial \dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}}{\partial \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}} = \left[\begin{array}{ccc} 0 & \frac{\omega_{\bar{\mathbf{z}}_{\mathcal{I}}}(I_{yy}-I_{zz})-h_z}{I_{xx}} & \frac{\omega_{\bar{\mathbf{y}}_{\mathcal{I}}}(I_{yy}-I_{zz})+h_y}{I_{xx}} \\ \frac{\omega_{\bar{\mathbf{z}}_{\mathcal{I}}}(I_{zz}-I_{xx})+h_z}{I_{yy}} & 0 & \frac{\omega_{\bar{\mathbf{x}}_{\mathcal{I}}}(I_{zz}-I_{xx})-h_x}{I_{yy}} \\ \frac{\omega_{\bar{\mathbf{y}}_{\mathcal{I}}}(I_{xx}-I_{yy})-h_y}{I_{zz}} & \frac{\omega_{\bar{\mathbf{x}}_{\mathcal{I}}}(I_{xx}-I_{yy})+h_x}{I_{zz}} & 0 \end{array} \right]. \tag{4}$$

$\frac{\partial \dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}}{\partial \mathbf{q}}$ is however much more difficult to determine. The first step is to determine which components of Equation 2 are dependent on the attitude quaternion of the satellite. The control torque $\boldsymbol{\tau}_c$ is the sum of the torques generated by the ADCS actuators, which means that $\boldsymbol{\tau}_c$ is independent of $\mathbf{q}$. $\boldsymbol{\tau}_{\text{gyro}}$ is calculated using only the moment of inertia

matrix, the angular rates and the stored angular momentum, which means that $\boldsymbol{\tau}_{\text{gyro}}$ is also independent of $\mathbf{q}$. Although there are many sources of disturbance torques, $\boldsymbol{\tau}_d$ at a LEO orbit is simplified to contain only two major components, namely gravity gradient torque ($\boldsymbol{\tau}_{gg}$) and aerodynamic torque ($\boldsymbol{\tau}_{\text{aero}}$). Even though both these components are dependent on the attitude of the satellite, only $\boldsymbol{\tau}_{gg}$ can be calculated accurately, thus

$$\boldsymbol{\tau}_d \approx \boldsymbol{\tau}_{gg} \tag{5}$$

$\boldsymbol{\tau}_d$ can thus easily be expressed in terms of quaternions using a simplification of Equation 4.21 as

$$\begin{aligned}
\mathrm{N}_{d\bar{\mathbf{x}}_{\mathcal{B}}} &\approx k_{g\bar{\mathbf{x}}_{\mathcal{B}}} \left(2\left[q_2q_3 + q_1q_4\right]\right)\left(-q_1^2 - q_2^2 + q_3^2 + q_4^2\right) \\
\mathrm{N}_{d\bar{\mathbf{y}}_{\mathcal{B}}} &\approx k_{g\bar{\mathbf{y}}_{\mathcal{B}}} \left(2\left[q_1q_3 - q_2q_4\right]\right)\left(-q_1^2 - q_2^2 + q_3^2 + q_4^2\right) \\
\mathrm{N}_{d\bar{\mathbf{z}}_{\mathcal{B}}} &\approx k_{g\bar{\mathbf{z}}_{\mathcal{B}}} \left(2\left[q_1q_3 - q_2q_4\right]\right)\left(2\left[q_2q_3 + q_1q_4\right]\right)
\end{aligned} \tag{6}$$

$\frac{\partial \dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}}{\partial \mathbf{q}}$ can now be calculated as

$$\frac{\partial \dot{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}}}{\partial \mathbf{q}} = \mathbf{J}^{-1}\left[\frac{\partial \boldsymbol{\tau}_d}{\partial \mathbf{q}}\right] = \mathbf{K}\begin{bmatrix} \mathbf{d}_1 & \mathbf{d}_2 & \mathbf{d}_3 & \mathbf{d}_4 \end{bmatrix} \tag{7}$$

where

$$\mathbf{K} = \begin{bmatrix} 2k_{g\bar{\mathbf{x}}_{\mathcal{B}}} & 0 & 0 \\ 0 & 2k_{g\bar{\mathbf{y}}_{\mathcal{B}}} & 0 \\ 0 & 0 & 2k_{g\bar{\mathbf{z}}_{\mathcal{B}}} \end{bmatrix} \tag{8}$$

and

$$\begin{aligned}
\mathbf{d}_1 &= \begin{bmatrix} \frac{-q_1A_{23}+q_4A_{33}}{I_{xx}} \\ \frac{-q_1A_{13}+q_3A_{33}}{I_{yy}} \\ \frac{q_3A_{23}+q_4A_{13}}{I_{zz}} \end{bmatrix} \quad \mathbf{d}_2 = \begin{bmatrix} \frac{-q_2A_{23}+q_3A_{33}}{I_{xx}} \\ \frac{-q_2A_{13}-q_4A_{33}}{I_{yy}} \\ \frac{-q_4A_{23}+q_3A_{13}}{I_{zz}} \end{bmatrix} \\
\mathbf{d}_3 &= \begin{bmatrix} \frac{q_3A_{23}+q_2A_{33}}{I_{xx}} \\ \frac{q_3A_{13}+q_1A_{33}}{I_{yy}} \\ \frac{q_1A_{23}+q_2A_{13}}{I_{zz}} \end{bmatrix} \quad \mathbf{d}_4 = \begin{bmatrix} \frac{q_4A_{23}+q_1A_{33}}{I_{xx}} \\ \frac{q_4A_{13}-q_2A_{33}}{I_{yy}} \\ \frac{-q_2A_{23}+q_1A_{13}}{I_{zz}} \end{bmatrix}.
\end{aligned} \tag{9}$$

$\frac{\partial \dot{\mathbf{q}}}{\partial \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}}$ and $\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}}$ can be determined by partially deriving the time derivative of $\mathbf{q}$, which is

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}})\mathbf{q}, \tag{10}$$

where

$$\boldsymbol{\Omega}(\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}}) = \begin{bmatrix} 0 & \omega_{zo} & -\omega_{yo} & \omega_{xo} \\ -\omega_{zo} & 0 & \omega_{xo} & \omega_{yo} \\ \omega_{yo} & -\omega_{xo} & 0 & \omega_{zo} \\ -\omega_{xo} & -\omega_{yo} & -\omega_{zo} & 0 \end{bmatrix} \tag{11}$$

The relationship between $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}$ and $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}}$ is given by

$$\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}} = \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}} - \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}} \begin{bmatrix} 0 \\ -\omega_o \\ 0 \end{bmatrix} = \begin{bmatrix} \omega_{\bar{\mathbf{x}}_{\mathcal{I}}} + \omega_o A_{12} \\ \omega_{\bar{\mathbf{y}}_{\mathcal{I}}} + \omega_o A_{22} \\ \omega_{\bar{\mathbf{z}}_{\mathcal{I}}} + \omega_o A_{32} \end{bmatrix} \tag{12}$$

which means that $\frac{\partial \dot{\mathbf{q}}}{\partial \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}}$ can be determined as

$$\frac{\partial \dot{\mathbf{q}}}{\partial \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}} = \begin{bmatrix} \frac{\partial \dot{q}_1}{\partial \omega_{\bar{\mathbf{x}}_{\mathcal{I}}}} & \frac{\partial \dot{q}_1}{\partial \omega_{\bar{\mathbf{y}}_{\mathcal{I}}}} & \frac{\partial \dot{q}_1}{\partial \omega_{\bar{\mathbf{z}}_{\mathcal{I}}}} \\ \frac{\partial \dot{q}_2}{\partial \omega_{\bar{\mathbf{x}}_{\mathcal{I}}}} & \frac{\partial \dot{q}_2}{\partial \omega_{\bar{\mathbf{y}}_{\mathcal{I}}}} & \frac{\partial \dot{q}_2}{\partial \omega_{\bar{\mathbf{z}}_{\mathcal{I}}}} \\ \frac{\partial \dot{q}_3}{\partial \omega_{\bar{\mathbf{x}}_{\mathcal{I}}}} & \frac{\partial \dot{q}_3}{\partial \omega_{\bar{\mathbf{y}}_{\mathcal{I}}}} & \frac{\partial \dot{q}_3}{\partial \omega_{\bar{\mathbf{z}}_{\mathcal{I}}}} \\ \frac{\partial \dot{q}_4}{\partial \omega_{\bar{\mathbf{x}}_{\mathcal{I}}}} & \frac{\partial \dot{q}_4}{\partial \omega_{\bar{\mathbf{y}}_{\mathcal{I}}}} & \frac{\partial \dot{q}_4}{\partial \omega_{\bar{\mathbf{z}}_{\mathcal{I}}}} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \hat{q}_4 & -\hat{q}_3 & \hat{q}_2 \\ \hat{q}_3 & \hat{q}_4 & -\hat{q}_1 \\ -\hat{q}_2 & \hat{q}_1 & \hat{q}_4 \\ -\hat{q}_1 & -\hat{q}_2 & -\hat{q}_3 \end{bmatrix} \tag{13}$$

$\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}}$ can be determined by substituting Equation 12 and Equation 11 into Equation 10, which delivers

$$\begin{aligned} \dot{q}_1 &= \frac{1}{2}\left(q_2\left(\omega_{\bar{\mathbf{z}}_{\mathcal{I}}} - \omega_o A_{32}\right) - q_3\left(\omega_{\bar{\mathbf{y}}_{\mathcal{I}}} - \omega_o A_{22}\right) + q_4\left(\omega_{\bar{\mathbf{x}}_{\mathcal{I}}} - \omega_o A_{12}\right)\right) \\ \dot{q}_2 &= \frac{1}{2}\left(-q_1\left(\omega_{\bar{\mathbf{z}}_{\mathcal{I}}} - \omega_o A_{32}\right) + q_3\left(\omega_{\bar{\mathbf{x}}_{\mathcal{I}}} - \omega_o A_{12}\right) + q_4\left(\omega_{\bar{\mathbf{y}}_{\mathcal{I}}} - \omega_o A_{22}\right)\right) \\ \dot{q}_3 &= \frac{1}{2}\left(q_1\left(\omega_{\bar{\mathbf{y}}_{\mathcal{I}}} - \omega_o A_{22}\right) - q_2\left(\omega_{\bar{\mathbf{x}}_{\mathcal{I}}} - \omega_o A_{12}\right) + q_4\left(\omega_{\bar{\mathbf{z}}_{\mathcal{I}}} - \omega_o A_{32}\right)\right) \\ \dot{q}_4 &= \frac{1}{2}\left(-q_1\left(\omega_{\bar{\mathbf{x}}_{\mathcal{I}}} - \omega_o A_{12}\right) - q_2\left(\omega_{\bar{\mathbf{y}}_{\mathcal{I}}} - \omega_o A_{22}\right) - q_3\left(\omega_{\bar{\mathbf{z}}_{\mathcal{I}}} - \omega_o A_{32}\right)\right) \end{aligned} \tag{14}$$

By partially deriving Equations 14 and performing some mathematical manipulation, $\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}}$ can be calculated as

$$\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} = \frac{1}{2}\left[\boldsymbol{\Omega}\left(\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{O}}\right)\right] + \omega_o \begin{bmatrix} \hat{q}_1\hat{q}_3 & \hat{q}_1\hat{q}_4 & 1-\hat{q}_1^2 & -\hat{q}_1\hat{q}_2 \\ \hat{q}_2\hat{q}_3 & \hat{q}_2\hat{q}_4 & -\hat{q}_1\hat{q}_2 & 1-\hat{q}_2^2 \\ -\left(1-\hat{q}_3^2\right) & \hat{q}_3\hat{q}_4 & -\hat{q}_1\hat{q}_3 & -\hat{q}_2\hat{q}_3 \\ \hat{q}_3\hat{q}_4 & -\left(1-\hat{q}_4^2\right) & -\hat{q}_1\hat{q}_4 & -\hat{q}_2\hat{q}_4 \end{bmatrix} \tag{15}$$

Measurement Perturbation Jacobian Matrix

The derivation of the measurement perturbation Jacobian matrix, $\mathbf{H}_k$, is repeated from G.Janse van Vuuren [22], with slight adaptions in notation. This derivation is required for the execution of the full state EKF, as described in Section 4.3.2.

The discrete measurement perturbation matrix $\mathbf{H}_k$ can be determined by partially deriving the non-linear function $\boldsymbol{h}(\mathbf{x}_k)$, which is as

$$\boldsymbol{h}(\mathbf{x}_k) = \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}} \mathbf{v}_{\mathcal{O}_k}. \tag{16}$$

Since $\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}$ is constructed from $\mathbf{q}$ only, Equation 16 suggests that $\boldsymbol{h}(\mathbf{x}_k)$ is independent of $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}$, thus

$$\mathbf{H}_k = \begin{bmatrix} 0 & 0 & 0 & \frac{\partial h_1}{\partial q_1} & \frac{\partial h_1}{\partial q_2} & \frac{\partial h_1}{\partial q_3} & \frac{\partial h_1}{\partial q_4} \\ 0 & 0 & 0 & \frac{\partial h_2}{\partial q_1} & \frac{\partial h_2}{\partial q_2} & \frac{\partial h_2}{\partial q_3} & \frac{\partial h_2}{\partial q_4} \\ 0 & 0 & 0 & \frac{\partial h_3}{\partial q_1} & \frac{\partial h_3}{\partial q_2} & \frac{\partial h_3}{\partial q_3} & \frac{\partial h_3}{\partial q_4} \end{bmatrix}_{\mathbf{q}=\hat{\mathbf{q}}} \tag{17}$$

$\mathbf{H}_k$ can thus be calculated as

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{0}_{[3\times 3]} & \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_3 & \mathbf{H}_4 \end{bmatrix}, \tag{18}$$

where

$$\begin{aligned} \mathbf{H}_1 &= 2 \begin{bmatrix} q_1 & q_2 & q_3 \\ q_2 & -q_1 & q_4 \\ q_3 & -q_4 & -q_1 \end{bmatrix} \mathbf{v}_{\mathcal{O}_k}, \\ \mathbf{H}_2 &= 2 \begin{bmatrix} -q_2 & q_1 & -q_4 \\ q_1 & q_2 & q_3 \\ q_4 & q_3 & -q_2 \end{bmatrix} \mathbf{v}_{\mathcal{O}_k} \\ \mathbf{H}_3 &= 2 \begin{bmatrix} -q_3 & q_4 & q_1 \\ -q_4 & -q_3 & q_2 \\ q_1 & q_2 & q_3 \end{bmatrix} \mathbf{v}_{\mathcal{O}_k} \end{aligned} \tag{19}$$

$$\text{and} \qquad \mathbf{H}_4 = 2 \begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \end{bmatrix} \mathbf{v}_{\mathcal{O}_k}.$$

System Noise Covariance Matrix

The derivation of the system noise covariance matrix, $\mathbf{Q}_k$, is repeated from G.Janse van Vuuren [22], with slight adaptions in notation. This derivation is required for the execution of the full state EKF, as described in Section 4.3.2.

The system noise covariance matrix $\mathbf{Q}_k$ can easily be determined from the continuous domain system noise covariance matrix $\mathbf{Q}_t$ if the following assumptions are made:

- The angular rate noise (due to unmodelled disturbance torques and modelling errors) is uncorrelated.
- The system noise is small enough to allow the state matrix $\mathbf{\Phi}_k$ to be approximated using only two terms without significant inaccuracies, thus $\mathbf{\Phi}_k \approx \mathbf{I} + [T_s \mathbf{F}_t]_{t=kT_s}$.
- The angular rate noise for each axis is equal, thus $\sigma_{\omega \bar{\mathbf{x}}_{\mathcal{B}}^{\mathcal{O}}} = \sigma_{\omega \bar{\mathbf{y}}_{\mathcal{B}}^{\mathcal{O}}} = \sigma_{\omega \bar{\mathbf{z}}_{\mathcal{B}}^{\mathcal{O}}} = \sigma_{\omega_{\mathcal{B}}^{\mathcal{O}}}$.

Given the above-mentioned assumptions, the angular rate noise covariance matrix $\mathbf{Q}_{\boldsymbol{\omega},t}$ is given as

$$\mathbf{Q}_{\boldsymbol{\omega},t} = \begin{bmatrix} \sigma_\omega^2 & 0 & 0 \\ 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & \sigma_\omega^2 \end{bmatrix} \tag{20}$$

$\mathbf{q}$ is completely described by the equations in Section 5.4, which means that the noise covariance of the last four states of the system ($\mathbf{Q}_{\mathbf{q},t}$) is simply a zero matrix, or

$$\mathbf{Q}_{\mathbf{q},t} = \mathbf{0}_{[4\times4]} \tag{21}$$

$\mathbf{Q}_t$ can be formed from Equations B.23 and B.24 as

$$\begin{aligned} \mathbf{Q}_t &= \begin{bmatrix} \mathbf{Q}_{\boldsymbol{\omega},t} & \mathbf{0}_{[3\times4]} \\ \mathbf{0}_{[4\times3]} & \mathbf{Q}_{\mathbf{q},t} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{Q}_{\boldsymbol{\omega},t} & \mathbf{0}_{[3\times4]} \\ \mathbf{0}_{[4\times3]} & \mathbf{0}_{[4\times4]} \end{bmatrix} \end{aligned} \tag{22}$$

$\mathbf{F}_t$ can also be express in the form of Equation B.26 as

$$\mathbf{F}_t = \begin{bmatrix} \mathbf{F}_{11[3\times3]} & \mathbf{F}_{12[3\times4]} \\ \mathbf{F}_{21[4\times3]} & \mathbf{F}_{22[4\times4]} \end{bmatrix} \tag{23}$$

$\mathbf{Q}_k$ can now be determined by converting $\mathbf{Q}_t$ to the discrete domain. Through a process of integration [21], $\mathbf{Q}_k$ is determined to be

$$\mathbf{Q}_k = T_s\mathbf{S}_1 + \frac{1}{2}T_s^2\mathbf{S}_2 + \frac{1}{3}T_s^3\mathbf{S}_3 \tag{24}$$

where $\mathbf{S}_1 = \mathbf{Q}_t$

$$\begin{aligned}\mathbf{S}_2 &= \begin{bmatrix} \mathbf{Q}_{\boldsymbol{\omega},t}\mathbf{F}_{11}^{\mathrm{T}} + \mathbf{F}_{11}\mathbf{Q}_{\boldsymbol{\omega},t} & \mathbf{Q}_{\boldsymbol{\omega},t}\mathbf{F}_{21}^{\mathrm{T}} \\ \mathbf{F}_{21}\mathbf{Q}_{\boldsymbol{\omega},t} & \mathbf{0}_{[4\times4]} \end{bmatrix}_{t=kT_s} \\ \mathbf{S}_3 &= \begin{bmatrix} \mathbf{F}_{11}\mathbf{Q}_{\boldsymbol{\omega},t}\mathbf{F}_{11}^{\mathrm{T}} & \mathbf{F}_{11}\mathbf{Q}_{\boldsymbol{\omega},t}\mathbf{F}_{21}^{\mathrm{T}} \\ \mathbf{F}_{21}\mathbf{Q}_{\boldsymbol{\omega},t}\mathbf{F}_{11}^{\mathrm{T}} & \mathbf{F}_{21}\mathbf{Q}_{\boldsymbol{\omega},t}\mathbf{F}_{21}^{\mathrm{T}} \end{bmatrix}_{t=kT_s}\end{aligned} \tag{25}$$

The computational load of calculating $\mathbf{Q}_k$ can be reduced if the assumption is made that $\mathbf{F}_{11} << \mathbf{F}_{21}$ [21]. $\mathbf{Q}_k$ can then be simplified to

$$\mathbf{Q}_k = \begin{bmatrix} T_s\mathbf{Q}_{\boldsymbol{\omega},t} & \frac{1}{2}{T_s}^2\mathbf{Q}_{\boldsymbol{\omega},t}\mathbf{F}_{21}^{\mathrm{T}} \\ \frac{1}{2}{T_s}^2\mathbf{F}_{21}\mathbf{Q}_{\boldsymbol{\omega},t} & \frac{1}{3}{T_s}^3\mathbf{F}_{21}\mathbf{Q}_{\boldsymbol{\omega},t}\mathbf{F}_{21}^{\mathrm{T}} \end{bmatrix}_{t=kT_s} \tag{26}$$

Measurement Noise Covariance Matrix

The derivation of the measurement noise covariance matrix, $\mathbf{R}_k$, is repeated from G.Janse van Vuuren [22], with slight adaptions in notation. This derivation is required for the execution of the full state EKF, as described in Section 4.3.2.

The relationship between the true measured vector $\overline{\mathbf{v}}_{\mathcal{B}_k}$ and the true modelled vector $\overline{\mathbf{v}}_{\mathcal{O}_k}$ is given as

$$\overline{\mathbf{v}}_{\mathcal{B}_k} = \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\mathbf{q}_k\right)\overline{\mathbf{v}}_{\mathcal{O}_k}. \tag{27}$$

It should be noted that $\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B},k} = \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\mathbf{q}_k\right)$. The added notation, which merely implies that $\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}$ is a function of $\mathbf{q}_k$, will prove to be useful in the remainder of this section.

The measured and modelled vectors are furthermore related to their respective true vectors through

$$\begin{aligned} \mathbf{v}_{\mathcal{B}_k} &= \overline{\mathbf{v}}_{\mathcal{B}_k} + \mathbf{m}_{\mathcal{B}_k} \\ \text{and} \quad \mathbf{v}_{\mathcal{O}_k} &= \overline{\mathbf{v}}_{\mathcal{O}_k} + \mathbf{m}_{\mathcal{O}_k} \end{aligned} \tag{28}$$

If $\Delta\mathbf{q}_k$ is defined as the difference between the true quaternion $\mathbf{q}_k$ and the estimated quaternion $\hat{\mathbf{q}}_k$,

$$\Delta\mathbf{q}_k = \mathbf{q}_k - \hat{\mathbf{q}}_k \tag{29}$$

then Equation 27 can also be expressed as

$$\mathbf{v}_{\mathcal{B}_k} - \mathbf{m}_{\mathcal{B}_k} = \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k + \Delta\mathbf{q}_k\right)\left(\mathbf{v}_{\mathcal{O}_k} - \mathbf{m}_{\mathcal{O}_k}\right) \tag{30}$$

The Taylor series expansion from Equation 4.31 can also be used to approximate $\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k + \Delta\mathbf{q}_k\right)$ as

$$\begin{aligned} \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k + \Delta\mathbf{q}_k\right) &\approx \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) + \mathbf{C}_k\Delta\mathbf{q}_k \\ \text{where} \quad \mathbf{C}_k &= \left[\frac{\partial \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right)}{\partial\hat{\mathbf{q}}_k}\right] \end{aligned} \tag{31}$$

Substituting Equation 31 into Equation 30 delivers

$$\mathbf{v}_{\mathcal{B}_k} - \mathbf{m}_{\mathcal{B}_k} = \left(\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) + \mathbf{C}_k\Delta\mathbf{q}_k\right)\left(\mathbf{v}_{\mathcal{O}_k} - \mathbf{m}_{\mathcal{O}_k}\right) \tag{32}$$

Given that the innovation $\mathbf{e}_k$ is defined by Equation 4.41 as

$$\mathbf{e}_k = \mathbf{v}_{\mathcal{B}_k} - \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right)\mathbf{v}_{\mathcal{O}_k}, \tag{33}$$

substitution can be used to manipulate Equation 32 into

$$\mathbf{e}_k = \left(\mathbf{C}_k \mathbf{v}_{\mathcal{O}_k} - \mathbf{C}_k \mathbf{m}_{\mathcal{O}_k}\right) \Delta\mathbf{q}_k + \mathbf{m}_{\mathcal{B}_k} - \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) \mathbf{m}_{\mathcal{O}_k} \tag{34}$$

If it is assumed that $\mathbf{m}_{\mathcal{O}_k}$ and $\Delta\mathbf{q}_k$ are extremely small compared respectively to $\mathbf{v}_{\mathcal{O},k}$ and $\mathbf{q}_k$, then

$$\mathbf{m}_{\mathcal{O}_k} \Delta\mathbf{q}_k \approx 0 \tag{35}$$

Equation 34 can thus be simplified to

$$\begin{aligned} \mathbf{e}_k &= \left[\mathbf{0}_{[3\times3]} \mathbf{C}_k \mathbf{v}_{\mathcal{O}_k}\right] \Delta\mathbf{x}_k + \mathbf{m}_k, \\ \text{where} \quad \mathbf{m}_k &= \mathbf{m}_{\mathcal{B}_k} - \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) \mathbf{m}_{\mathcal{O}_k} \end{aligned} \tag{36}$$

The covariance matrix $\mathbf{R}_k$ of $\mathbf{m}_k$ is defined as

$$\begin{aligned} \mathbf{R}_k =& \mathbb{E}\left\{\left(\mathbf{m}_k\right)\left(\mathbf{m}_k\right)^{\mathrm{T}}\right\} \\ =& \mathbb{E}\left\{\left(\mathbf{m}_{\mathcal{B}_k} - \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) \mathbf{m}_{\mathcal{O}_k}\right)\left(\mathbf{m}_{\mathcal{B}_k} - \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) \mathbf{m}_{\mathcal{O}_k}\right)^{\mathrm{T}}\right\} \\ =& \mathbb{E}\left\{\mathbf{m}_{\mathcal{B}_k}\mathbf{m}_{\mathcal{B}_k}^{\mathrm{T}} - \right. \\ & \quad \mathbf{m}_{\mathcal{B}_k}\mathbf{m}_{\mathcal{O}_k}^{\mathrm{T}} \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right)^{\mathrm{T}} - \\ & \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) \mathbf{m}_{\mathcal{O}_k}\mathbf{m}_{\mathcal{B}_k}^{\mathrm{T}} + \\ & \left. \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) \mathbf{m}_{\mathcal{O}_k}\mathbf{m}_{\mathcal{O}_k}^{\mathrm{T}} \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right)^{\mathrm{T}}\right\} \\ =& \mathbb{E}\left\{\mathbf{m}_{\mathcal{B}_k}\mathbf{m}_{\mathcal{B}_k}^{\mathrm{T}}\right\} - \\ & \mathbb{E}\left\{\mathbf{m}_{\mathcal{B}_k}\mathbf{m}_{\mathcal{O}_k}^{\mathrm{T}} \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right)^{\mathrm{T}}\right\} - \\ & \mathbb{E}\left\{\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) \mathbf{m}_{\mathcal{O}_k}\mathbf{m}_{\mathcal{B}_k}^{\mathrm{T}}\right\} + \\ & \mathbb{E}\left\{\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) \mathbf{m}_{\mathcal{O}_k}\mathbf{m}_{\mathcal{O}_k}^{\mathrm{T}} \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right)^{\mathrm{T}}\right\} \end{aligned} \tag{37}$$

where $\mathbb{E}$ indicates the expected value operator. The last term of Equation 37 can be simplified to

$$\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) \mathbf{m}_{\mathcal{O}_k}\mathbf{m}_{\mathcal{O}_k}^{\mathrm{T}} \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right)^{\mathrm{T}} = \mathbf{m}_{\mathcal{O}_k}\mathbf{m}_{\mathcal{O}_k}^{\mathrm{T}} \tag{38}$$

since $\mathbf{m}_{\mathcal{O}_k}\mathbf{m}_{\mathcal{O}_k}^{\mathrm{T}}$ is a scalar value and $\boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right) \boldsymbol{A}_{\mathcal{O}}^{\mathcal{B}}\left(\hat{\mathbf{q}}_k\right)^{\mathrm{T}} = 1$. If it is furthermore assumed that the measurement noise and the model noise are uncorrelated and that each noise vector has equal variance in its 3 axes, then Equation 37 becomes

$$\begin{aligned} \mathbf{R}_k =& \mathbb{E}\left\{\mathbf{m}_{\mathcal{B}_k}\mathbf{m}_{\mathcal{B}_k}^{\mathrm{T}}\right\} + \\ & \mathbb{E}\left\{\mathbf{m}_{\mathcal{O}_k}\mathbf{m}_{\mathcal{O}_k}^{\mathrm{T}}\right\} \\ =& \left(\sigma_{\mathcal{B}}^2 + \sigma_{\mathcal{O}}^2\right) \mathbf{I}_{3\times3} \end{aligned} \tag{39}$$

where $\sigma_{\mathcal{B}}$ and $\sigma_{\mathcal{O}}$ are the respective standard deviations of $\mathbf{m}_{\mathcal{B}_k}$ and $\mathbf{m}_{\mathcal{O}_k}$. It is also

assumed that $\sigma_{\mathcal{B}}$ and $\sigma_{\mathcal{O}}$ are constant.