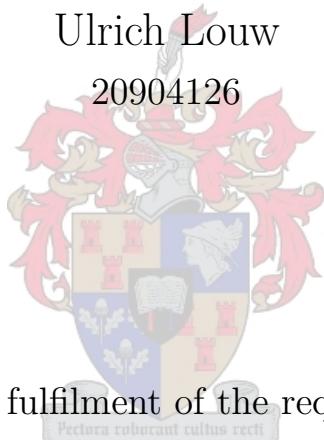


Autonomous Diagnosis of Satellite Sensor Anomalies to Ensure Fault-Tolerant Control

Ulrich Louw

20904126



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

September 2022

Acknowledgements

I would like to thank my dog, Muffin. I also would like to thank the inventor of the incubator; without him/her, I would not be here. Finally, I would like to thank Dr Herman Kamper for this amazing report template.



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 it 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. Dienooreenkomsdig 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
Voorletters en van / Initials and surname	Datum / Date

Abstract

English

The English abstract.

Afrikaans

Die Afrikaanse uittreksel.

Contents

Declaration	ii
Abstract	iii
List of Figures	vii
List of Tables	ix
Nomenclature	x
1. Introduction	1
1.1. Background	1
1.2. Problem Description	2
1.3. Project definition	3
1.4. Thesis outline	4
2. Literature Study	5
2.1. Fault Detection Isolation and Recovery in General	5
2.2. Fault Tolerant Control of Satellites	6
2.3. Fault Tolerant Control with Focus on Sensor Anomalies	7
2.3.1. Innovation in Fault Detection	9
2.4. Evaluation	10
2.5. Summary	10
3. Satellite Design	11
3.1. CubeSat	13
3.2. Sensors	14
3.3. Actuators	14
4. Simulation Environment	15
4.1. Satellite Orbit Fundamentals	15
4.1.1. Coordinate Frames	15
4.1.2. Attitude	17
4.1.3. Satellite Kinematics and Dynamics	19
4.2. Environment	19
4.2.1. Orbit Propagation	20

4.2.2. Sun	21
4.2.3. Geomagnetic field	22
4.3. Disturbance models	23
4.3.1. Gravity Gradient	24
4.3.2. Aerodynamic Disturbance	24
4.3.3. Wheel Imbalance	26
5. Attitude Determination and Control System	29
5.1. Sensor models	29
5.2. Attitude Determination	29
5.2.1. Extended Kalman Filter	30
5.3. Attitude Control	34
5.3.1. Quaternion Feedback Controller	34
5.3.2. Momentum Dumping	35
6. Anomalies	37
6.1. Reflection of Solar Panels on Sun Sensor	37
6.1.1. Influence of Anomaly on Estimation	39
6.2. Moon in Field of View of Nadir Sensor	40
6.2.1. Simulating Nadir Sensor Infra-red Image	40
6.2.2. Calculating the Centre of the Earth	43
6.2.3. Influence of Anomaly on Estimation	45
6.3. Magnetic Moment Disturbance	45
6.3.1. Influence of Anomaly on Estimation	48
6.4. Reaction wheels	49
6.5. Summary	49
7. Feature Extraction	51
7.1. Local Outlier Factor	51
7.2. Linear Regression Model	53
7.3. Summary	54
8. Recovery	55
8.1. Analysis	56
8.1.1. Sun Reflection	56
8.1.2. Magnetic Moment Disturbance	56
8.1.3. Reaction Wheel Failure	56
9. Detection and Isolation	57
9.1. Supervised Learning	57
9.1.1. Decision Trees	57

9.1.2. Random Forests	59
9.1.3. Support Vector Machines	59
9.2. Unsupervised Learning	62
9.2.1. Isolation Forests	63
9.3. Analysis	65
9.4. Summary	65
10. Isolation	66
10.1. Decision Trees	66
10.2. Random Forests	66
10.3. Support Vector Machines	66
10.4. Summary	66
11. Results	67
11.1. Perfect Extended Kalman Filter	67
11.2. Unsupervised Detection and Supervised Isolation	67
11.3. Supervised Detection and Isolation	67
11.4. Summary	67
12. Conclusion	68
12.1. Project/thesis/dissertation summary	68
12.2. Appraisal of project/thesis/dissertation contributions	69
12.3. Suggestions for future work	69
12.4. What the student has learnt during this project	69
Bibliography	70
A. System Perturbation Matrix	75
B. Measurement Perturbation Jacobian Matrix	78
C. System Noise Covariance Matrix	80
D. Measurement Noise Covariance Matrix	82

List of Figures

1.1.	On-orbit failure per subsystem [1]	1
1.2.	System Diagram	3
2.1.	Development of FDIR technique according to expert-system-based approach.	5
2.2.	Development of FDIR technique according to data-driven approach.	6
2.3.	Development of FDIR technique according to data-driven approach.	8
2.4.	Development of FDIR technique according to data-driven approach.	9
3.1.	The percentage occurrences of different ADCS sensors on small satellites . .	12
3.2.	The percentage occurrences of different ADCS actuators on small satellites .	12
3.3.	The modelled satellite, with solar panels.	13
4.1.	Euler angles	17
4.2.	Graphical quaternion representation	18
4.3.	SGP4 orbit propagation	20
4.4.	Geometry for satellite eclipse	22
4.5.	The magnitude of geomagnetic field according to the 13th generation of the IGRF model [2].	23
4.6.	Gravity Gradient Torque in SBC	25
4.7.	Aerodynamic Torques in SBC	26
4.8.	Static Imbalance	27
4.9.	Dynamic Imbalance	27
4.10.	Wheel disturbance torques in SBC	28
5.1.	Sun vector in SBC	30
5.2.	Estimation Metric	33
5.3.	Pointing Metric	35
5.4.	Magnetic Control Torques	36
6.1.	Definition of solar reflection point from $ABCD$ -plane.	37
6.2.	Comparison of measured sun vector with and without reflections (gray areas indicate the eclipse period of the orbit)	39
6.3.	Estimation Metric with reflection on sun sensor	40
6.4.	Earth to Moon and Earth to Satellite Vectors	41
6.5.	Plane perpendicular to \mathbf{R}_{SE} and at center of Earth	42

6.6. Projection of moon and Earth on plane	43
6.7. Circular Fit Algorithm	44
6.8. Comparison of Earth Vector with and without moon on the horizon	45
6.9. Earth Values	46
6.10. Dipole Moment from circular loop in solar panel	46
6.11. Shadow created by CubeSat body on Solar Panels	47
6.12. Solar Panel Disturbance Torques	48
6.13. Comparison of Magnetic field Vector with and without solar Panel magnetic field	48
6.14. Estimation Metric with induced dipole moment	49
6.15. Estimation Metric with failure of Reaction Wheels	50
7.1. Local outlier factor of random data set to demonstrate the outlier score produced by the local outlier factor algorithm.	52
9.1. Decision Tree as binary classification of sun reflection. 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.	58
9.2. Support Vector Machine Example	60
9.3. Support Vector Machine Higher Dimension Example	61
9.4. Support Vector Machine Polynomial Example	62
9.5. Isolation Forest	63
9.6. Isolation Forest	64
9.7. Slicing of Isolation Forest	65

List of Tables

3.1. Physical dimensions of simulated CubeSat	13
3.2. Standard deviation for each sensor	14
4.1. Orbit parameters for simulated CubeSat	21
4.2. Description and definition of Earth orbit parameters	22

Nomenclature

Variables and functions

$p(x)$	Probability density function with respect to variable x .
$P(A)$	Probability of event A occurring.
ε	The Bayes error.
ε_u	The Bhattacharyya bound.
B	The Bhattacharyya distance.
s	An HMM state. A subscript is used to refer to a particular state, e.g. s_i refers to the i^{th} state of an HMM.
\mathbf{S}	A set of HMM states.
\mathbf{F}	A set of frames.
\mathbf{o}_f	Observation (feature) vector associated with frame f .
$\gamma_s(\mathbf{o}_f)$	A posteriori probability of the observation vector \mathbf{o}_f being generated by HMM state s .
μ	Statistical mean vector.
Σ	Statistical covariance matrix.
$L(\mathbf{S})$	Log likelihood of the set of HMM states \mathbf{S} generating the training set observation vectors assigned to the states in that set.
$\mathcal{N}(\mathbf{x} \mu, \Sigma)$	Multivariate Gaussian PDF with mean μ and covariance matrix Σ .
a_{ij}	The probability of a transition from HMM state s_i to state s_j .
N	Total number of frames or number of tokens, depending on the context.
D	Number of deletion errors.
I	Number of insertion errors.
S	Number of substitution errors.

Acronyms and abbreviations

AOCS	Attitude and Orbit Control System
ADCS	Attitude Determination and Control System
EKF	Extended Kalman Filter
FDIR	Fault Detection, Isolation and Recovery
EIC	Earth Inertial Coordinate
EFC	Earth Fixed Coordinate
GHA	Greenwich Hour Angle
ORC	Orbit-referenced Coordinate
SBC	Satellite Body Coordinate
DCM	Direct Cosine Matrix
TLE	Two-line Element
RAAN	Right Ascension of the Ascending Node
AP	Argument of Perigee
SGP	Simplified General Perturbations
IGRF	International Geomagnetic Reference Field
IAGA	International Association of Geomagnetism and Aeronomy
LEO	Low Earth Orbit
CoM	Centre of Mass
CoP	Centre of Pressure
RW	Reaction Wheel
FoV	Field of View
DMD	Dynamic Mode Decomposition
CART	Classification and Regression Trees
BST	Binary Search Tree
LOF	Local Outlier Factor
IRC	Inertial-referenced Coordinate
CART	Classification and Regression Tree
SVM	Support Vector Machines
RF	Random Forest
IR	Infrared
GPS	Global Positioning System

Chapter 1

Introduction

1.1. Background

Since all systems are prone to failure, an engineer's responsibilities involve ensuring that crucial systems never fail and therefore that proper testing and continual maintenance is performed. In the case of satellites, the problem proves even more severe, since most failures are unrecoverable. Satellite systems must therefore be tested thoroughly and be robust to any anomaly. According to [1], the attitude and orbit control system (AOCS) contributes to the largest percentage of satellite failures as shown in Figure 1.1. A study conducted by [3] on small satellite mission failures provides a deeper insight into the attitude control's role in satellite failures, since most missions are highly dependent on the attitude control for complete mission success. This is due to many mission specifications that rely on accurate control of the satellite to ensure that payloads, such as cameras, are able to operate as required.

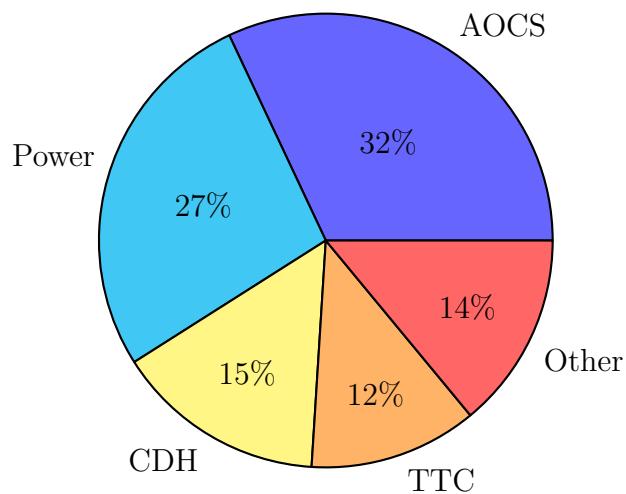


Figure 1.1: On-orbit failure per subsystem [1]

The database provided by [4] 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 satellite constellations.

Consequently, most aspects of the satellite should operate autonomously, especially in the case of attitude control. The focus of this thesis is on a specific aspect of the attitude control which is sensitive to anomalies, namely the attitude determination.

The extended Kalman filter (EKF) is implemented for attitude estimation of the satellite. The objective of this thesis is therefore to develop methods that can be used to avoid unstable and inaccurate estimations from the EKF caused by sensor anomalies.

1.2. Problem Description

For many satellite missions, attitude determination and control system (ADCS) is of high importance. It is necessary to effectively control the attitude to fulfill the mission requirements. The control performance is also limited by both the estimation accuracy and performance. A classical satellite attitude requirement is to be earth-following during an eclipse, to point the payload to a target and to otherwise point and track the sun for solar charging. Good attitude estimation throughout the entire orbit is necessary to best fulfill the requirements.

The attitude estimation is, however, highly influenced by the sensor readings. Different sensor measurements are fused together, normally with the use of an EKF, to produce a single attitude estimate. If an erroneous, or false measurement is present in the collection of sensors, it will deter and influence the outcome of the fusion algorithm. Depending on the number of sensors available and the size of the error, the result can either only be reduced estimation accuracy or divergence and instability also. It is therefore good practice to develop appropriate tests to protect the EKF against incorrect measurements.

Anomaly detection in satellite sensors have been investigated in many previous research. The current trend is to use generic sensor anomalies, such as bias drift, high noise, sudden failure or any drastic change in the behavior of the sensor and to develop techniques to detect these anomalies. 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. Majority of satellites, even with relatively low attitude requirements, have some form of sun sensor. The sun sensor also 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.

Since the EKF is also 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 also be detected and isolated to ensure robust estimation. The satellite must be able to autonomously detect,

classify (isolate) and recover from the anomaly to ensure safe operation during orbit. 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.

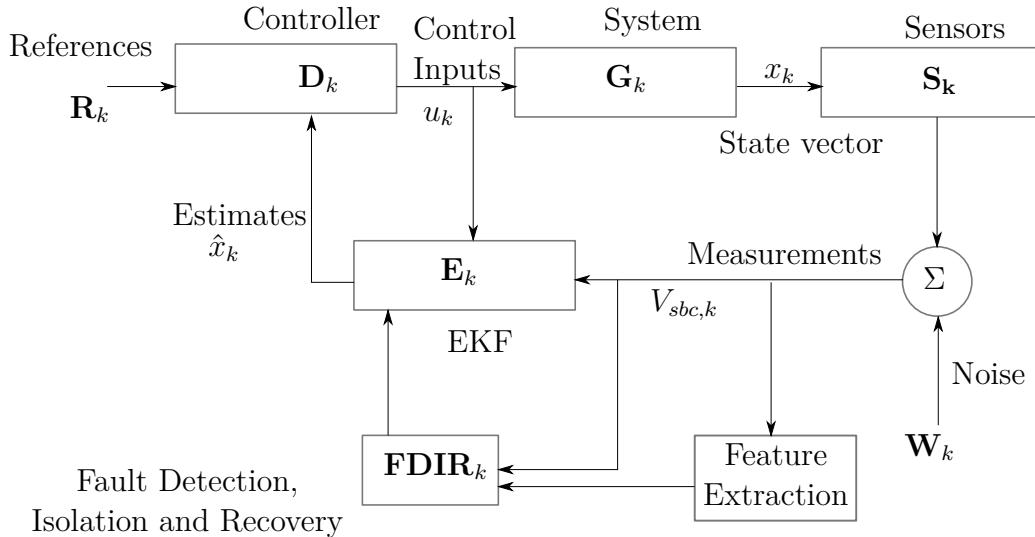


Figure 1.2: System Diagram

The anomalies discussed and modelled in this thesis are specific to the design of the satellite. The attitude sensors and the anomalies for each sensor is a sun sensor with solar reflection from the solar panels, a infrared-nadir sensor with the moon on the earth's horizon and a magnetometer with magnetic disturbances caused by the magnetic induced dipole moment of the solar panels. The actuator failure is that of a reaction wheel not responding to control inputs. These anomalies should be accurately detected and classified to ensure a reliable EKF. The recovery methods after classification will be discussed in chapter 8.

1.3. Project definition

This project aims to develop a simulation model wherein practical anomalies are simulated during satellite orbit. The simulation must then be 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 classification models for detection and isolation, respectively. The trained models should be tested on the simulation environment and the estimation accuracy should be compared between different models and different recovery methods.

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 should also be discussed. A thorough analysis of the each individual component of the FDIR should be conducted and discussed.

1.4. Thesis outline

Chapter 1 provides the background and motivation for this research as well as the project definition and thesis outline.

Chapter 2 discusses the relevant research that has been done on FDIR.

Chapter 4 demonstrates the development and implementation of the simulation environment, \mathbf{G}_k as shown in Figure 1.2.

Chapter 5 discusses the satellite design and expands on the elements \mathbf{D}_k , \mathbf{S}_k , \mathbf{E}_k and \mathbf{W}_k in Figure 1.2.

Chapter 6 provides the mathematical models of the specific anomalies and the effects thereof on the satellite.

Chapter 7 describes the feature extraction methods used in this thesis to enhance the accuracy of the detection and classification models, also provided as a element in Figure 1.2.

Chapter 8 provides various recovery methods and demonstrates the theoretical possibility of the methods based on perfect prediction accuracy, the final component of \mathbf{FDIR}_k in Figure 1.2.

Chapter 9 describes the different algorithms and methods used to detect an anomaly in the system, the first component of \mathbf{FDIR}_k in Figure 1.2.

Chapter 10 describes the different algorithms and methods used to classify an anomaly in the system, the second component of \mathbf{FDIR}_k in Figure 1.2.

Chapter 11 provides a summary of all the results for the combination of best methods as provided in Chapter 8, Chapter 9 and Chapter 10.

Chapter 12 discusses the influence of modelling specific anomalies on the prediction accuracy and robustness of a EKF.

Chapter 2

Literature Study

Defining the research space wherein this thesis fits, requires the explanation of the larger FDIR field. This field includes any industry with time-varying systems that require fault-tolerant control. 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. The even narrower research space is fault-tolerant control with the focus on Kalman filters and sensor anomalies.

2.1. Fault Detection Isolation and Recovery in General

Anomaly detection is a well researched field in robotics, that can be divided into three main categories, namely data-driven, model-based and *expert-system-based* approaches [5, 6]. Where the latter is the simplest way of anomaly detection and is implemented with human knowledge based logic tests or rules, also known as rule-based [7, 8]. Expert-system-based approaches are traditionally implemented in most 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.

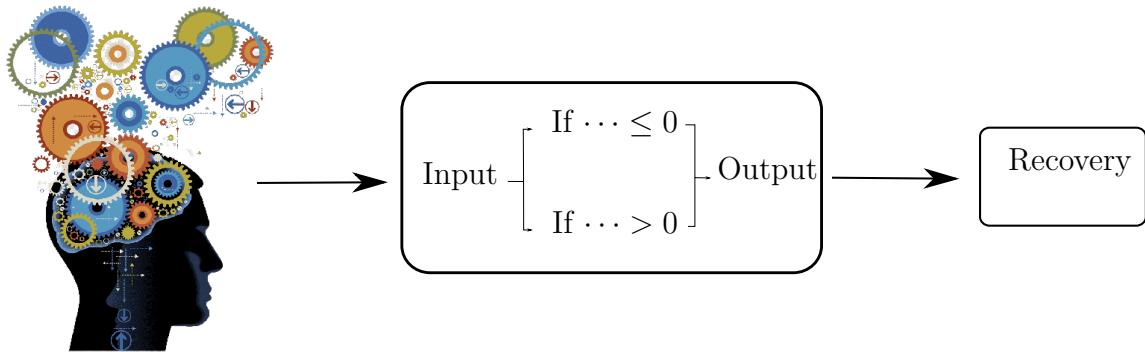


Figure 2.1: Development of FDIR technique according to expert-system-based approach.

Expert-based methods however are very closely aligned to model based and sometimes the lines between the two categories are blurred. This is due to many expert-system-based methods relying on the model and sensor inputs to ensure that the system is behaving as required. This is therefore the reason why many expert-system-based compare sensor

measurements with a mathematical model of that sensor measurement to develop a fault detection system.

Expert-system-based systems are required to create different rules for each sensor's implementation, and 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 in consideration. Data-driven approaches let one 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. 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 neighbors, Long Short Term Memory, (LSTM), Kalman Filters, Decision Trees and Isolation Forests that is used depending on the nature of the data [9–11]. The decision of which method to implement relies highly on whether the data is based on a time-series model or not. It also depends on whether labelled data is available, with certain methods requiring labelled data to predict whether or not data samples are anomalous. The details of the methods used for anomaly detection is discussed in Chapter 9.

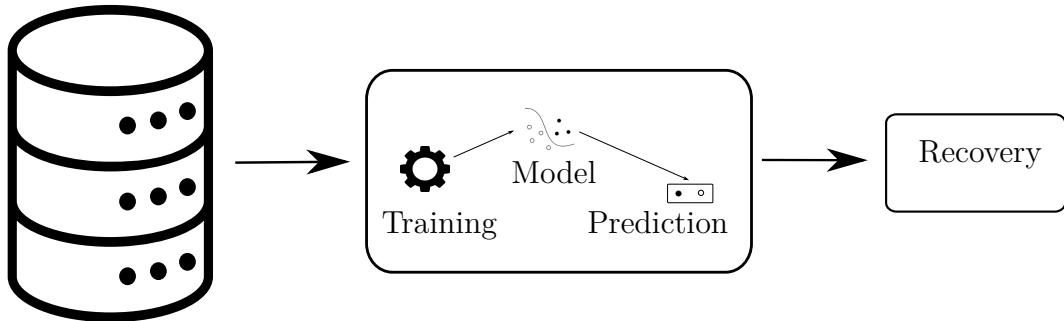


Figure 2.2: Development of FDIR technique according to data-driven approach.

This data base can be either simulated data or data from actual satellite missions. The problem with the actual satellite data is that it is difficult and expensive to get the specific data required for practical anomalies and to have it labelled. Therefore many research opt for simulation environments to implement and test the developed methods for FDIR.

2.2. Fault Tolerant Control of Satellites

Traditionally the expert-based approach combine with the model-based approach dominated satellite fault detection. An expert in the field typically developed a rule-based system for each failure that the expert is aware of [7]. 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 most 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. An example of a subsystem that is dependent on the ADCS for success is the power subsystem. since it requires accurate pointing of the solar panels towards the sun.

This leads to the reasoning 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 for failed actuators or robust estimation for failed sensors. The failure of the sensor or actuator is not necessarily complete failure, but influences the mission success of the satellite. To ensure robust control after actuator failure or recovering thereof for reaction wheels, gyroscopes and other actuators [12–17].

[13] developed a data driven method for fault isolation that incorporates random forests, decision trees, and nearest neighbors to classify failures of reaction wheels. These failures are modelled for time-varying faults where the reaction wheels bus voltage or current is varied to induce failures. This however only focuses on isolating the failures and not recovering from these failures, whereas [14] 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 is implemented to recover from modelled reaction wheel failures such as viscous friction variations due to temperature [12]. Both [12] and [13] provide two different implementations for fault tolerant control based on practical modelled actuator failure on satellites. Where [14] tests the provided method on a bias fault condition and not a practical modelled failure.

This aspect of fault tolerance is of high importance, however to ensure a successful mission, the attitude determination of the satellite should also be able to recover from anomalies. Sensor failures should therefore also be detected and recovered from.

2.3. Fault Tolerant Control with Focus on Sensor Anomalies

To ensure that a satellite is accurately controlled, the estimation of the satellite attitude is required. It is for this reason that the prediction of sensor anomalies and the recovery thereof is crucial, since it directly influence the attitude estimation. Actuator faults also influence the estimation and fault control of satellites with the focus on robust estimation is demonstrated in Figure 2.3. Most research to ensure robust estimation only focuses on sensor faults.

Based on the assumptions of typical sensor failures some work has been done on the fault detection of attitude sensors. For instance, due to considerable noise in sensors, [18]

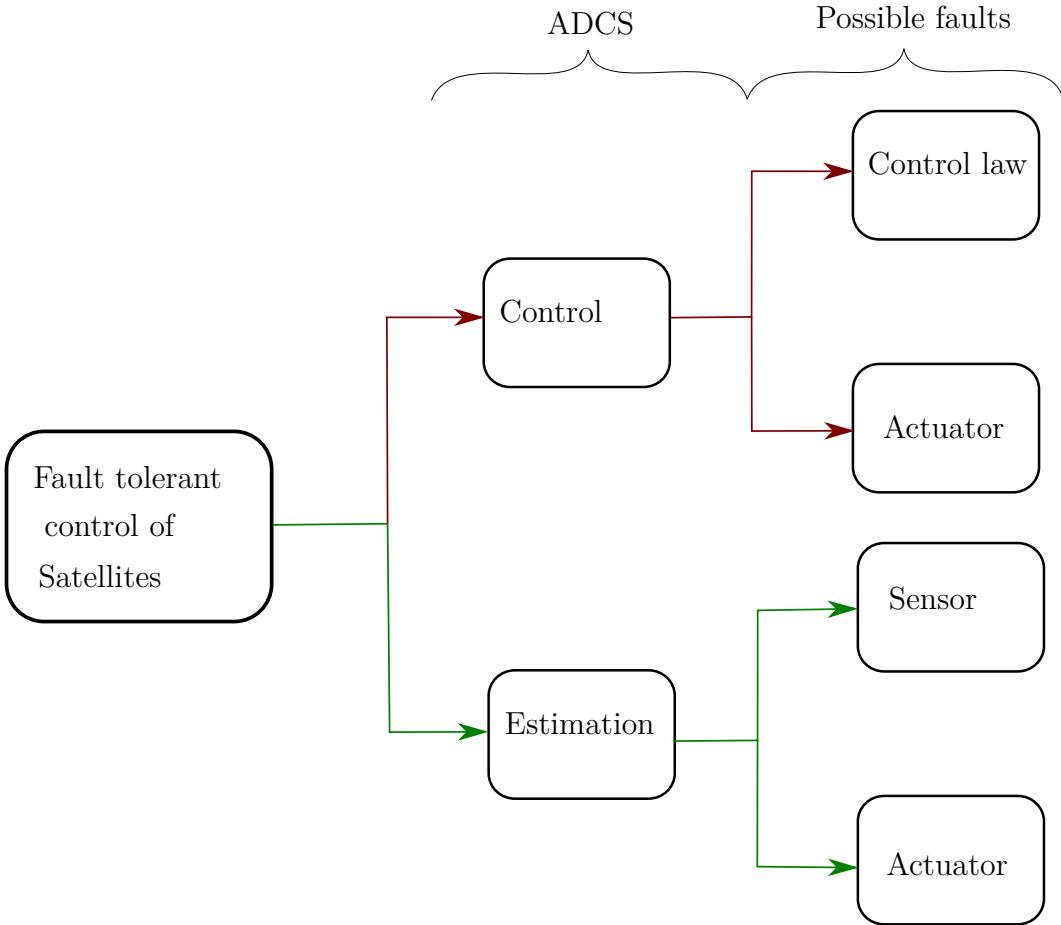


Figure 2.3: Development of FDIR technique according to data-driven approach.

proposed an adaptive unscented Kalman filter with multiple-model adaptive estimation for sensor fault estimation and isolation. The performance of this method is tested on a simulation model where gradual failures, abrupt failures and high noise is implemented on the sensor. [19] 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 is tested on a simulation environment where a sun sensor, earth sensor and gyroscope is used for attitude determination. A sudden bias failure for the sun sensor, earth sensor and gyroscope is implemented as well as an incipient fault on the sun sensor. [20] implement a fault tolerant federated Kalman filter with three sub-filters for multi-sensor fault estimation. The failures are invalid outputs where the measured vectors are equal to **0**, constant bias faults as well as noise amplification. [21] provide a fault detection and recovery method by implementing a non-linear observer, to detect anomalies in attitude and rate sensors. The recovery is implemented through the tuning of controller gains after the classification of sensor failures. Another example is development of an algorithm to evaluate the control of a gyroscope and detect whether drifting exists [22]. If drifting is detected another algorithm is deployed to ensure the recovery of the gyroscope drift by updating the error state vector. [23] developed an adaptive modification to the EKF with

the testing thereof on an aircraft. The faults for testing this methods is that of oscillation, bias drift and increased noise.

All these sensor fault detection methods only test the methods on general sensor faults and not specific modelled faults for each sensor. This thesis focuses on specific anomalies for each sensor and not general anomalies as given in the examples [18–23]. This is visually demonstrated in Figure 2.4, where the norm of sensor fault detection is the red path of general anomalies, while this thesis is the green path of the diagram which test methods on practical modelled anomalies.

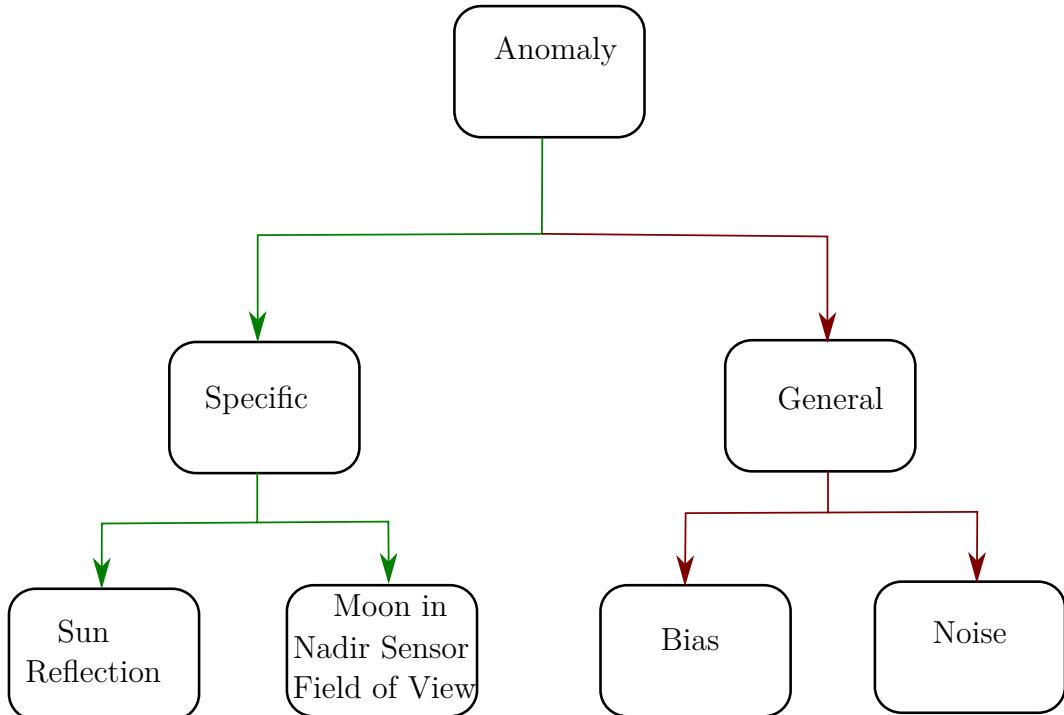


Figure 2.4: Development of FDIR technique according to data-driven approach.

2.3.1. Innovation in Fault Detection

Fault detection in sensor and developing robust Kalman filters is not isolated to that of satellites. All systems governed by underlying physics have many similarities in the approach of fault detection. [24] developed a novel method for feature extraction that 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, determined by the error estimated with dynamic mode decomposition, (DMD), and a Kalman filter. This is provided as additional input to a predictive model – decision tree, to detect sensor anomalies. This method is suitable for satellite attitude sensors, due to the underlying physics that governs the system.

2.4. Evaluation

Methods based on the research of various authors [18–23] however, are tested on sensor failures that are not modelled by the orbital nature of the satellite or specific design failures. The failures are limited to the sudden failure, bias drift or an increase in sensor noise. On the basis of [24] this thesis provides a methodology of fault detection, isolation and recovery through extraction anomalous features and providing it as an additional input to various classification algorithms. This work is an example of implementing the anomaly detection on the sensor level and diagnosing which sensor is experiencing the anomaly. Therefore, 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 be excluded from the measurement update sequence.

This method is furthermore not tested on a generic sensor failures, but a specific practical failure mode for each individual sensor that is used during the measurement update of the EKF. This provides more relevant and specific analysis of the method and this can be extended to the testing of this method on other modelled failures.

2.5. Summary

Just as [14] provides results from general actuator anomalies and applies that to reaction wheels, the sensor anomaly detection research implement general sensor anomalies on specific sensors. This is the difference in this thesis that the anomalies of sensors are specifically modelled for each sensor and general sensor anomalies are not implemented to test the different classification and recovery methods.

Chapter 3

Satellite Design

To simulate the ADCS of the satellite the sensors and actuators that will be used for the CubeSat must be decided. This decision is based on the current trend and implementation of CubeSat as well as what is required of this mission of pointing the payload towards the Earth during eclipse and being sun-following otherwise.

The trend from 1990 to 2015 of what sensors and actuators small satellites implement for the ADCS system is provided in Figure 3.1 and Figure 3.2 [25]. A study conducted by [3] provides examples of 2 partially failed missions that are due to sensor failure. One of which is ascribe to magnetometer faults and one towards that of the sun sensor. Another total mission failure is due to the failure of both the magnetometer and sun sensor. There is however no example of partial or total mission failure due to any other sensor failure. It should however be noted that the sun sensor and magnetometer are more frequently implemented in the period of [3] research as shown in Figure 3.1. Since most satellites implement a magnetometer and a sun sensor, 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 small satellites ADCS operations has an interesting anomaly [26]. The anomaly of the horizon sensor is the moon on the earth horizon and influencing the algorithm for calculating the centre of the Earth. This anomaly will occur often and is unavoidable through the CubeSat design. This anomaly will therefore will be modelled and the influence of the anomaly on the sensor measurement will be discussed.

The percentage of ADCS systems that utilizes a variety of actuators is visualised in Figure 3.2. It is evident Figure 3.2 that the two actuators that is utilized the most is the magnetorquers and reaction wheels. Therefore these two actuators are implemented for the control output of the ADCS. However since the focus of this thesis is to test various methods for robust estimation and the reaction wheels will have a significant influence on the estimation, whereas the magnetorquers will only be implemented for momentum dumping. An anomaly for reaction wheels will therefore be discussed, but the magnetorquers will be assumed to perform perfectly during all operations.

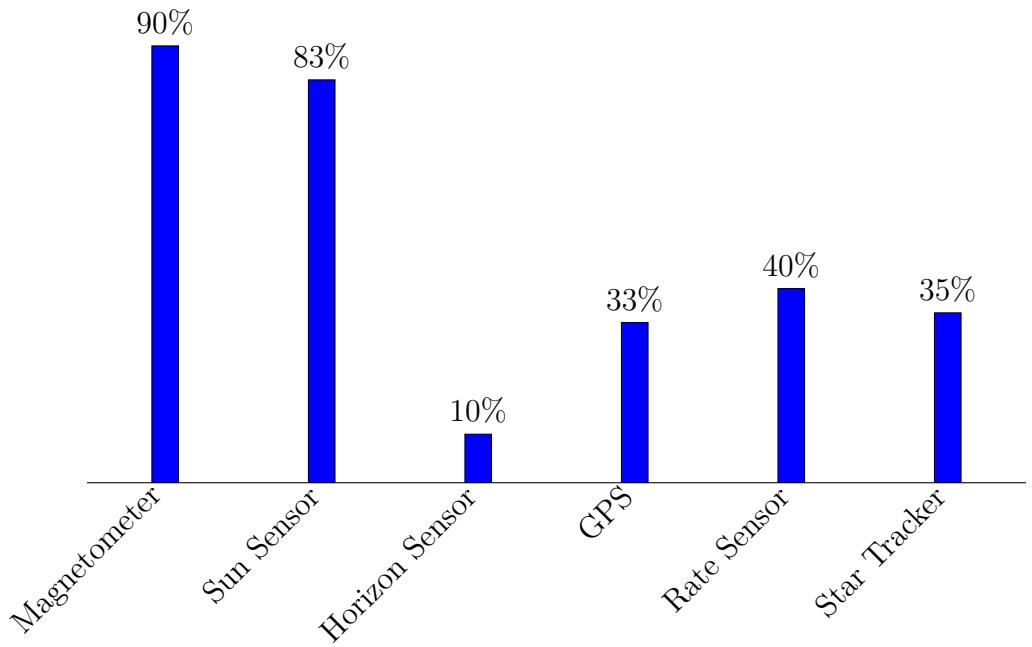


Figure 3.1: The percentage occurrences of different ADCS sensors on small satellites

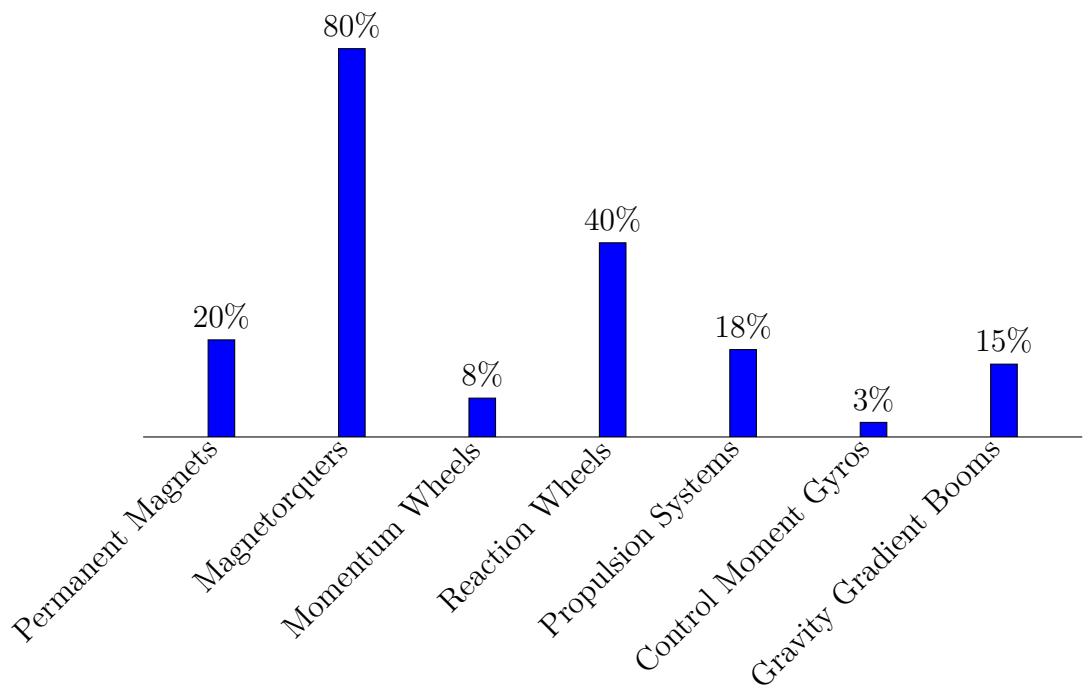


Figure 3.2: The percentage occurrences of different ADCS actuators on small satellites

3.1. CubeSat

To ensure accurate modelling and simulation of the satellite orbit as well as anomaly modelling a outer design of the CubeSat is required. The elements that required specific dimensions for the dynamics and kinematics is the solar panels and the satellite body. The sun sensor also requires specific dimensions for the sun reflection anomaly discussed in Section 6.1 and the dimensions of the sun sensor are from the Sputnix CubeSat sun sensor model [27]. The summary of these dimensions is given in Table 3.1.

Table 3.1: Physical dimensions of simulated CubeSat

Dimensions	Satellite Body	Solar Panels	Sun Sensor
\bar{x}_B	0.3 m	0.3 m	0.028 m
\bar{y}_B	0.3 m	0.3 m	0.023 m
\bar{z}_B	0.4 m	0.002 m	N/A

An entire solar panel is denoted as the surface area of ABCD as shown in Figure 3.3. This will be referenced for the sun reflection anomaly as well as the magnetic moment disturbance anomaly. The coordinate frames and the background thereof will be discussed in Section 4.1.1.

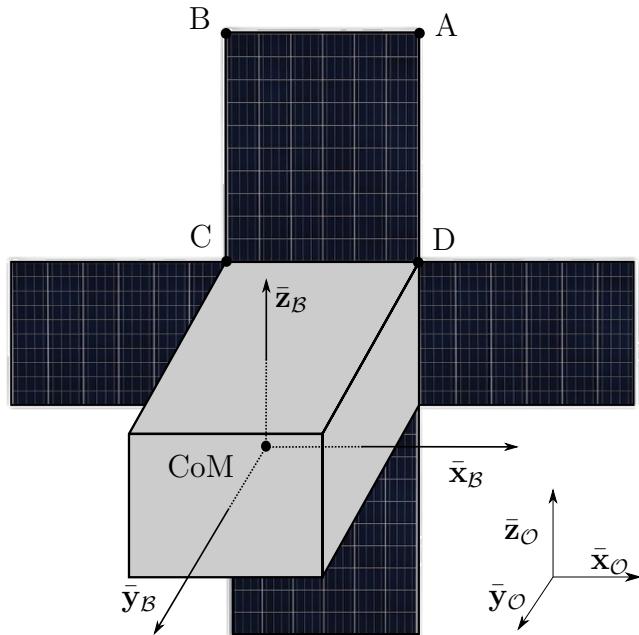


Figure 3.3: The modelled satellite, with solar panels.

3.2. Sensors

To ensure that the models of the sensors, which is largely only focused on the standard deviation of the zero-mean Gaussian noise as well as the the anomaly modelled in Chapter 6. The standard deviation of the noise as well as the position for each sensor in the satellite body coordinate (SBC) frame, as discussed in Section 4.1.1, is therefore given in Table 3.2. The sensor model is only discussed in Section 5.1 after the theoretical background of the satellite simulation. **TODO: Make sure positions are correct**

Table 3.2: Standard deviation for each sensor

Sensor	Standard deviation (σ)	Position $\{\bar{x}_B, \bar{y}_B, \bar{z}_B\}$
Magnetometer	0.75 [28]	$\{0.15, 0.15, 0\}$ m
Horizon Sensor	0.14 [26]	$\{0, 0.15, 0\}$ m
Sun Sensor	0.055 [27]	$\{0.15, 0.15, 0\}$ m

3.3. Actuators

The magnetorquers and reaction wheels both have maximum torque outputs as well as parameters that influence disturbance models. The reaction wheel is the RW3-0.06 from Sinclair Interplanetary [29] and the magnetorquer is the medium CubeTorquer by CubeSat [30]. The parameters for that of the reaction wheel is the static and dynamic wheel imbalance required to simulated the wheel imbalance discussed in Section 4.3.3. The static imbalance and dynamic imbalance is given as $U_s = 0.0208$ gcm and $U_d = 0.0208$ gcm² 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 $I_w = 88.1 \times 10^{-6}$ kgm². The magnetic moment of the magnetorquer is 0.66 Am².

Three magnetorquers as well as three reaction wheels are implemented. A single magnetorquer and reaction wheel are position in each SBC axis. This allows for three-axis rotation and control of the satellite.

Chapter 4

Simulation Environment

In order to implement and research various FDIR systems on satellites, a simulation of satellite dynamics and kinematics is developed. The focus of this thesis is directed towards small satellites and, more specifically, CubeSats. During the development of the satellite simulation, [25,31–33] were frequently referenced. The simulation was developed in Python to simulate the dynamics and kinematics during a satellite orbit.

4.1. Satellite Orbit 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. To ensure this is accurately simulated, the different coordinate frames dominating a satellite orbit, the attitude of the satellite, as well as the satellite dynamics and kinematics is discussed in this section.

4.1.1. Coordinate Frames

The coordinate frames in aerospace is a fundamental part of the ADCS. In order to determine the orientation and position of an object, it should be relative to a fixed frame. Consequently, the Earth inertial coordinate (EIC) frame, $\mathcal{E}\{\bar{\mathbf{x}}_{\mathcal{E}}, \bar{\mathbf{y}}_{\mathcal{E}}, \bar{\mathbf{z}}_{\mathcal{E}}\}$, is the fixed frame from which every other frame is relative to.

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 $\bar{\mathbf{z}}_{\mathcal{A}}$. The axes of the coordinate frame is appropriately named as the X-axis, Y-axis and Z-axis. A vector ($\mathbf{r}_{\mathcal{A}}$) within the coordinate frame can thus be expressed as

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

where the magnitude of \mathbf{r} , denoted as $\|\mathbf{r}\|$, is equal to

$$\|\mathbf{r}\| = \sqrt{a^2 + b^2 + c^2}. \quad (4.2)$$

The Earth-centered coordinate frames are divided into two, namely the EIC and Earth fixed coordinate (EFC) frame, $\mathcal{F}\{\bar{\mathbf{x}}_{\mathcal{F}}, \bar{\mathbf{y}}_{\mathcal{F}}, \bar{\mathbf{z}}_{\mathcal{F}}\}$. EFC is fixed to the Earth and rotates

with it, while EIC is inertial fixed.

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. The EFC is a copy of the EIC, with the Z-axis being identical. The EFC, however, rotates with the Earth. The EFC in relation to the EIC can be expressed by a single angle of rotation, which is the Greenwich Hour Angle (GHA), α_G . With the elapsed time, t , since t_0 , the angular rate of the Earth, ω_E , and the GHA, α_{G_0} , at $t = t_0$ known, α_G can be calculated as

$$\alpha_G = \omega_E t + \alpha_{G_0}. \quad (4.3)$$

To transform a vector from one coordinate frame to another, a transformation matrix, \mathbf{A} , is required. For example vector \mathbf{r}_F can be transformed to \mathbf{r}_E with

$$\mathbf{r}_E = \mathbf{A}_F^E \mathbf{r}_F \quad (4.4)$$

with \mathbf{A}_F^E being the EFC-to-EIC transformation matrix. Due to the definition of both coordinate frames, \mathbf{A}_F^E can be defined as

$$\mathbf{A}_F^E = \begin{bmatrix} \cos(\alpha_G) & -\sin(\alpha_G) & 0 \\ \sin(\alpha_G) & \cos(\alpha_G) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.5)$$

To determine the satellite position, satellite-centred coordinate frames must be used. Three satellite-centred coordinate frames are used, namely the inertial-reference coordinate frame (IRC), $\mathcal{I}\{\bar{\mathbf{x}}_{\mathcal{I}}, \bar{\mathbf{y}}_{\mathcal{I}}, \bar{\mathbf{z}}_{\mathcal{I}}\}$, which remains inertial fixed, 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 ORC frame changes location as the satellite move. The Z-axis is, however, always pointing towards the centre of the Earth, with the Y-axis being the orbit anti-normal and the X-axis completing the orthogonal set. To transform a vector from the EIC frame to the ORC frame the unit position vector, \mathbf{r}_{sat} and the unit velocity vector, \mathbf{v}_{sat} in EIC is required [34]. The EIC to ORC transformation matrix, \mathbf{A}_E^O , is calculate as

$$\begin{aligned} \mathbf{A}_E^O &= [\mathbf{u} \ \mathbf{v} \ \mathbf{w}]^T \\ \text{where } \mathbf{w} &= -\frac{\mathbf{r}_{sat}}{\|\mathbf{r}_{sat}\|} \\ \mathbf{v} &= -\frac{\mathbf{r}_{sat} \times \mathbf{v}_{sat}}{\|\mathbf{r}_{sat} \times \mathbf{v}_{sat}\|} \\ \mathbf{u} &= \mathbf{v} \times \mathbf{w}. \end{aligned} \quad (4.6)$$

The SBC frame is the frame fixed to the satellite and it is the relative rotation of the satellite in relation to the ORC. Thus for the mission of this satellite it is required that the SBC and ORC frames coincide during an eclipse. For the transformation of a vector

from the ORC to SBC frame, the direct cosine matrix (DCM), also referred to as \mathbf{A} or $\mathbf{A}_{\mathcal{O}}^{\mathcal{B}}$, is used. For the remainder of the document the DCM will be referred to as $\mathbf{A}_{\mathcal{O}}^{\mathcal{B}}$ to avoid any confusion. The calculation of this transformation matrix is discussed in §4.1.2 and implemented with Eq 4.10.

4.1.2. Attitude

To determine the attitude of an object, a mathematical model must be used to determine the rotation of an object in three dimensions. For this the visual and intuitive example of the Euler angles exist. Euler angles are the rotation of an object around three orthogonal axes which change orientation with the rotation of the object. The three axes of SBC, denoted by $\bar{\mathbf{x}}_{\mathcal{B}}$, $\bar{\mathbf{y}}_{\mathcal{B}}$ and $\bar{\mathbf{z}}_{\mathcal{B}}$ rotate with the object as depicted in Figure 4.1.

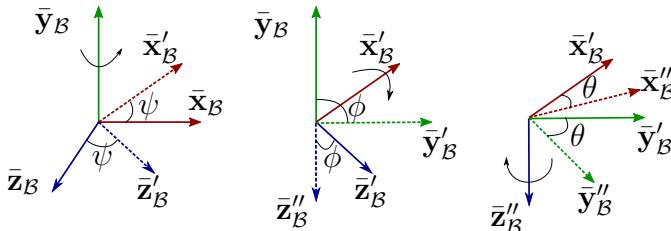


Figure 4.1: Euler angles

$\mathbf{A}_{\mathcal{O}}^{\mathcal{B}}$ 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. $\mathbf{A}_{\mathcal{O}}^{\mathcal{B}}$ can therefore be calculated as

$$\begin{aligned} \mathbf{A}_{\mathcal{O}}^{\mathcal{B}} &= \mathbf{A}_{\psi} \mathbf{A}_{\phi} \mathbf{A}_{\theta} \\ &= \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}. \end{aligned} \quad (4.7)$$

Euler angles, however, does 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 [35]. The method of describing 3D rotation with quaternions is therefore more convenient and more often used.

A quaternion, \mathbf{q} , has four components that are dependent on one another and constrained by

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \quad (4.8)$$

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 $\mathbf{e} = [e_1 \ e_2 \ e_3]^T$ and the angle of the

Euler rotation is Φ then \mathbf{q} can be expressed as

$$\mathbf{q} = \begin{bmatrix} e_1 \sin\left(\frac{\Phi}{2}\right) \\ e_2 \sin\left(\frac{\Phi}{2}\right) \\ e_3 \sin\left(\frac{\Phi}{2}\right) \\ \cos\left(\frac{\Phi}{2}\right) \end{bmatrix} \quad (4.9)$$

Although it is difficult to visualize a quaternion, the most simplistic method of conceptualising it is shown in Figure 4.2. 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 4.1 the angle θ is the angle of rotation around the \bar{z}_B'' -axis. For quaternions the angle of rotation is the same principle, however, the axis around which the object is rotating, is the unit vector, q_{1-3} . Therefore, q_4 provides the angle of rotation while q_{1-3} represents the unit vector, however, with the condition of Eq 4.8.

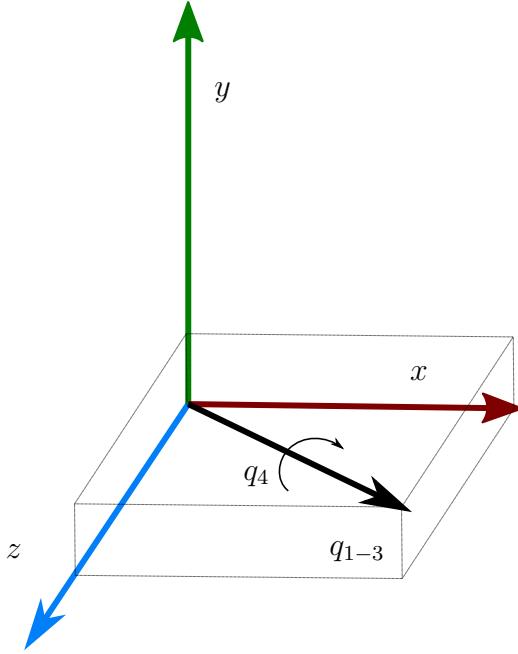


Figure 4.2: Graphical quaternion representation

\mathbf{A}_O^B can also be transformed as a function of \mathbf{q} [36] through

$$\mathbf{A}_O^B = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_1 q_2 - q_3 q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}. \quad (4.10)$$

The quaternion is used for attitude determination and therefore also for the attitude control. A error between the commanded quaternion, \mathbf{q}_c and the current quaternion, \mathbf{q} , is required for proportional control. This is discussed in section 5.3.1.

4.1.3. Satellite Kinematics and Dynamics

The conservation of momentum dominates the dynamics of a satellite. This consists of the torques applied to the satellite and are mainly control torques, \mathbf{N}_c , or disturbance torques, \mathbf{N}_d , as well as the moment of inertia of the satellite, \mathbf{I} , multiplied by the inertial-referenced angular acceleration of the satellite, $\dot{\omega}_B^T$. The control torques used in this design are only reaction wheel torques, \mathbf{N}_w , and magnetorquer torques, \mathbf{N}_m . The disturbance torques are discussed in detail in section 4.3. It can therefore only be mentioned that the disturbance torques are the gravity gradient torque, \mathbf{N}_{gg} , the wheel imbalance torque, \mathbf{N}_{rw} , the gyroscopic coupling torque, \mathbf{N}_{gyro} , and the aerodynamic disturbance torque, \mathbf{N}_{aero} . The Euler dynamic equation can therefore be given as

$$\mathbf{J}\dot{\omega}_B^T = \mathbf{N}_c + \mathbf{N}_d,$$

where $\mathbf{N}_d \approx \mathbf{N}_{aero} - \mathbf{N}_{gyro} + \mathbf{N}_{gg} + \mathbf{N}_{rw}$,

and $\mathbf{N}_c = \mathbf{N}_m - \mathbf{N}_w$.

(4.11)

This is the overarching equation that will be used to determine the control torque as well as the model update of the EKF. The integration method to solve the differential equations used in the simulation is the 4th order Runge-Kutta method. This is demonstrated with Algorithm 4.1.

Algorithm 4.1: 4th order Runge-Kutta

```

1: Definitions:  $T_s$  - Timestamp;
2:  $h = T_s/I$ 
3: for  $n := 1$  to  $I$  do
4:    $k_1 = hf(x_n, y_n)$ 
5:    $k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$ 
6:    $k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})$ 
7:    $k_4 = hf(x_n + h, y_n + k_3)$ 
8:    $y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}$ 
9: end for
```

Where h is the step size, I is the number of iterations set to 10 and the time step, T_s , is equal to one second. $f(x_n, y_n)$ is the Euler dynamic function and Algorithm 4.1 is used to calculate $\dot{\omega}_B^T$. With this procedure the dynamics and kinematics of the satellite can be simulated after each time step.

4.2. Environment

To ensure an accurate simulation environment, certain aerospace phenomena must be simulated to create an accurate representation of the satellite orbit and therefore ensure that all anomalies can be accurately modelled. The position of the satellite with respect

to the Earth (orbit propagation) is therefore required to determine most of the elementary principles of the satellite mission. The orbit propagation is further used to determine the moon on the Earth's horizon anomaly. The sun position is required for an eclipse as well as simulating the sun reflecting from the solar panels unto the sun sensor. The magnetic field is required to simulate \mathbf{N}_m of the magnetorquer as well as the solar panel dipole anomaly and disturbance torque.

4.2.1. Orbit Propagation

The satellite position, \mathbf{r}_{sat} and velocity \mathbf{v}_{sat} at a given time step is required to determine the multiple different variables required for the simulation environment. Therefore the refined version and fourth generation of the simplified general perturbations (SGP) model, namely SGP4, is used as orbit propagator of the satellite after each time step [37].

To determine \mathbf{r}_{sat_k} and \mathbf{v}_{sat_k} at 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, that requires parameters such as the semimajor axis, a , right ascension of the ascending node (RAAN), Ω , argument of perigee (AP), ω , inclination, i , eccentricity, e , and the time at the beginning of the orbit as a Julian date, J_t . With these parameters and the elapsed time since J_t , both \mathbf{r}_{sat_k} and \mathbf{v}_{sat_k} can be determined from the World Geodetic System 72 constants that is implemented through the SGP4 model. An example of a satellite orbit propagated by the SGP4 model is illustrated in Figure 4.3.

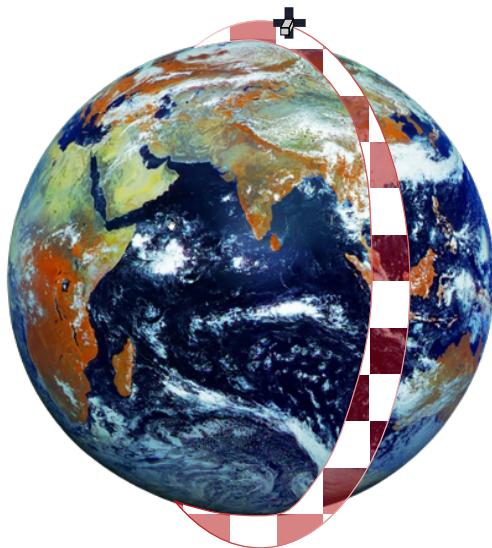


Figure 4.3: SGP4 orbit propagation

The SGP4 is implemented with the SGP4 Python package [38]. The SGP4 outputs the \mathbf{r}_{sat_k} and \mathbf{v}_{sat_k} in the EIC reference frame. Therefore, with \mathbf{r}_{sat} and \mathbf{v}_{sat} known, $\mathbf{A}_{\mathcal{E}}^{\mathcal{O}}$ can now be calculated according to Eq 4.6.

Table 4.1: Orbit parameters for simulated CubeSat

Parameter	Value
Revolutions per day	15.2355
Inclination	97.4°
Right ascension of the ascending node	275°

4.2.2. Sun

For the 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 control operation. Therefore, the model from [39] is 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. This model requires various calculations as given in Eq 4.13. 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

$$T_{JC} = \frac{J_t - 2\,452\,545}{36\,525}. \quad (4.12)$$

With T_{JC} known, \mathbf{r}_{sun} can then be calculated with

$$\mathbf{r}_{sun} = r_{\oplus} \begin{bmatrix} \cos(\lambda_e) \\ \cos(\epsilon)\sin(\lambda_e) \\ \sin(\epsilon)\sin(\lambda_e) \end{bmatrix},$$

$$\text{where } r_{\oplus} = 1.000\,140\,612 - 0.016\,708\,617 \cos(M_{\oplus}) - 0.001\,395\,89 \cos(2M_{\oplus}), \quad (4.13)$$

$$M_{\oplus} = 357.527\,723\,300^{\circ} + 35\,999.050\,340 T_{JC},$$

$$\lambda_e = \lambda_{M_{\oplus}} + 1.914\,666\,471 \sin(M_{\oplus}) + 0.019\,994\,643 \sin(2M_{\oplus}),$$

$$\lambda_{M_{\oplus}} = 280.460\,618\,400^{\circ} + 36\,000.770\,053\,610 T_{JC},$$

$$\text{and } \epsilon = 23.439\,291^{\circ} - 0.013\,004\,200 T_{JC}.$$

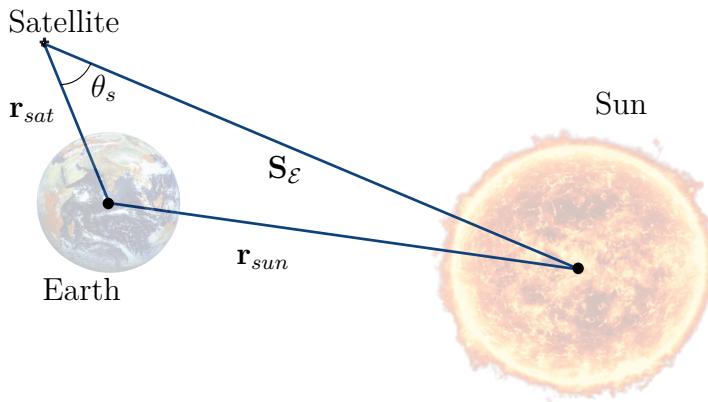
The definitions of the parameters used in the calculation and the description thereof is tabulated in Table 4.2. After determining the sun position, it is crucial to calculate whether the satellite is in an eclipse or not. This can be done with basic geometry after calculating the position of the sun relative to the satellite through

$$\mathbf{S}_{\mathcal{E}} = \mathbf{r}_{sun} - \mathbf{r}_{sat}. \quad (4.14)$$

Table 4.2: Description and definition of Earth orbit parameters

Symbol	Definition	Description
r_{\oplus}	Sun 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 Earth position.
M_{\oplus}	Mean anomaly	The fraction of the orbit's period after the Earth 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. [40]

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 very small angle required to change the satellite from a partial eclipse to a full eclipse, due to the comparative distances of the sun to satellite and satellite to Earth. The eclipse is therefore defined as the period during which θ_s is smaller than θ_E . Where $\theta_E = \sin(\frac{R_E}{\|r_{sat}\|})$ and $\theta_s = \mathbf{r}_{sat} \cdot \mathbf{S}_{\mathcal{E}}$ as shown in Figure 4.4. R_E is the radius of the Earth.

**Figure 4.4:** Geometry for satellite eclipse

4.2.3. Geomagnetic field

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. The magnetorquers interact with the geomagnetic field for momentum dumping and the magnetometers measure the geomagnetic field for attitude estimation. The modelling of the geomagnetic field is therefore required for an accurate simulation environment.

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 13th generation of the model is implemented [41]. The scalar potential function,

$$V(r_s, \theta, \phi, t) = R_E \sum_{n=1}^N \left(\frac{R_E}{r_s} \right)^{n+1} \sum_{m=0}^n (g_n^m(t) \cos(m\phi) + h_n^m(t) \sin(m\phi)) P_n^m(\cos\theta), \quad (4.15)$$

is used to calculate the geomagnetic field, \mathbf{B} with

$$\mathbf{B} = -\nabla V. \quad (4.16)$$

Therefore, the geomagnetic field is the gradient of the scalar potential function given in Eq 4.15. Where R_E is the mean Earth radius of 6371.2km, r_s is the radial distance from the centre of the Earth, θ is the latitude and ϕ is the longitude. $g_n^m(t)$ and $h_n^m(t)$ is known as the Gauss coefficients that slowly change with time and, consequently, the IGRF-13 provide values for these coefficients at 5-year epoch intervals. The $P_n^m(\cos\theta)$ is the Legendre functions of the degree n and m [42]. The magnitude of the geomagnetic field is visually demonstrated in Figure 4.5.

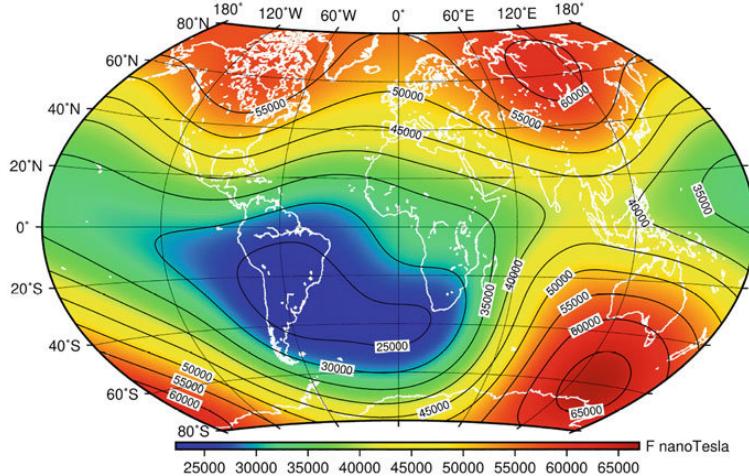


Figure 4.5: The magnitude of geomagnetic field according to the 13th generation of the IGRF model [2].

4.3. Disturbance models

During orbit, a satellite is exposed to various disturbance torques. 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. Other disturbances that occur only with anomalies are discussed in Chapter 6. Some disturbances are excluded from the simulation environment and only the major sources of

disturbance torques are modelled and discussed.

The first disturbance torque is that of the gyroscopic coupling which can be calculated with

$$\mathbf{N}_{gyro} = \dot{\boldsymbol{\omega}}_{\mathcal{B}}^T \times (\mathbf{I}\dot{\boldsymbol{\omega}}_{\mathcal{B}}^T + \mathbf{h}_w), \quad (4.17)$$

where \mathbf{h}_w is the angular momentum of the reaction wheels. The other disturbance torques are discussed in more detail below.

4.3.1. Gravity Gradient

The gravity gradient is 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 of the satellite, while the part of the satellite furthest from the Earth will experience the smallest gravitational force and the largest centrifugal force. According to [36] the gravity gradient disturbance torque, \mathbf{N}_{gg} can be calculated as

$$\mathbf{N}_{gg} = 3\omega_{\mathcal{O}}^2(\mathbf{z}_{\mathcal{B}} \times \mathbf{I}\mathbf{z}_{\mathcal{B}}). \quad (4.18)$$

The orbit nadir vector is calculated as,

$$\mathbf{z}_{\mathcal{B}} = \mathbf{A}_{\mathcal{O}}^{\mathcal{B}} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T. \quad (4.19)$$

Due to the design of the satellite and the positions of the solar panels, the equation for \mathbf{N}_{gg} can not be simplified. 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. \mathbf{N}_{gg} in SBC is shown in Figure 4.6.

TODO: Change all plots x, y, and z should be in `mathbf` and the colors should correspond to that of the pitch figure.

4.3.2. Aerodynamic Disturbance

The aerodynamic disturbance torques are cause by air density in the atmosphere creating a force on each individual segment of the satellite [43]. This is significant due to the low Earth orbit (LEO) of the satellite, where the atmospheric density is higher. The aerodynamic disturbance torque, \mathbf{N}_{aero} is therefore a summation of all the torques created by the air force on each segment area, A_i . \mathbf{N}_{aero} can therefore be calculated as

$$\begin{aligned} \mathbf{N}_{aero} = \sum_{i=1}^n & \left(\rho \|\mathbf{v}_{\mathcal{B}}\|^2 A_i H \{\cos(\alpha_i)\} \cos(\alpha_i) (\sigma_t (\mathbf{r}_{pi} \times \bar{\mathbf{v}}_{\mathcal{B}}) \right. \\ & \left. + [\sigma_n S + (2 - \sigma_n - \sigma_t) \cos(\alpha_i)] (\mathbf{r}_{pi} \times \bar{\mathbf{n}}_i)) \right), \end{aligned} \quad (4.20)$$

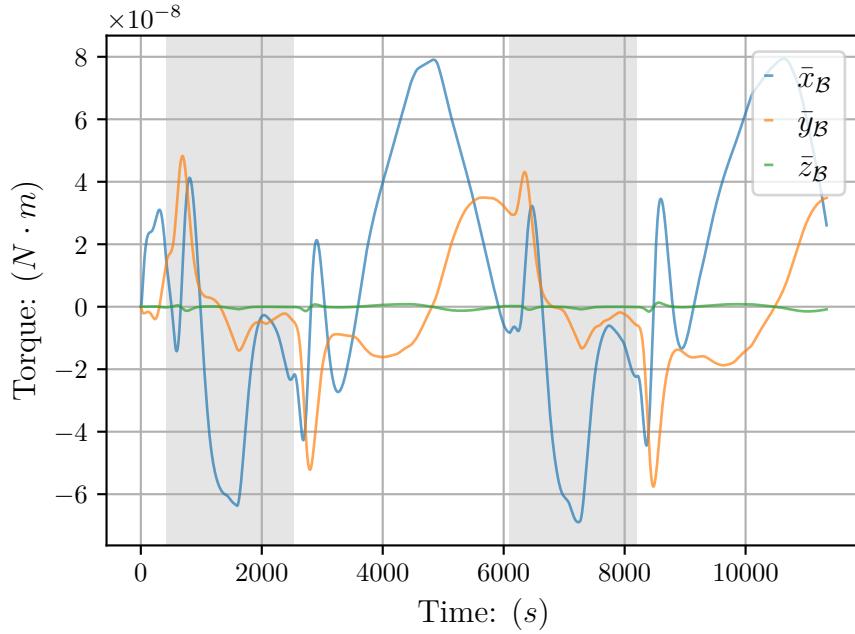


Figure 4.6: Gravity Gradient Torque in SBC

where n is the number of segments of the satellite. The factors that influence the aerodynamic disturbance torque is the atmospheric velocity in SBC, \mathbf{v}_B , the atmospheric density, ρ , each segment's surface area, A_i , and the offset vector between the segment's centre of mass (CoM) and the centre of pressure (CoP), \mathbf{r}_p . $H\{x\}$ is the Heaviside function which is equal to 0 when x is smaller than 0 and otherwise equal to 1. α_i is the incidence angle of \mathbf{v}_B on segment i , while σ_t is the tangential accommodation coefficient and σ_n is the normal accommodation coefficient. S is the ratio of molecular exit velocity to \mathbf{v}_B and $\bar{\mathbf{n}}_i$ is the unit inward normal vector of segment i [25].

The atmospheric density is a function of the distance from the Earth surface. The density model provided by [39] is given as

$$\rho = \rho_o e^{-\frac{h(t)-h_o}{H}}, \quad (4.21)$$

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 [44] the atmospheric density is $\frac{1}{2}\rho$ during an eclipse. Furthermore \mathbf{v}_B is calculated as

$$\mathbf{v}_B = \mathbf{A}_O^B \mathbf{A}_E^O \mathbf{v}_E$$

where $\mathbf{v}_E = \begin{bmatrix} 0 \\ 0 \\ \omega_E \end{bmatrix} \times \mathbf{r}_{sat} - \mathbf{v}_{sat}.$ (4.22)

σ_n and σ_t are both assumed to be equal to 0.8, while S is 0.8 [44]. From these equations

the aerodynamic disturbance can be calculated and is shown in Figure 4.7.

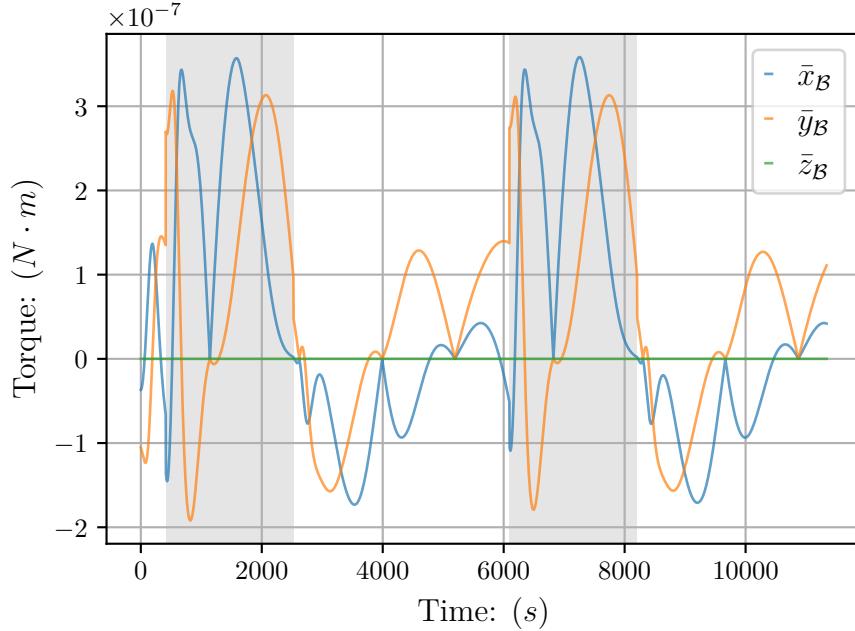


Figure 4.7: Aerodynamic Torques in SBC

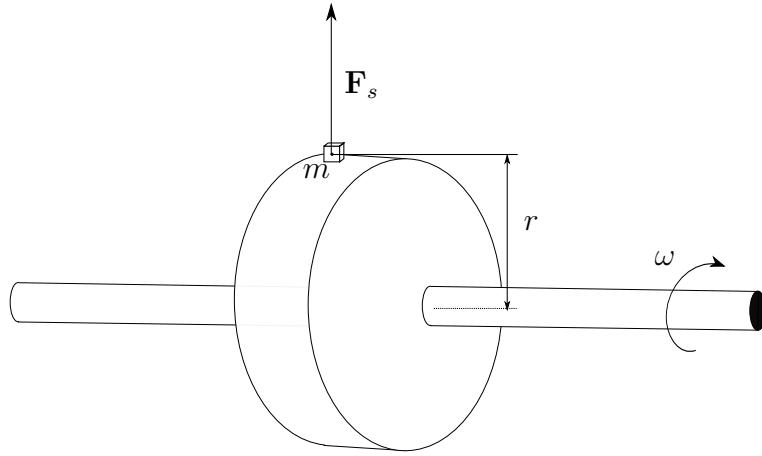
4.3.3. Wheel Imbalance

Since the reaction wheel imbalance is considered to be the most significant disturbance on the reaction wheel, it is the only disturbance torque modelled for this simulation [45]. 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 displaced.

The static imbalance of the reaction wheel is caused by the reaction wheel CoM offset from the rotational axis. Therefore, to model the static imbalance of the reaction wheels it is 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 provided by the reaction wheel manufacturers.

To determine the resulting torque from the wheel imbalance, the torque generated by each wheel is individually calculated. Consequently, for the reaction wheel in direction \bar{x}_B , $RW_{\bar{x}B}$, the force, $F_{s\bar{x}B}$, generated by U_s is dependent on the angular rate, ω , of the reaction wheel as well as the current position of m . Therefore, $F_{s\bar{x}B}$ can be expressed as

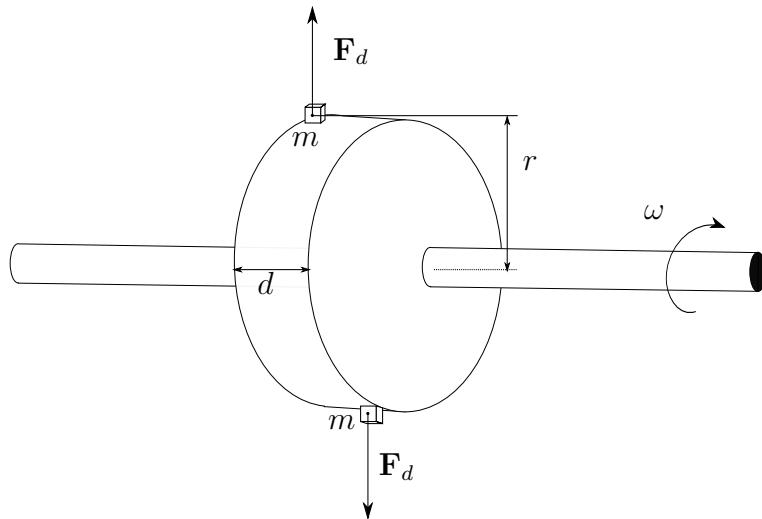
$$\mathbf{F}_{s\bar{x}B} = U_s \omega^2 \begin{bmatrix} 0 \\ \sin(\omega t + \phi_s) \\ \cos(\omega t + \phi_s) \end{bmatrix}, \quad (4.23)$$

**Figure 4.8:** Static Imbalance

where the current angle of m is defined as $\omega t + \phi_s$, with ϕ_s as an arbitrary phase and for the sake of simplification is set to 0. With $F_{s\bar{x}_B}$ exerted on $RW_{\bar{x}_B}$ known, the torque on the satellite can be calculated with the known position vector, $\mathbf{w}_{\bar{x}_B}$, of $F_{s\bar{x}_B}$ to the satellite CoM. Therefore, $N_{s\bar{x}_B}$ can be calculated as

$$\mathbf{N}_{s\bar{x}_B} = \mathbf{w}_{\bar{x}_B} \times F_{s\bar{x}_B}. \quad (4.24)$$

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 180° apart 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.

**Figure 4.9:** Dynamic Imbalance

The dynamic wheel imbalance torque, $\mathbf{N}_{d\bar{x}_B}$, for $RW_{\bar{x}_B}$ can be calculated as

$$\mathbf{N}_{d\bar{x}_B} = U_d \omega^2 \begin{bmatrix} 0 \\ \sin(\omega t + \phi_d) \\ \cos(\omega t + \phi_d) \end{bmatrix}. \quad (4.25)$$

where $U_d = mrd$ as the dynamic imbalance. Both U_d and U_s are provided by the manufacturer and based on the reaction wheel as discussed in Section ???. The net wheel imbalance torque from both the static and dynamic wheel imbalance is provided in Figure 4.10.

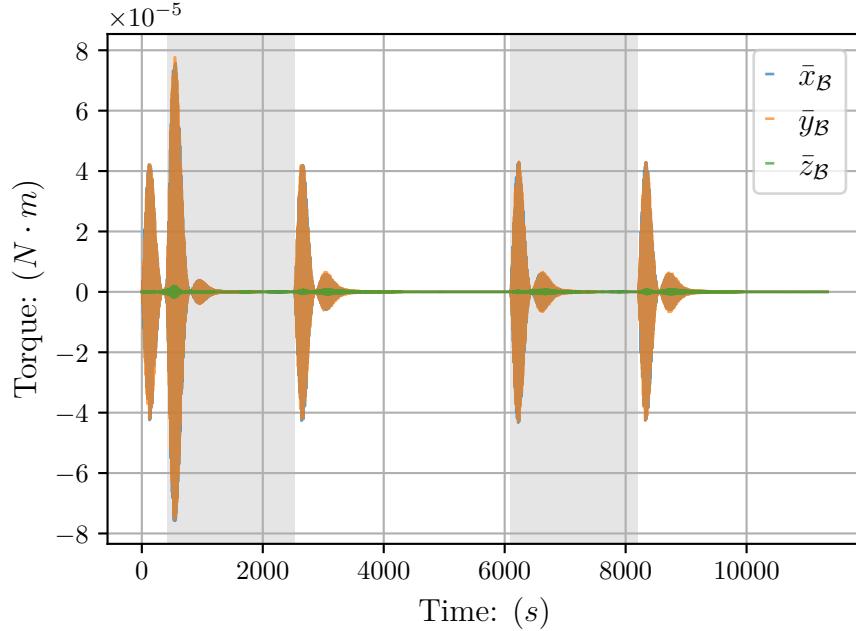


Figure 4.10: Wheel disturbance torques in SBC

Chapter 5

Attitude Determination and Control System

This chapter discusses the design of the ADCS. Therefore, the position of the actuators and sensors are discussed as well as the EKF algorithm and control methods for eclipse and sunlit phase.

5.1. 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. Therefore, each sensor's position on the satellite is provided in Section 3.2. 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 in the SBC frame, \mathbf{v}_B , can be calculated as

$$\mathbf{v}_B = \mathbf{A}_O^B \mathbf{v}_O + \mathbf{m}_v, \quad (5.1)$$

where \mathbf{m}_v is the measurement noise of the current sensor and \mathbf{v}_I is the reference ORC vector. The measured unit vectors of the sun as an example of the sensor measurements is shown in Figure 5.1 where 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.

5.2. Attitude Determination

This section discusses the estimation 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 physical models to estimate the current attitude. The EKF is highly sensitive to sensor anomalies and actuator failures and this section discusses the implementation of the EKF.

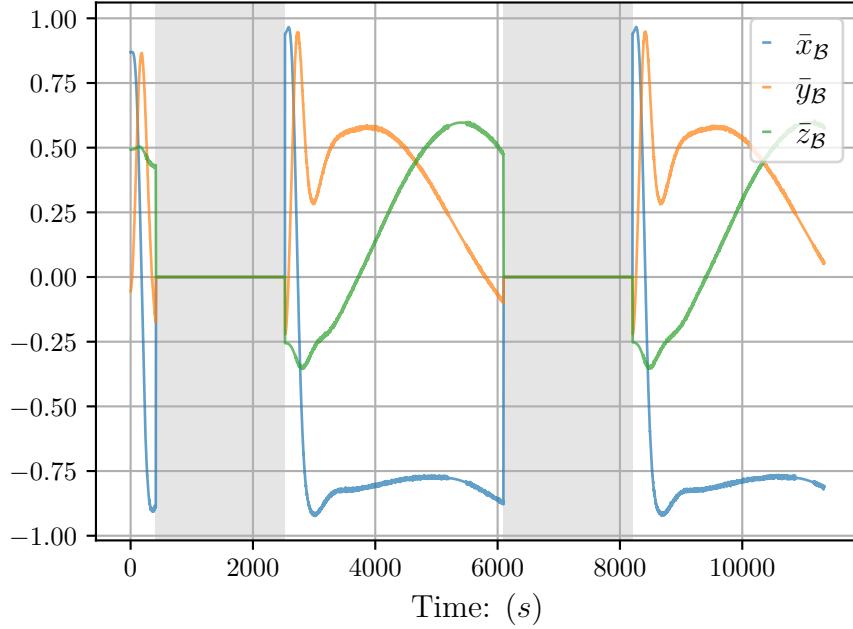


Figure 5.1: Sun vector in SBC

5.2.1. Extended Kalman Filter

The implementation of the EKF is for estimation of the current satellite attitude with sensor fusion of the magnetometer, star tracker, sun sensor and nadir sensor to accurately estimate the attitude and rotation rate of the satellite. The EKF will be used due to the non-linear nature of the system. The EKF consists of two fundamental parts, the model update and the measurement update. The estimated state vector, \mathbf{x} , will be denoted as $\hat{\mathbf{x}}$ and the 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

$$\dot{\hat{\mathbf{x}}}_t = \mathbf{f}(\mathbf{x}_t) + s_t, \quad (5.2)$$

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

$$\begin{aligned} \mathbf{f}(x_t) &= \mathbf{f}(\hat{\mathbf{x}}_t) + \left[\frac{\partial \mathbf{f}}{\partial \hat{\mathbf{x}}_t} \right] (\mathbf{x}_t - \hat{\mathbf{x}}_t) + \frac{1}{2!} \left[\frac{\partial^2 \mathbf{f}}{\partial \hat{\mathbf{x}}_t^2} \right] (\mathbf{x}_t - \hat{\mathbf{x}}_t)^2 \\ &\approx \mathbf{f}(\hat{\mathbf{x}}_t) + \mathbf{F} \Delta \mathbf{x}_t, \end{aligned} \quad (5.3)$$

where $\mathbf{F}_t = \left[\frac{\partial \mathbf{f}}{\partial \hat{\mathbf{x}}_t} \right]$

and $\Delta \mathbf{x}_t = \mathbf{x}_t - \hat{\mathbf{x}}_t$.

To linearise the state vector, \mathbf{x} , for the full 7-state EKF consists of the quaternion

vector, \mathbf{q} and the inertial-referenced angular velocity, $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}$ given as

$$\mathbf{x} = [\mathbf{q}, \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}]^T. \quad (5.4)$$

To calculate the model update the dynamics and kinematics of the system model is used to calculate both $\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{I}}$ and \mathbf{q} . The integration method used in the simulation is the 4th order Runge-Kutta method to solve the differential equations. The integration method is shown in Algorithm 5.2 where $(\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}})_k^-$ is calculated for the first step of the model update.

Algorithm 5.2: Runge-Kutta 4th order Algorithm at time k

```

1: Satellite Body Inertia  $\mathbf{J} = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.45 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$ 
2: Timestep ( $T_s$ ) = 1s
3: Number of iterations ( $I$ ) = 10
4: Step size  $h = \frac{T_s}{I}$ 
5: Disturbance torques  $\mathbf{N}_d = N_{gg} - N_{gyro}$ 
6: Control torques  $\mathbf{N}_c = N_m - N_w$ 
7:  $\mathbf{N} = \mathbf{N}_c + \mathbf{N}_d$ 
8: for  $n := 1$  to  $I$  do
9:    $k_1 = h(\mathbf{J}^{-1}\mathbf{N})$ 
10:   $k_2 = h(\mathbf{J}^{-1}\mathbf{N} + \frac{k_1}{2})$ 
11:   $k_3 = h(\mathbf{J}^{-1}\mathbf{N} + \frac{k_2}{2})$ 
12:   $k_4 = h(\mathbf{J}^{-1}\mathbf{N} + k_3)$ 
13:   $\boldsymbol{\omega}_{n+1} = \boldsymbol{\omega}_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}$ 
14: end for
15:  $(\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}})_k^- = \boldsymbol{\omega}_{n+1}$ 
16: return  $(\hat{\boldsymbol{\omega}}_{\mathcal{B}}^{\mathcal{I}})_k^-$ 

```

With reference to [25], $\hat{\mathbf{q}}_k^-$ can be calculated as

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

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

$$\begin{aligned} (\hat{\boldsymbol{\omega}}_{\mathcal{B}})_k^- &= (\hat{\boldsymbol{\omega}}_{\mathcal{B}})_k^{\mathcal{I}} - \hat{\mathbf{A}}_{\mathcal{O}_k}^{\mathcal{B}} \begin{bmatrix} 0 & -(\omega_{\mathcal{O}})_k & 0 \end{bmatrix}^T \\ &= [\hat{\boldsymbol{\omega}}_{\bar{x}_{\mathcal{O}}} \quad \hat{\boldsymbol{\omega}}_{\bar{y}_{\mathcal{O}}} \quad \hat{\boldsymbol{\omega}}_{\bar{z}_{\mathcal{O}}}]^T \end{aligned} \quad (5.5)$$

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

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

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

$$\hat{\mathbf{x}}_k^- = [(\hat{\boldsymbol{\omega}}_{\mathcal{B}})_k^{\mathcal{I}} \quad \hat{\mathbf{q}}_k^-] \quad (5.6)$$

Some of the parameters required for the calculations of the EKF will only be provided here in name and the derivations thereof is provided in the Appendices. The detail of each variable and the mathematical discussion thereof is beyond the scope of this thesis. The system noise covariance matrix, \mathbf{Q}_k , is assumed to be zero-mean and Gaussian and is derived in Appendix ???. The discrete system perturbation model, Φ_k is derived in Appendix ???. The discrete measurement perturbation Jacobian Matrix, \mathbf{H}_k is derived in Appendix ???. While the measurement noise covariance matrix, \mathbf{R}_k , is derived in Appendix ???.

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

$$\mathbf{P}_k^- = \Phi_k \mathbf{P}_{k-1}^+ \Phi_k^T. \quad (5.7)$$

The error, \mathbf{e}_k , between the measured and modelled vector is calculated as

$$\mathbf{e}_k = \mathbf{v}_{\mathcal{B}} - \hat{\mathbf{A}}_{\mathcal{O}_k}^{\mathcal{B}} \mathbf{v}_{\mathcal{O}}. \quad (5.8)$$

where $\mathbf{v}_{\mathcal{B}}$ is the measured vector in SBC and $\mathbf{v}_{\mathcal{O}}$ is the modelled ORC vector. The gain matrix \mathbf{K}_k is used to determine the influence of \mathbf{e}_k on the updated state vector, $\hat{\mathbf{x}}_k^+$. \mathbf{K}_k

can be calculated as

$$\mathbf{K}_k = \mathbf{P}_k^- (\mathbf{H}_k^-)^T \left[\mathbf{H}_k^- \mathbf{P}_k^- (\mathbf{H}_k^-)^T + \mathbf{R}_k \right]^{-1}. \quad (5.9)$$

after which the updated state vector can be calculated as

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{e}_k. \quad (5.10)$$

The state covariance matrix can then be updated as

$$\mathbf{P}_k^+ = [\mathbf{I}_{7 \times 7} - \mathbf{K}_k \mathbf{H}_k^+] \mathbf{P}_k [\mathbf{I}_{7 \times 7} - \mathbf{K}_k \mathbf{H}_k^+] + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T. \quad (5.11)$$

Where Φ_k is the discrete system perturbation model is calculated as

$$\begin{aligned} \Phi_k &= \left[e^{T_s \mathbf{F}_t} \right]_{\mathbf{x}=\hat{\mathbf{x}}, t=kT_s} \\ \text{and simplified as } \Phi_k &\approx \left[\mathbf{I} + T_s \mathbf{F}_t + \frac{1}{2!} T_s^2 \mathbf{F}_t^2 \right]_{\mathbf{x}=\hat{\mathbf{x}}, t=kT_s}, \end{aligned} \quad (5.12)$$

according to [46]. To validate the results of the EKF, the estimation error is given in Figure 5.2. Where the estimation error is the difference between the current \mathbf{q} and $\hat{\mathbf{q}}$ in degrees.

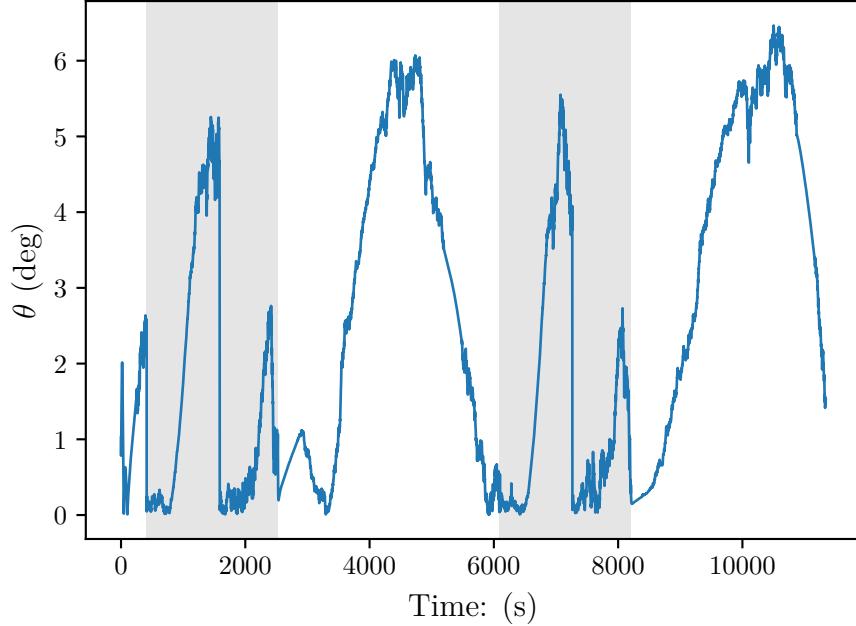


Figure 5.2: Estimation Metric

During the measurement update \mathbf{e}_k is largely affected by anomalous behaviour in the sensor measurements. The sensitivity of the Kalman filter to the various anomalies is discussed in Chapter 6.

5.3. Attitude Control

To ensure that the satellite is able to satisfy the mission requirements, control of the satellite attitude is required. Therefore, the satellite's payload must 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 of the reaction wheels is implemented and a momentum dumping with the magnetorquers is implemented to ensure that the wheel disturbance remains within reasonable boundaries.

5.3.1. 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 [47]. The controller is provided with $\hat{\mathbf{x}}$ as input and outputs \mathbf{N}_w . To calculate the required torque, \mathbf{N}_w , the definition according to [48] for all cases at time step, k , is given as

$$\mathbf{N}_w = K_{PI} \mathbf{J} \mathbf{q}_{err} + K_{DI} \mathbf{I} \hat{\omega}_{\mathcal{B}}^{\mathcal{O}} - \hat{\omega}_{\mathcal{B}}^{\mathcal{T}} \times [\mathbf{J} \hat{\omega}_{\mathcal{B}}^{\mathcal{T}} + \mathbf{h}_w], \quad (5.13)$$

where \mathbf{h}_w is the measured angular momentum of the wheels and the control gains are defined as

$$\begin{aligned} K_{PI} &= 2\omega_n^2 \\ K_{DI} &= 2\zeta\omega_n. \end{aligned} \quad (5.14)$$

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

$$\begin{aligned} \mathbf{q}_{err} &= \mathbf{q}_c \Theta \hat{\mathbf{q}} \\ \begin{bmatrix} q_{1e} \\ q_{2e} \\ q_{3e} \\ q_{4e} \end{bmatrix} &= \begin{bmatrix} q_{4c} & q_{3c} & -q_{4c} & -q_{4c} \\ -q_{3c} & q_{4c} & q_{1c} & -q_{2c} \\ q_{2c} & -q_{1c} & q_{4c} & -q_{3c} \\ q_{1c} & q_{2c} & q_{3c} & q_{4c} \end{bmatrix} \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \\ \hat{q}_4 \end{bmatrix}, \end{aligned} \quad (5.15)$$

where $\hat{\mathbf{q}}$ is the current estimated quaternion and \mathbf{q}_c is the command quaternion which is $[0 \ 0 \ 0 \ 1]^T$ during eclipse and during the sun following phase, the attitude command according to [49] can be calculated as

$$\mathbf{q}_c = \begin{bmatrix} \mathbf{u}_c \sin(\frac{\delta}{2}) \\ \cos(\frac{\delta}{2}) \end{bmatrix}, \quad (5.16)$$

where

$$\mathbf{u}_c = \frac{\mathbf{u}_{sp_{\mathcal{B}}} \times \mathbf{s}_{\mathcal{O}}}{\|\mathbf{u}_{sp_{\mathcal{B}}} \times \mathbf{s}_{\mathcal{O}}\|}. \quad (5.17)$$

$\mathbf{s}_{\mathcal{O}}$ is the measured unit sun vector in ORC, and the main solar panel's position is denoted as a unit vector, \mathbf{u}_{sp_B} . The angle between \mathbf{u}_{sp_B} and $\mathbf{s}_{\mathcal{O}}$, δ , can be calculated with the vector dot-product. This can then be used as the reference for the control. The reference ω_B^T is always $[0, 0, 0]$.

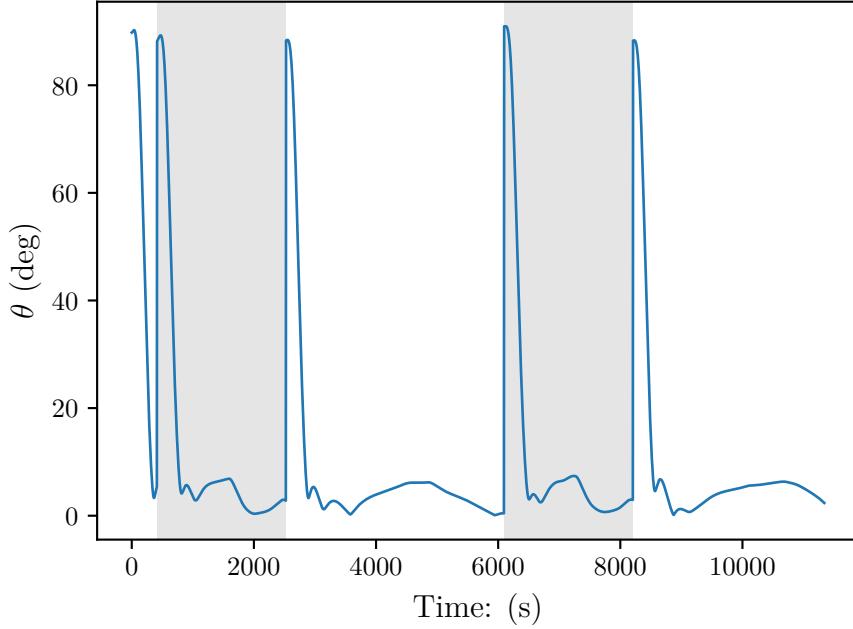


Figure 5.3: Pointing Metric

5.3.2. Momentum Dumping

Momentum dumping is crucial to ensure that the wheel disturbance does not cause the system to become unstable. Momentum dumping is implemented during eclipse after the satellite is in a stable nadir-pointing attitude. The momentum dumping is implemented with magnetic torquers based on a Cross-Product controller. The magnetic dipole moment \mathbf{M} is calculated as

$$\mathbf{M} = \frac{\mathbf{e} \times \mathbf{B}}{\|\mathbf{B}_b\|^2}, \quad (5.18)$$

where \mathbf{B}_b is the geomagnetic field and the error vector, \mathbf{e} can be calculated as

$$\mathbf{e} = -K_w(\mathbf{h}_w - \mathbf{h}_{w,\text{ref}}) \quad (5.19)$$

where K_w is a positive gain. This momentum dumping is implemented 200s after sun-following phase is implemented, to ensure stable control and reduce the momentum in the reaction wheels. The magnetorquers torques are shown in Figure 5.4 and 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.

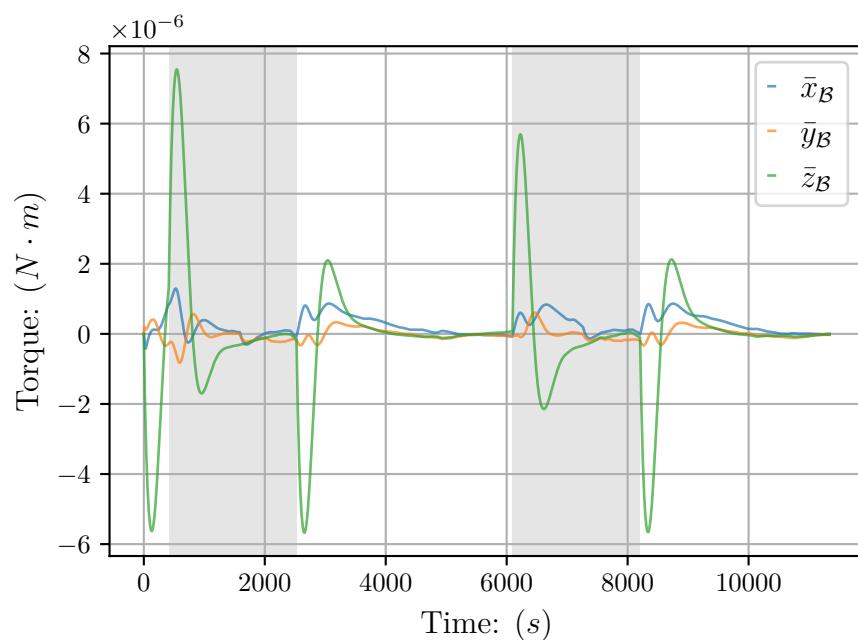


Figure 5.4: Magnetic Control Torques

Chapter 6

Anomalies

To ensure that the prediction and classification of anomalies are not based on generalised sensor failures a few anomalies are simulated. 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. There is a 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, \mathbf{N}_w and the torque implemented on the reaction wheel is not 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.

6.1. Reflection of Solar Panels on Sun Sensor

The 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, $\mathcal{B}\{\bar{x}_{\mathcal{B}}, \bar{y}_{\mathcal{B}}, \bar{z}_{\mathcal{B}}\}$.

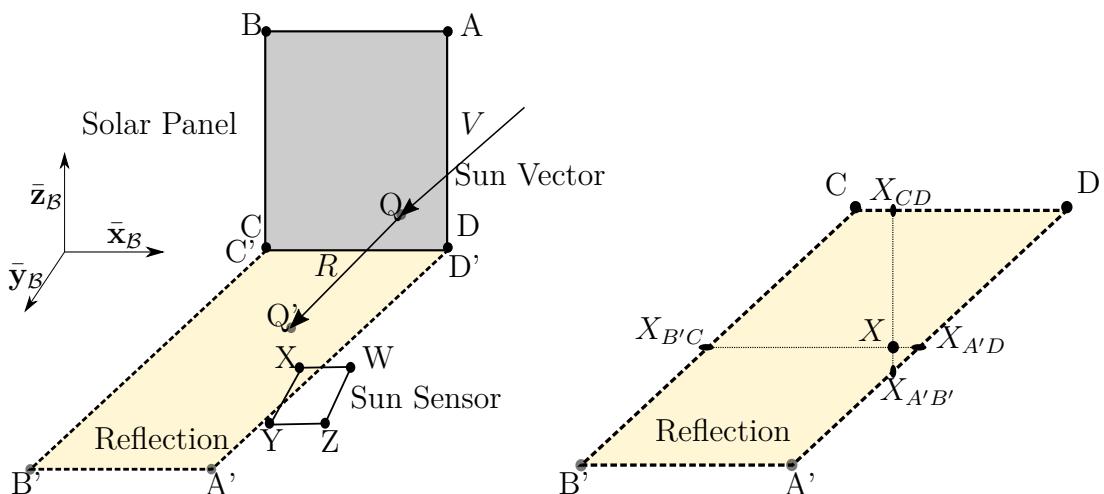


Figure 6.1: Definition of solar reflection point from $ABCD$ -plane.

The assumption is that the solar panel can be modeled as a geometric plane. Therefore, light from the solar panel will reflect similarly to 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 produce the worst-case behaviour. The intensity of the light vector is also disregarded.

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

$$\mathbf{p}_{ABCD} = [p_x, p_y, p_z]^\top, \text{ and} \\ \bar{\mathbf{n}}_{ABCD} = [n_p x, n_p y, n_p z]^\top, \quad (6.1)$$

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

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

$$\mathbf{r}_{\text{ref}} = \mathbf{r}_{\text{sun}} - 2\bar{\mathbf{n}}_{ABCD}^\top (\mathbf{r}_{\text{sun}} \cdot \bar{\mathbf{n}}_{ABCD}), \quad (6.2)$$

where \mathbf{r}_{sun} is the incoming sun vector. To calculate the intersection of the reflected vector with the $WXYZ$ -plane of the sun sensor, the equation of the $WXYZ$ -plane, the reflected vector, and the point of origin is required. The reflection of the sun vector is illustrated in Figure 6.1. The reflection from Q to Q' can thus be calculated as a projection of \mathbf{r}_{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 6.1 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. Therefore it is 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 R with plane $WXYZ$. We also know the position of A , based on the satellite design, and can therefore calculate A' . The same applies to B and B' . To determine whether Y or X is within the reflection region, we assume that the plane $WXYZ$ is a 2D plane, and we omit the third dimension. Therefore, the axis changes from x, y, z to only x, y . We calculate whether x is between the lines of $A'D$ and $B'C$ and between the lines CD and $A'B'$. By determining the line equation between reflected points in the form

$$y_{A'B'} = mx_{A'B'} + c, \quad (6.3)$$

from the coordinates of A' and B' , the corresponding $X_{A'B',y}$ can be calculated by substituting X_x into Eq 6.3. 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 and without reflection are shown in Figure 6.2. There is a clear difference between the true sun vector and the possible measurement influenced by the reflection. This false reflection vector may affect the estimation and, thus also the attitude control of the satellite.

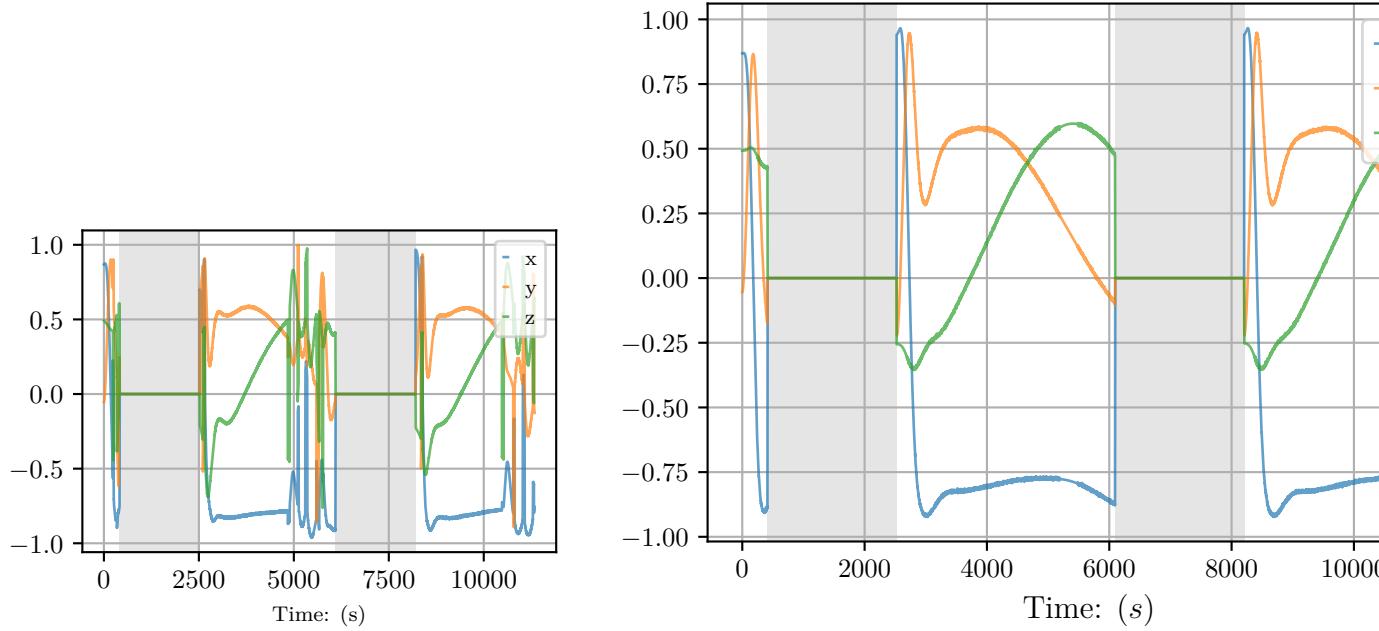


Figure 6.2: Comparison of measured sun vector with and without reflections (gray areas indicate the eclipse period of the orbit)

6.1.1. Influence of Anomaly on Estimation

To determine whether the reflection on the sun sensor has an influence on the ADCS, the estimation metric is shown in Figure 6.3. The estimation metric, is the angle difference between the actual attitude and the estimated attitude. It is evident that the reflection has a large influence on the estimation when Figure 6.3 is compared with Figure 5.2. The maximum estimation error is 6° for normal operation and it is evident in Figure 6.3 that the estimation error is sometimes even 150° . It is also clear that during the eclipse the estimation returns to a more accurate estimation. This is due to the fact that all sensors are ignored if the measured vector is 0 and the sun vector is 0 during eclipse.

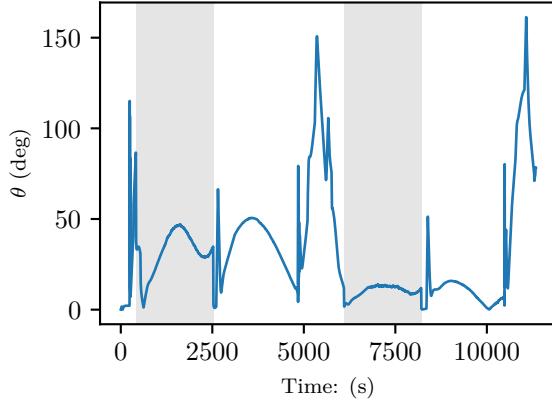


Figure 6.3: Estimation Metric with reflection on sun sensor

6.2. Moon in Field of View of Nadir Sensor

An anomaly that can be experienced by an infrared (IR) nadir sensor is the moon overlapping the horizon of the Earth in die nadir sensor's field of view (FoV). This is shown in Figure 6.6. This influences the edge detection and circular fit algorithm [26, 50] and consequently the calculated centre of the Earth. Firstly, it is required to simulate the image seen by the nadir sensor, thereafter the algorithm for detecting the centre of the Earth can be implemented.

6.2.1. Simulating Nadir Sensor Infra-red Image

Firstly, the vectors of both the satellite to Earth, \mathbf{R}_{SE} and the Earth to moon, \mathbf{R}_{EM} , is required. The moon 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 6.4.

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

$$P_{Earth} = [\bar{\mathbf{x}}_{\mathcal{E}_0}, \bar{\mathbf{y}}_{\mathcal{E}_0}, \bar{\mathbf{z}}_{\mathcal{E}_0}], \quad (6.4)$$

and

$$\mathbf{R}_{SE} = [\mathbf{n}_{\bar{\mathbf{x}}_{\mathcal{E}}}, \mathbf{n}_{\bar{\mathbf{y}}_{\mathcal{E}}}, \mathbf{n}_{\bar{\mathbf{z}}_{\mathcal{E}})]. \quad (6.5)$$

Therefore with the equation for the 3D plane defined as

$$A\bar{\mathbf{x}}_{\mathcal{E}} + B\bar{\mathbf{y}}_{\mathcal{E}} + C\bar{\mathbf{z}}_{\mathcal{E}} = Dm \quad (6.6)$$

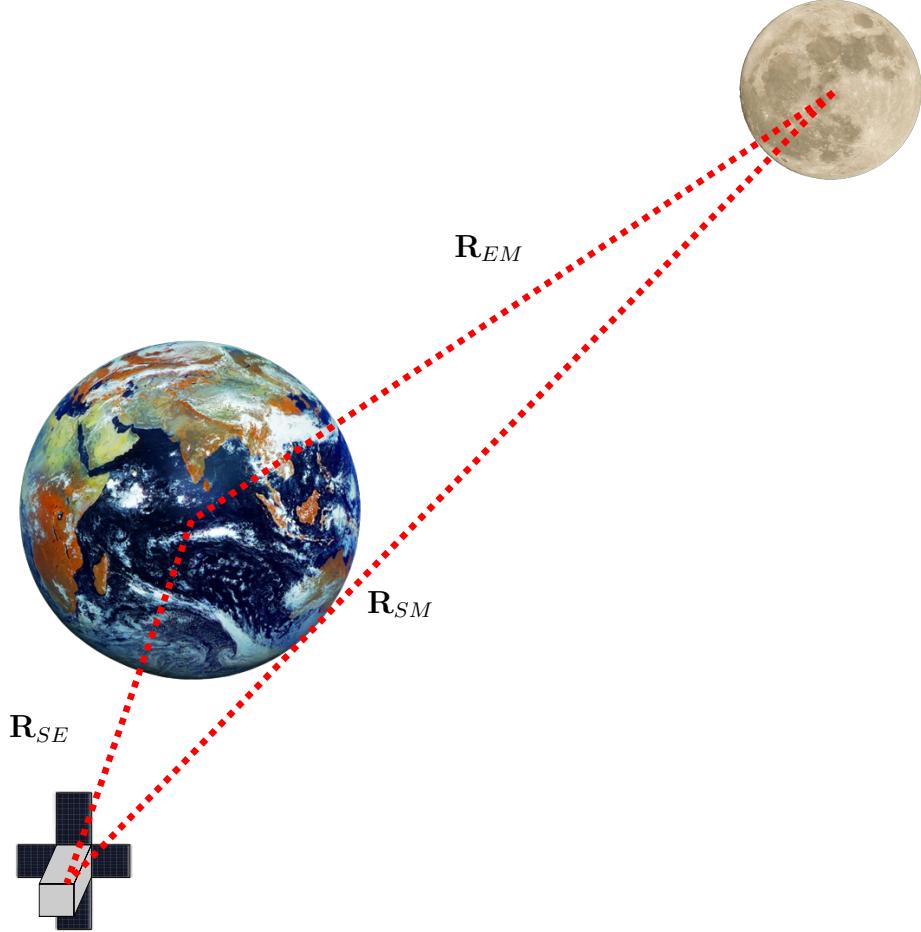


Figure 6.4: Earth to Moon and Earth to Satellite Vectors

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

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \mathbf{n}_{\bar{x}_E} \\ \mathbf{n}_{\bar{y}_E} \\ \mathbf{n}_{\bar{z}_E} \\ \mathbf{n}_{\bar{x}_E} \bar{x}_{E_0} + \mathbf{n}_{\bar{y}_E} \bar{y}_{E_0} + \mathbf{n}_{\bar{z}_E} \bar{z}_{E_0} \end{bmatrix} \quad (6.7)$$

This 3D plane slicing the Earth in half as seen from the satellite is shown in Figure 6.5. The moon and Earth can both be projected onto the 3D plane to determine the image seen by the nadir sensor. Therefore the nadir vector must also be projected onto the 3D plane. A circle can be drawn for the Earth, moon and the nadir sensor's FoV. The radius, of the moon as projected on the 3D plane can be calculated as

$$R_{moon} = \|\mathbf{R}_{SE}\| \frac{r_{moon}}{\|\mathbf{R}_{SM}\|} \quad (6.8)$$

the radius of the nadir sensor FoV, R_{FoV} can calculated as

$$R_{FoV} = \|\mathbf{R}_{SE}\| \tan(\theta) \quad (6.9)$$

With these variables defined and calculated, the edges of the moon and Earth within the nadir FoV can be determined.

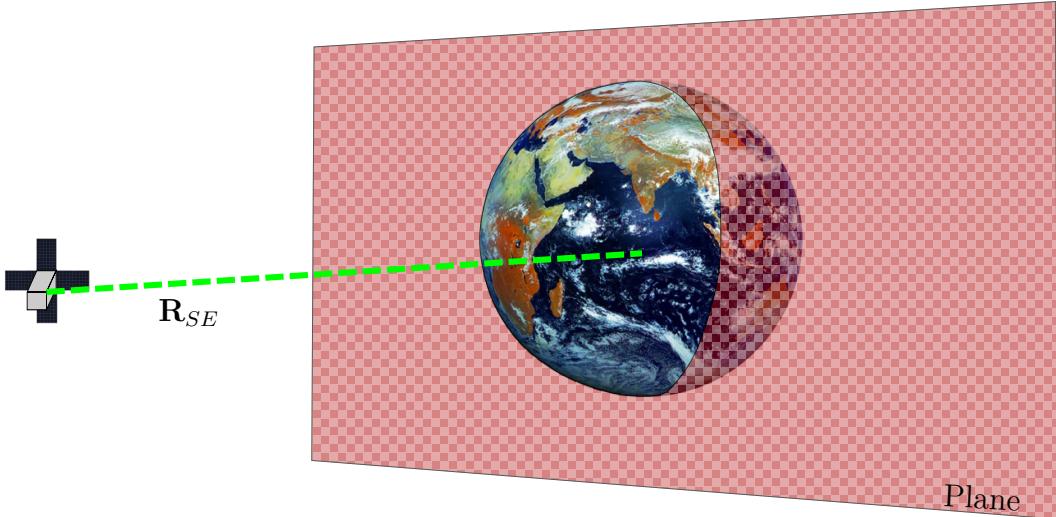


Figure 6.5: Plane perpendicular to \mathbf{R}_{SE} and at center of Earth

Firstly the edges of the moon and Earth are discretely determined due to the pixel width of the nadir sensor. The discrete points are therefore based on a fixed number of points for the Earth, N , 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 to $N \left(\frac{R_{moon}}{R_{Earth}} \right)$. The projected Earth and moon as discrete points unto the 3D plane is shown in Figure 6.6. The discrete edges of the Earth and moon that is within the FoV will be used for the algorithm to calculate the centre of the earth as discussed in Section 6.2.2. The discrete points from the earth used must satisfy the following conditions:

1. Distance between point and centre of nadir Sensor FoV must be smaller than R_{FoV} .
2. Distance between point and centre of 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 centre of Earth must be smaller than R_{Earth} for the moon to overlap the horizon.
2. Distance between point and centre of nadir Sensor FoV must be smaller than R_{FoV} .
3. Distance between point and centre of Earth must be larger than R_{Earth} .

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

TODO: Fix green arrow and text of Rmoon on Figure

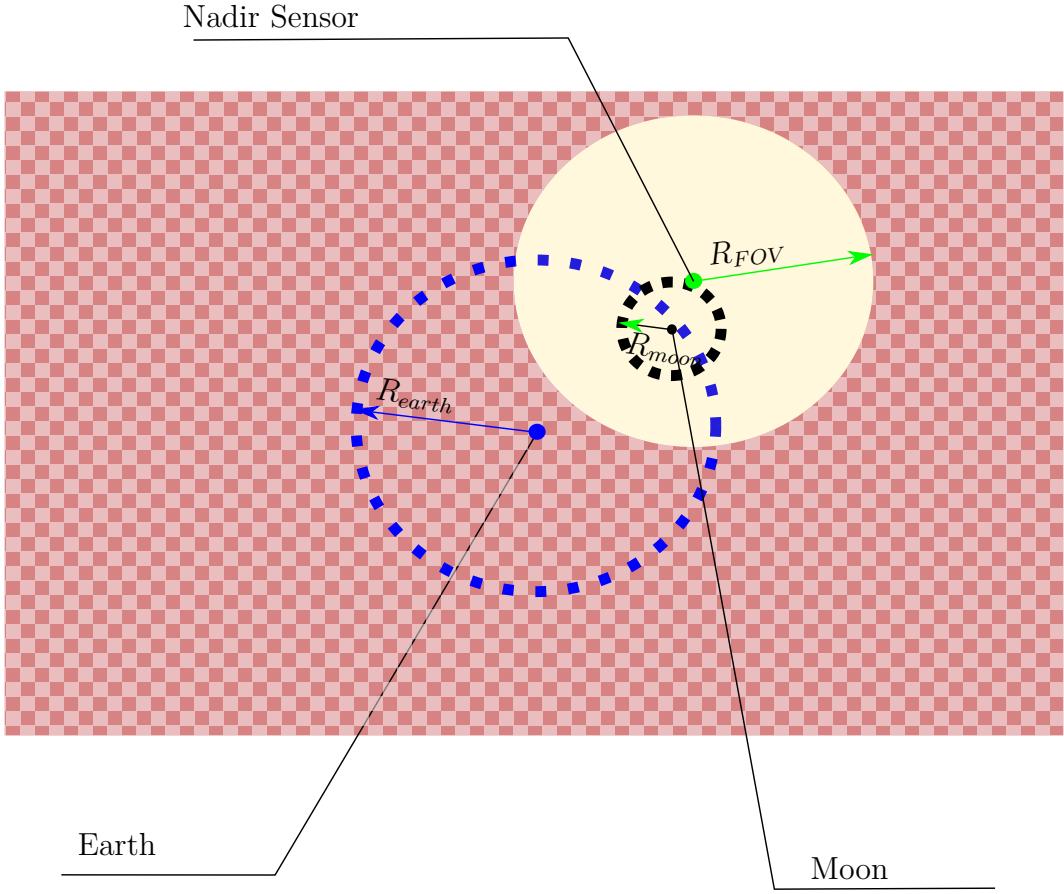


Figure 6.6: Projection of moon and Earth on plane

6.2.2. 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 nadir sensor's FoV. This will not be implemented in our case, since we can 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. Consequently, the IR horizon sensor must 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. Therefore, it can 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 its lowest temperature being warmer than that of space.

With this assumption the circular fit algorithm as shown in Figure 6.7 can now be used to determine the centre of the Earth on the plane [26]. For this calculation the 3D plane is transformed to a 2D plane and all the coordinates are 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 (x_c, y_c) and use it to transform the point to the 3D plane and thereafter calculate

the \mathbf{R}_{SE} .

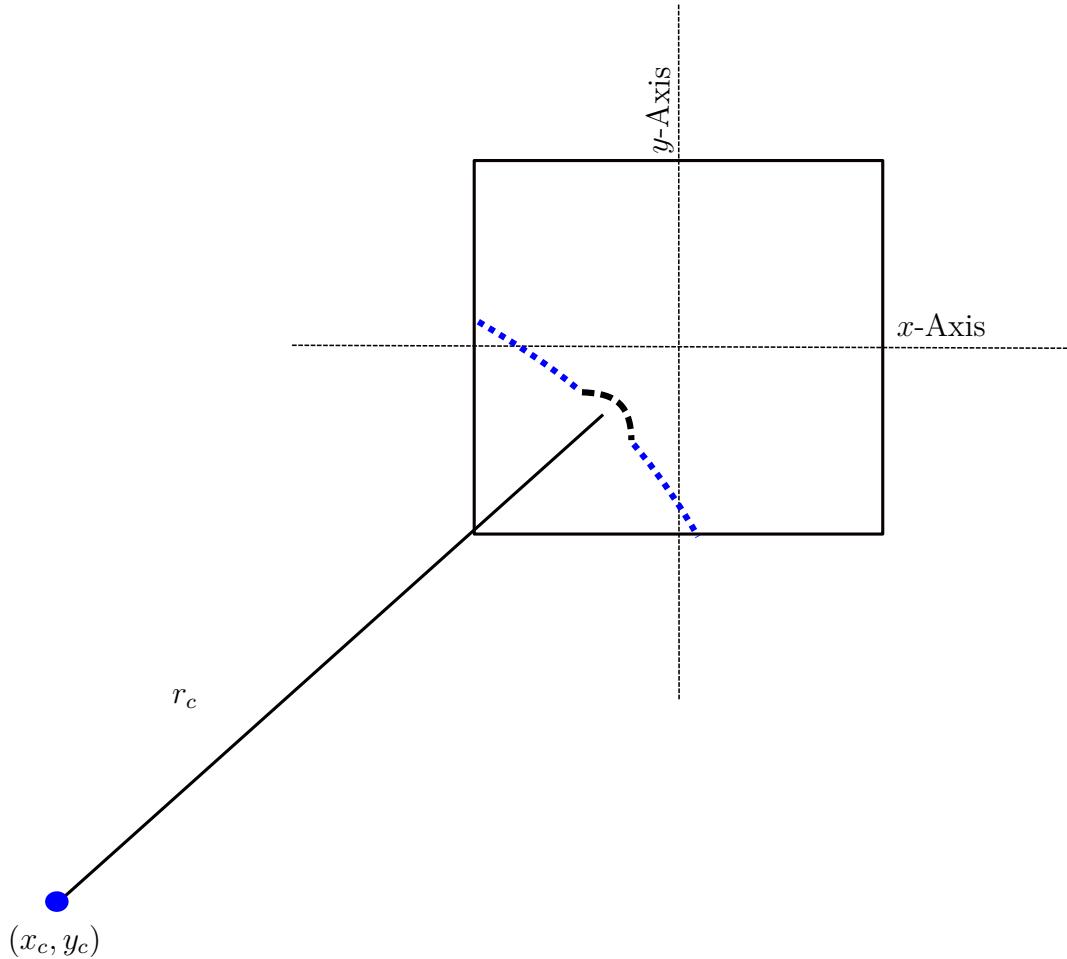


Figure 6.7: Circular Fit Algorithm

Firstly the curvature of a circle is described as

$$ax + by + c = x^2 + y^2, \quad (6.10)$$

where

$$\begin{aligned} a &= 2x_c \\ b &= 2y_c \\ c &= r_c^2 - \sqrt{x_c^2 + y_c^2}. \end{aligned} \quad (6.11)$$

Therefore using all the coordinates of discrete edges, (x_n, y_n) , within the nadir sensor FoV, a , b and c can be calculated as

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x_0^2 + y_0^2 \\ x_1^2 + y_1^2 \\ \vdots \\ x_n^2 + y_n^2 \end{bmatrix}. \quad (6.12)$$

It is thus evident that when the moon overlaps the horizon of the Earth from the nadir sensor's perspective the centre of the Earth will be incorrectly calculated and this anomaly must be dealt with. A similar anomaly where the sun is in the FoV of the nadir sensor will not provide a measurement, since the sun will saturate the Infra-red nadir sensor [26]. The anomaly will therefore not be modelled since it will only provide a sensor vector of 0 and will be ignored by default.

6.2.3. Influence of Anomaly on Estimation

To determine the effect of the discrete moon edges on the circular fit algorithm, the measured Earth vector with and without the moon on horizon anomaly is shown in Figure 6.8. It is clear from Figure 6.8 that the anomaly has no visible effect on the Earth vector. It is also evident in Figure 6.9 that the estimation metric is also not influenced negatively by this anomaly. This anomaly is therefore not included in the FDIR development, since there is no evident difference due to the anomaly. It will also have a negative influence on the detection algorithms, since the data from this anomaly will be similar to normal data and this will decrease the accuracy of the detection.

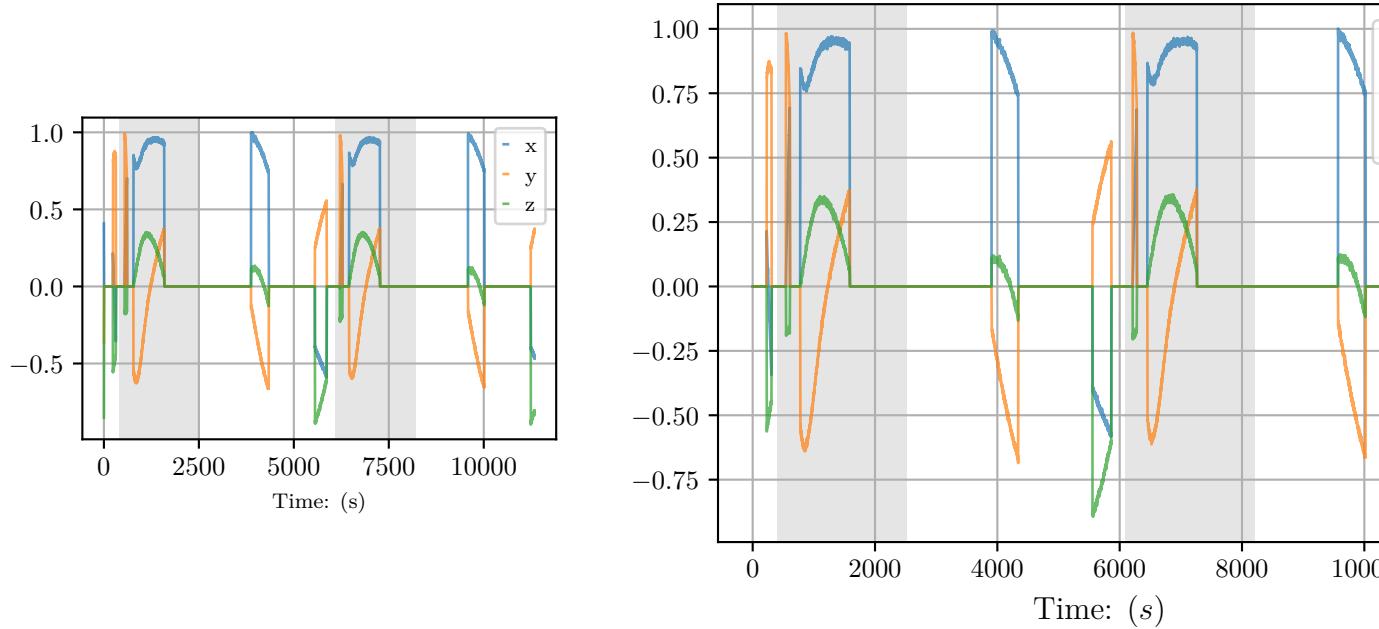


Figure 6.8: Comparison of Earth Vector with and without moon on the horizon

6.3. Magnetic Moment Disturbance

Magnetic moments produced by a coil in solar panels on a CubeSat can create a disturbance torque an influence the magnetometer measurements due to the induced magnetic field in

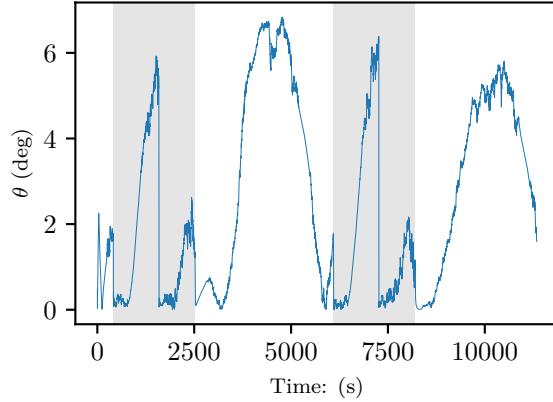


Figure 6.9: Earth Values

the coil of the solar panel [33, 51]. According to [33] 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, m , is shown in Figure 6.10. The inner area of the coil, A , is assumed to be the same as the surface area of the solar panel.

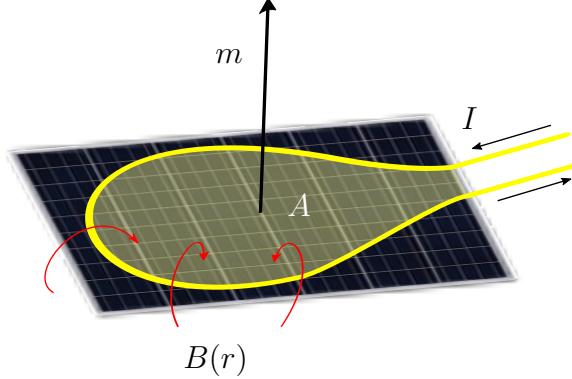


Figure 6.10: Dipole Moment from circular loop in solar panel

The dipole moment is calculated as

$$\mathbf{m} = AI. \quad (6.13)$$

The current is generated by the solar panel and thus depends on the incoming sun vector as well as the area on the solar panel illuminated by the sun. A shadow of the satellite body can cover areas of the solar panels as demonstrated in Figure 6.11. This decreases the current in these solar panels and also the induced dipole moment from these solar panels.

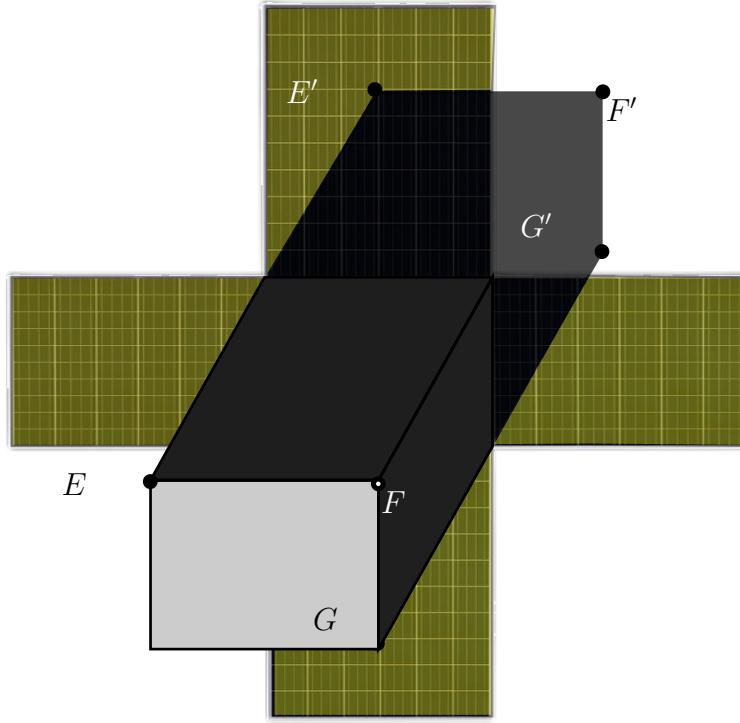


Figure 6.11: Shadow created by CubeSat body on Solar Panels

The current, I , can therefore be calculated as

$$I = I_{max} \frac{A_{total}}{A_{illuminated}} \cos(\theta), \quad (6.14)$$

where θ is the angle between the normal vector to the solar panel and the incoming sun vector and I_{max} depends on the solar panel. The dipole moment in term produces a disturbance torque on the CubeSat. With the resulting torque, from the dipole moment, expressed as

$$\mathbf{N}_{dm} = \mathbf{m} \times \mathbf{B}, \quad (6.15)$$

where \mathbf{B} is the magnetic 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 6.12. It is evident that \mathbf{N}_{dm} is 0 during eclipse, since there is no current from the solar panels and therefore no induced dipole moment.

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

$$\mathbf{B}_{r_m} = \frac{\mu_0}{4\pi} \frac{3\mathbf{r}_m(\mathbf{r}_m \cdot \mathbf{m}) - \mathbf{m}}{\|\mathbf{r}_m\|^3}, \quad (6.16)$$

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 position of the magnetometer. The vector, \mathbf{r}_m , will therefore

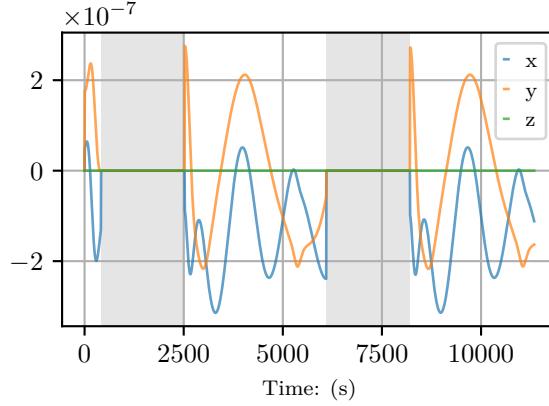


Figure 6.12: Solar Panel Disturbance Torques

change depending on each solar panel and is calculate as \mathbf{B}_r is then a summation of each solar panel's resultant magnetic field, \mathbf{B}_{r_m} .

6.3.1. Influence of 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 magnetic field, \mathbf{B} , and the magnetic field produced by the coils in the solar panels, \mathbf{B}_r . The resulting magnetometer measurement with and without the induced dipole moment is shown in Figure 6.13.

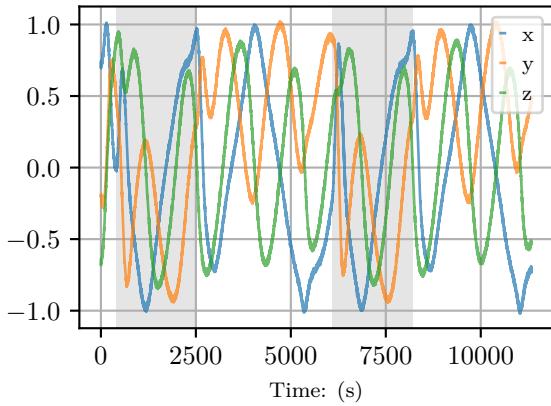


Figure 6.13: Comparison of Magnetic field Vector with and without solar Panel magnetic field

TODO: Plot the magnetic field due to the solar panels and not the unit vector.

From Figure 6.14 is is evident that this anomaly has a significant effect on the estimation, but not as large as the sun reflection on the sun sensor. However, it is not yet clear whether

the estimation error increases due to the magnetometer anomaly or the disturbance torque, \mathbf{N}_{dm} . This however will be evident when the anomaly recovery results are discussed in Chapter 8.

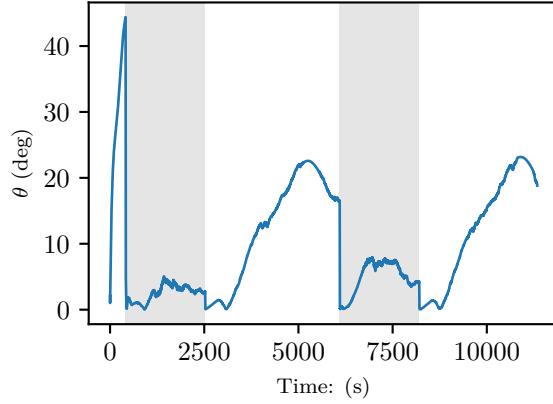


Figure 6.14: Estimation Metric with induced dipole moment

6.4. Reaction wheels

Failures of actuators will 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 at the control input it had before the sudden failure. 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. Therefore, this anomaly is included even though it is not a sensor anomaly, because this anomaly often occurs and a lot of research is done on reaction wheel failure. The recovery of the control torque, \mathbf{N}_w , 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, \mathbf{N}'_w . The resulting estimation metric for this anomaly is shown in Figure 6.15 and it is evident that this anomaly has a large negative effect on the EKF.

6.5. Summary

It is evident that the reflection of solar panels on sun sensor, reaction wheel failure and the magnetic moment disturbance have considerable increases in the estimation error. Therefore, these anomalies must be recovered from to ensure autonomous fault tolerant control. The moon in the FoV of the nadir sensor and on the Earth horizon has a negligible effect on the estimation accuracy and therefore is not included in the set of anomalies that

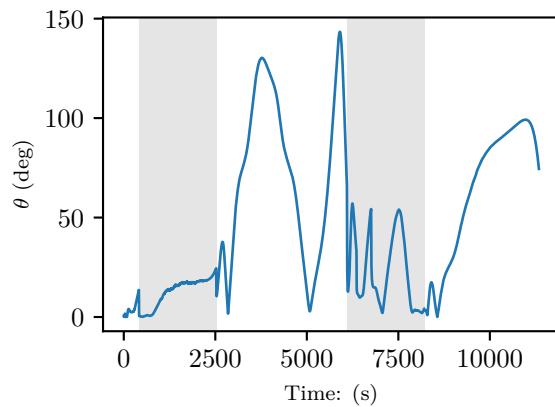


Figure 6.15: Estimation Metric with failure of Reaction Wheels

requires fault tolerant control. The results from recovery of these anomalies is discussed in Chapter 8.

Chapter 7

Feature Extraction

Feature extraction is a method of enhancing the performance of classification algorithms. Through different algorithms data sets with a large feature space is reduced to a smaller feature space. Classification algorithms has fewer features to assess, but the information gain of each feature is significantly larger, depending on the application. For anomaly detection, feature extraction focuses on producing features that specifically divides anomalies from normal data. Feature extraction can also be implemented for the isolation of different failures, this however is not within the scope of this thesis and only feature extraction for anomalies is discussed.

7.1. Local Outlier Factor

Most algorithms for anomaly detection are based on a metric which accounts for the entire dataset [52]. However, many anomalies are identifiable in relation to the local neighbourhood of data points and not the overall dataset. Therefore, [52] developed the local outlier factor (LOF) algorithm that provides a measure of a data point's "outlierness" within a subset of data points. This implies that a data point is not classified as an anomaly or not, but a local outlier factor is calculated to determine how much a data point is distantiated from it's k -nearest neighbours. This is clearly demonstrated in Figure 7.1 where the data points which are clustered together have smaller outlier scores than data points which are removed from the highly dense areas. The radius of the red circle around each data point is equivalent to the outlier score, **LOF**.

To calculate the **LOF**, the k -distance, d_k , must be calculated and also the local reachability density **LRD**. The d_k , is the k^{th} ranked $d(p_o, p_i)$. Where $d(p_o, p_i)$ is the distance between data point p_o and any data point p_i , with $i \in N$, where N is the number of data points within the dataset with a minimum value of N_{min} . The minimum data points for a data point, p_o , is a function of the minimum value of points and denoted as $N_{min}(d_o)$. To reduce fluctuations in the $d(p_o, p_i)$ the distance between p_o and p_i is replaced with

$$\max(d(p_o, p_i), d_k), \quad (7.1)$$

and will henceforth be referred to as the reachability distance, **d**, [52]. The **LRD** of a

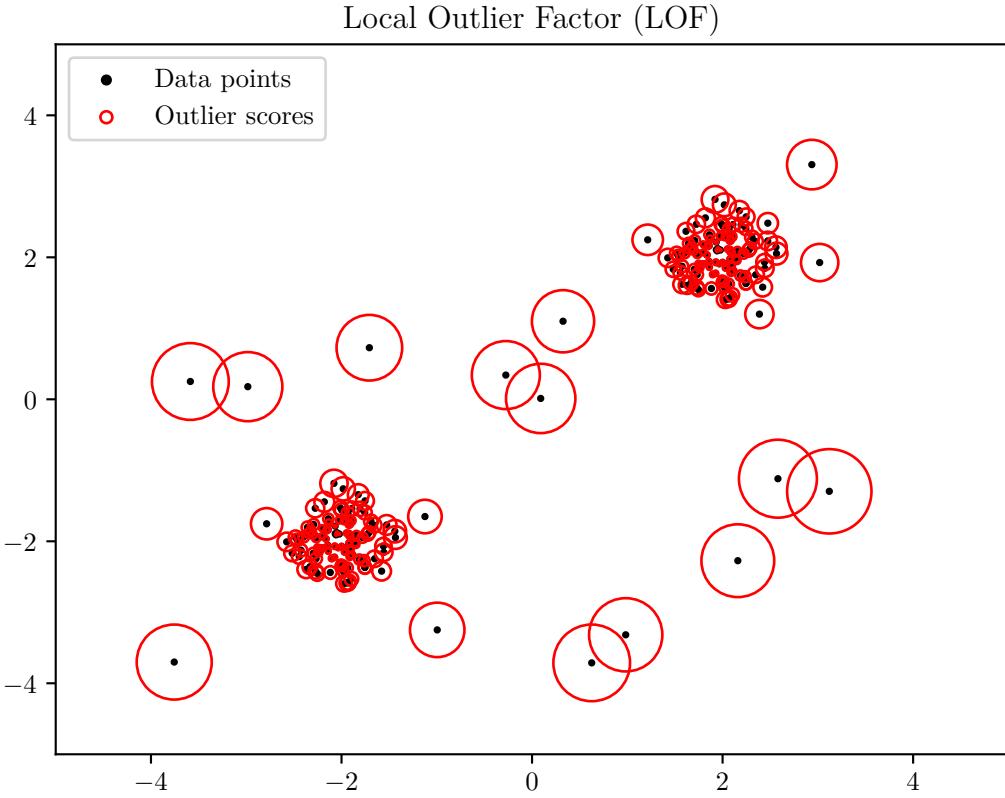


Figure 7.1: Local outlier factor of random data set to demonstrate the outlier score produced by the local outlier factor algorithm.

data point, p_o , is calculated as

$$\text{LRD}_{N_{min}}(p_o) = 1 / \left(\frac{\sum_{p_i \in N_{min}(p_o)} \mathbf{d}_{N_{min}}(p_o, p_i)}{|N_{min}(p_o)|} \right), \quad (7.2)$$

and denotes "the inverse of the average reachability distance based on the N_{min} -nearest neighbours of the p_o " [52]. Eq 7.2 enables the calculation for the **LOF** of point p_o given as

$$\text{LOF}_{N_{min}}(p_o) = \frac{\sum_{p_i \in N_{min}(p_o)} \frac{\text{LRD}_{N_{min}}(p_i)}{\text{LRD}_{N_{min}}(p_o)}}{|N_{min}(p_o)|}. \quad (7.3)$$

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

This method is aimed at producing a measure of the "outlierness" of a data point

within a local neighbourhood and not for all the data points. This method will thus be implemented for the satellite anomaly detection, since it will detect anomalies within the two neighbourhoods produced by the eclipse during orbit. This method will also be able to detect anomalies from measurements of the nadir sensor, sun sensor and magnetometers that drastically change from the training data set.

7.2. Linear Regression Model

The proposed method by [24] uses Dynamic Mode Decomposition (DMD), which was initially developed by [53] and further expanded to include control by [54], to provide an estimation of a sensor vector based on the previous measurement of the sensor as well as the measurements of the other sensors in the system. DMD was first developed in the fluids community and constructed a matrix \mathbf{A} to relate the state vector x with the following time step of the state vector, x_{k+1} . The state vector, in our case, will be the measurement vector of the specific sensor that we want to monitor.

$$\mathbf{x}_{k+1} = \mathbf{Ax}_k \quad (7.4)$$

Where \mathbf{x}_k and \mathbf{x}_{k+1} during a specified number of time steps, will be denoted as \mathbf{X} and \mathbf{X}' respectively.

The method of DMD, however, is useful for high order systems where the calculation of \mathbf{A} is computationally intensive. This is not the case for our system, and using DMD is not justifiable and consequently, a linear regression model is implemented. Therefore with the pseudo-inverse of \mathbf{X} , denoted as \mathbf{X}^\dagger , we calculate \mathbf{A} as

$$\mathbf{A} = \mathbf{XX}^\dagger \quad (7.5)$$

This necessitates data of the state vector over time. The article by [24] however includes \mathbf{B} to relate the vector measurements of the other sensors to adjust the predicted state, X_{k+1} of the monitored sensor.

$$\mathbf{X}_{k+1} = \mathbf{AX}_k + \mathbf{BY}_k \quad (7.6)$$

Where \mathbf{Y}_k is the other sensor measurements, this is adjusted for our use case, where \mathbf{Y}_k is the control torques for the magnetorquers and reaction wheels, while \mathbf{X}_k is all of the sensor measurements. Consequently, the model of Eq 7.6 denotes the prediction of the sensor measurements at time step $k + 1$ based on the current sensor measurements and control inputs. Thereafter, as implemented by [24] the model is adjusted with a Kalman

Filter. From \mathbf{A} and \mathbf{B} the Kalman filter can be implemented to predict \mathbf{X}_{k+1}

$$\hat{\mathbf{X}}_{k+1} = \mathbf{A}\hat{\mathbf{X}}_k + \mathbf{B}\mathbf{Y}_k + K(\mathbf{X}_k - \hat{\mathbf{X}}_k) \quad (7.7)$$

where $K = 0.001$. After the calculation of $\hat{\mathbf{X}}_{k+1}$ [24] proposes a moving average of the innovation covariance

$$\mathbf{V}_k = \frac{1}{N} \sum_{i=k-N}^k (\mathbf{X}_i - \hat{\mathbf{X}}_i)(\mathbf{X}_i - \hat{\mathbf{X}}_i)^T \quad (7.8)$$

where N is the number of timesteps to account for. The moving average is used as an additional input parameter for the classification of anomalies based on \mathbf{X} .

TODO: Plot sum of moving average for normal and anomaly

7.3. Summary

The feature extraction methods discussed in this chapter focuses on extracting features that provide information gain on whether a data point is an anomaly or not. For the LOF algorithm this is done by providing a value for the "outlierness" of that data point relative to its nearest neighbourhood [52]. The linear regression model is an adaptation of work by [24] to provides 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 of the estimated sensor measurements and the actual sensor measurements.

Chapter 8

Recovery

Three different methods of recovery are compared. These methods are all focused on ensuring that the sun reflection does not change the reliability and stability of the EKF.

- The *EKF-ignore* method uses the detected sensor that has failed and ignores the sensor measurement from the EKF measurement update. This method is based on the assumption that the EKF estimation is correct up until the moment when the sensor failure is detected. This, however, will highly depend on the accuracy of the anomaly detection method. A detection method with low accuracy will create instability of the EKF since many anomalous measurements will be included in the measurement update of the EKF.
- The *EKF-reset* method uses a buffer of $v_{meas,k}$, $v_{model,k}$ and \hat{x}_k^+ and other parameters that are used to update the EKF. If a sensor failure is detected, the sensor is excluded from the EKF, and the EKF is updated with the sensor data in the buffer, excluding the sensor that has failed. The EKF is, therefore, *reset* and updated from timestep t_{k-N} to t_k , where N is the size of the number of timesteps in the buffer. N , however, must be optimized based on the computational time used to reset the EKF but still ensure convergence of the EKF. If the sensor that was detected to have anomalous behavior changes back to normal again, the EKF will be reset once again, and the sensor will only be included in the measurement update of t_k since it was anomalous for timesteps before t_k .
- A backtracking method can be combined with the ignore method, *EKF-combination*. For example, where the backtracking method is implemented only after a specified number of sun reflections are predicted.
- Another method implemented and tested continually uses the two sensors' measurements, *EKF-top2*, that have the smallest mean squared error between the estimated SBC vector and the actual measured SBC vector. There are, however, setbacks to this method. Firstly, it requires the modeling of the ORC vector and requires the position of the satellite in orbit. Secondly, this method will not work with small drifts in a sensor measurement since the estimator will latch unto the drift in the sensor. The method will only detect sudden changes in the sensor and isolate the

sudden change even if it remains stable after the sudden change. This method will only be compared to the other methods as part of the analysis since the method is inherently different. The method is also used to aid the other methods during a period after an anomaly is detected.

8.1. Analysis

TODO: Provide results and comparison of each recovery method as well as coupling the method with EKF-top2 and different percentage detection and isolation accuracy. Do this for each individual anomaly

8.1.1. Sun Reflection

8.1.2. Magnetic Moment Disturbance

8.1.3. Reaction Wheel Failure

Chapter 9

Detection and Isolation

Detecting the anomaly is a binary classification. The method implemented must therefore be able to distinguish between normal data and anomalies. There are various methods that can be implemented for this use case, however only three supervised learning methodologies and two unsupervised learning methods are implemented. The major difference between the supervised learning and the unsupervised learning methods for anomaly detection is that for supervised learning the number of anomalous samples and normal samples should be more or less the same, while with unsupervised learning the anomalous samples should be sparse. For simplicity a dataset of n samples is given as

$$\mathbf{D} = (\mathbf{f}_1, y_1), \dots, (\mathbf{f}_n, y_n), \quad (9.1)$$

where $\mathbf{f} \in \mathbf{R}^m$.

\mathbf{f} is the point vector within the vector space, \mathbf{R} , with m dimensions, while y denotes the class of the feature, which in binary classification is either 0 or 1.

9.1. Supervised Learning

Supervised learning is a subgroup of machine learning which is based on training algorithms from labelled data to ensure accurate predictions.

9.1.1. Decision Trees

Decision Trees and Random Forests will be implemented to perform binary classification on regular data samples and anomalous data samples. Decision Trees [55] and Random Forests [56–58] are supervised learning algorithms that classify data based on threshold splitting. Data samples are split based on a threshold of a specific input parameter. For instance, binary classification can be performed on data samples from a satellite orbit to determine whether the satellite was in an eclipse or not. This would be done by determining whether the magnitude of the sun vector is smaller or greater than 0. Consequently, after two splits it can be determined which data samples are within an eclipse. The Decision Tree determines this split with the classification and regression tree (CART) algorithm.

However, to split the data for the anomalies, we need to decide which input parameter will be used to make the first split, the root node. The Gini index measures the probability of a data sample being wrongly classified at a given node. This can be calculated by

$$GI = 1 - \sum_{i=1}^n (P_i)^2. \quad (9.2)$$

The operator split that produces the lowest Gini index provides the purest split and will be used as the root node. For our use case, the CART algorithm will be used to optimize the Decision Tree, which also considers the most prominent information gained to construct the Decision Tree. Figure 9.1 is a graphical representation of the Decision Tree developed to classify anomalies. In Figure 9.1 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. The depth of a Decision Tree determines how many splits occur from the root node to the leaf node 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. However, the larger the depth, 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.

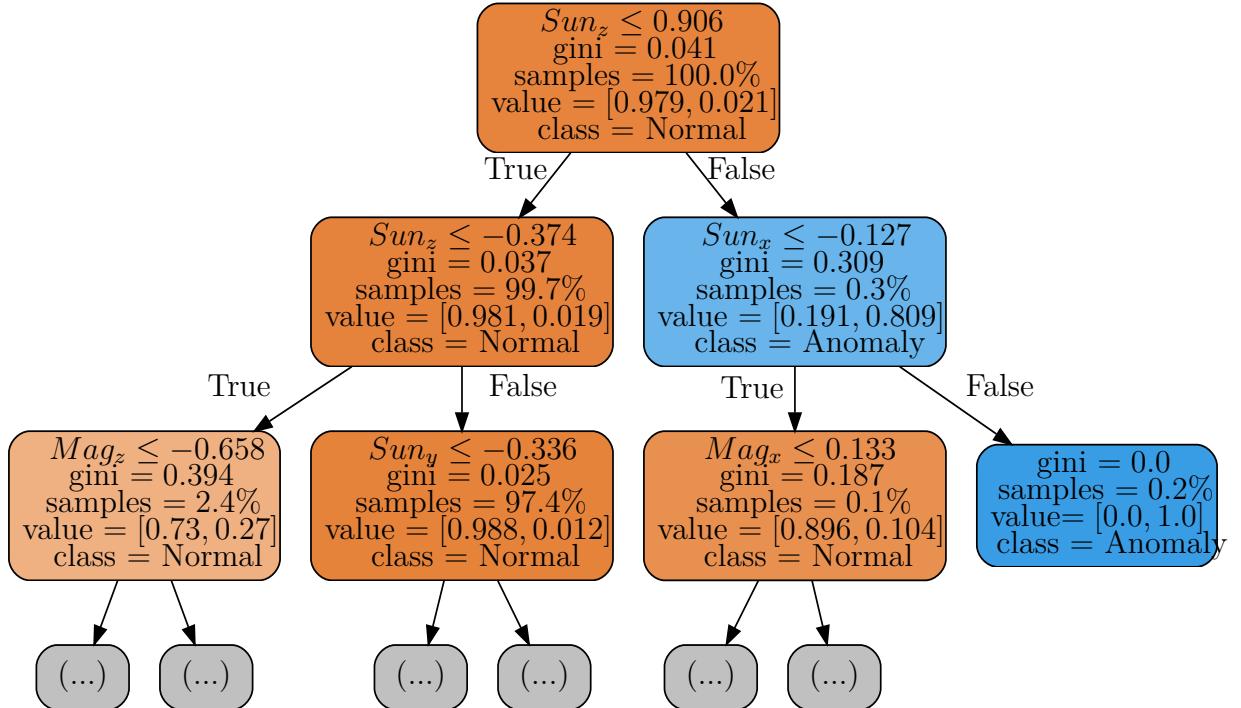


Figure 9.1: Decision Tree as binary classification of sun reflection. Orange nodes have more than 50% normal samples, while blue nodes have more than 50% anomalous samples. The darker the shade of orange or blue, the larger the percentage of the respective class.

It is evident in Figure 9.1 that the most notable splits at the beginning of the tree are the moving average from the feature extraction (DMD) as well as the sun sensor and

magnetometer measurements. This makes logical sense since the moving average provides the system with an extracted feature that correlates with the change in measurements. The sun reflection can also be detected on the sun sensor measurements, and changes in the control torques influences the magnetometer readings.

9.1.2. Random Forests

Random forests, (RF), is an extension of the decision tree. It generates various decision trees from the data set and provides a classification output, \mathbf{P} , for a new data sample, \mathbf{D}^* , by utilizing a majority vote from all the decision trees with an individual classification output of, \mathbf{p}_i . RF utilizes bootstrap aggregating, $\mathbf{ba}()$, to ensure that new random data sets, \mathbf{D}' with size of n' is created [56–58]. The individual decision trees of number k is then trained with Algorithm 9.3.

Algorithm 9.3: Training of Random Forest from Individual Decision Trees

```

1: for  $i := 1$  to  $k$  do
2:    $\mathbf{D}'_i = \mathbf{ba}(D)$ 
3:    $\mathbf{DT}_i = \mathbf{CART}(\mathbf{D}'_i)$ 
4: end for
```

The classification output, \mathbf{P} , is then provided with Algorithm 9.4.

Algorithm 9.4: Prediction of Random Forest from Individual Decision Trees

```

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

9.1.3. Support Vector Machines

A study conducted by [59] implemented Support vector machines (SVM) for sensor and actuator failure detection on small satellites systems. This example as well as many other fault detection research in other fields implement SVM for anomaly detection. SVM creates a hyperplane between the feature space to increase the margin between the data samples nearest to the hyperplane from all the classes within the data [60, 61]. The hyperplane can be defined as

$$\mathbf{w}^T(\mathbf{x}) + b = 0 \quad (9.3)$$

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

$$\mathbf{w}^T(\mathbf{x}_i) + b \geq 1, \quad \text{for all } 1 \leq i \leq n. \quad (9.4)$$

The hyperplane in it's most simplified form is linear and divides the data into two halves based as shown in Figure 9.2. The distance between the hyperplane and any point vector, \mathbf{f}_i , can be given as

$$d(\mathbf{x}_i) = \frac{\mathbf{w}^T(\mathbf{x}_i) + b}{\|\mathbf{w}\|^2} \quad (9.5)$$

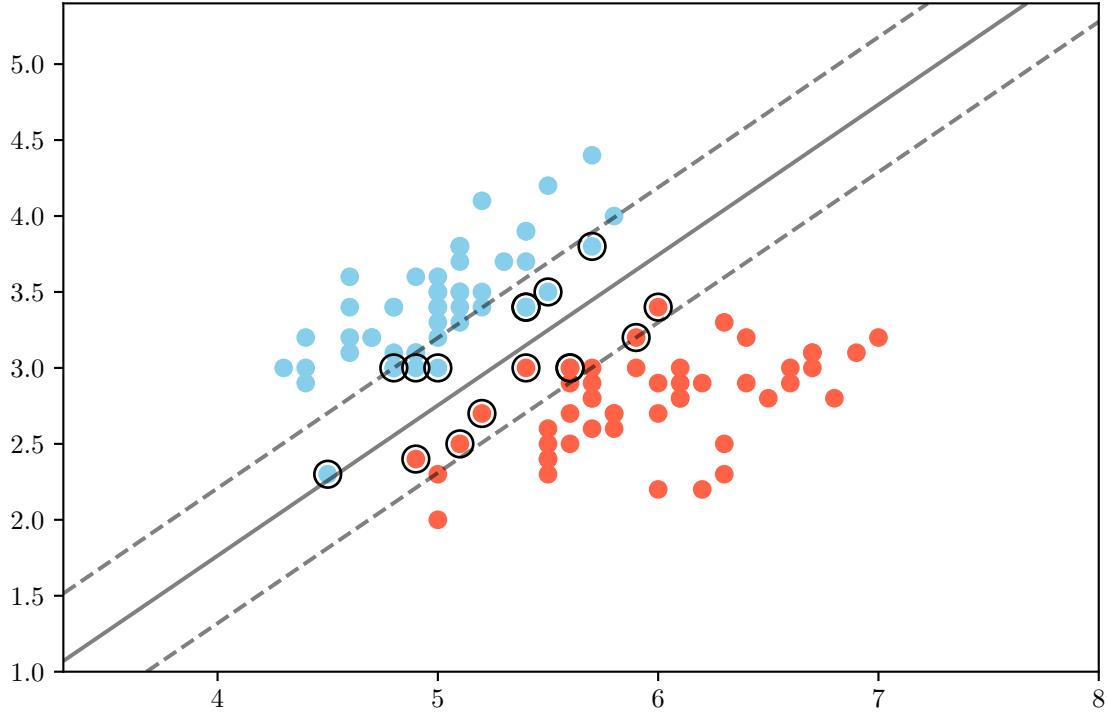


Figure 9.2: Support Vector Machine Example

For binary classification all point vectors, \mathbf{x}_n will be classified into a group . To increase the margin of between all point vectors, \mathbf{x}_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{x}_i , for all data samples n . This is however extended to a soft-margin optimization algorithm given as

$$\lambda \|\mathbf{w}\|^2 + \left[\frac{1}{n} \sum_{i=1}^n \max(0, 1 - \mathbf{w}^T(\mathbf{x}_i) + b) \right], \quad (9.6)$$

where λ increases and decreases the margin size ensuring that all training \mathbf{x}_i are on the correct side of the hyperplane. An example of an algorithm used to solve the optimisation of Eq 9.6 is the sub-gradient descent given as

$$f(\mathbf{w}, b) = \lambda \|\mathbf{w}\|^2 + \left[\frac{1}{n} \sum_{i=1}^n \max(0, 1 - \mathbf{w}^T(\mathbf{x}_i) + b) \right], \quad (9.7)$$

where f is a convex function and can be solved with an adaptation of classic gradient

descent methods. The difference is that the direction of gradient descent is that of the vector selected from the function's sub-gradient. With this implementation the hyperplane can divide two groups as shown given in Figure 9.2. The vector points within the margin size and with the additional black circle in Figure 9.2 are called the support vectors since these vector points are the minimum distance from the hyperplane and influence Eq 9.6.

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, \mathbf{m} , of $\mathbf{R}^{\mathbf{m}}$. For instance increasing m from 2 to 3 with

$$\mathbf{x}[2] = \sqrt{\mathbf{x}[0]^2 + \mathbf{x}[1]^2} \quad (9.8)$$

provides a higher dimension for the split of the data. This can be implemented with any equation for increasing \mathbf{m} . An example of this can be seen by increasing m from 2 in Figure 9.2 to 3 in Figure 9.3 by adding another feature from the data. A 3D hyperplane can now divide the data into two separate groups.

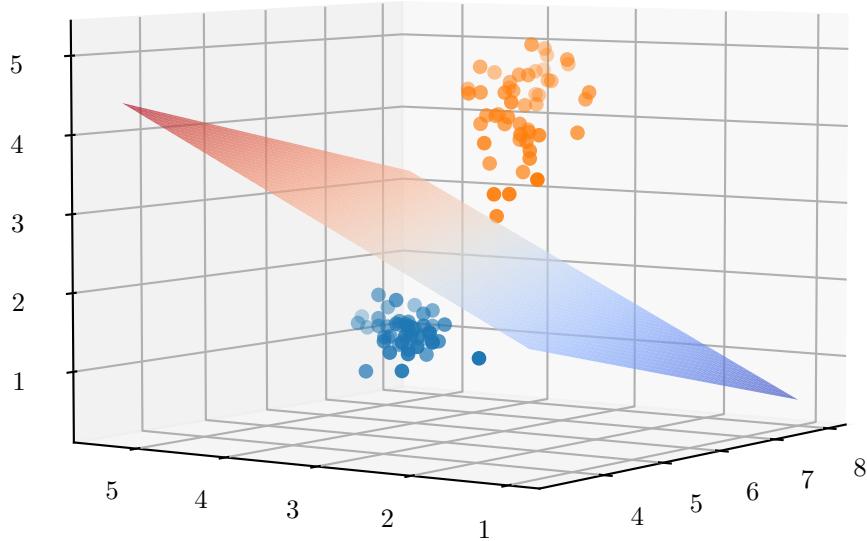


Figure 9.3: Support Vector Machine Higher Dimension Example

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. A linear kernel is given as

$$\mathbf{K}(x_i, x_j) = x_i^T x_j + \mathbf{C}, \quad (9.9)$$

whereas a polynomial kernel is given as

$$\mathbf{K}(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \quad \gamma > 0. \quad (9.10)$$

This is provided with the example of a polynomial division of the data shown in Figure 9.2 with a polynomial fit of the 3rd degree as shown in Figure 9.4. The kernel will depend on the nature of the data and the relationship between the different features. This is therefore an iterative process to decide the kernel unless available opensource algorithms are used to determine which kernel structure is the best suited for the current dataset.

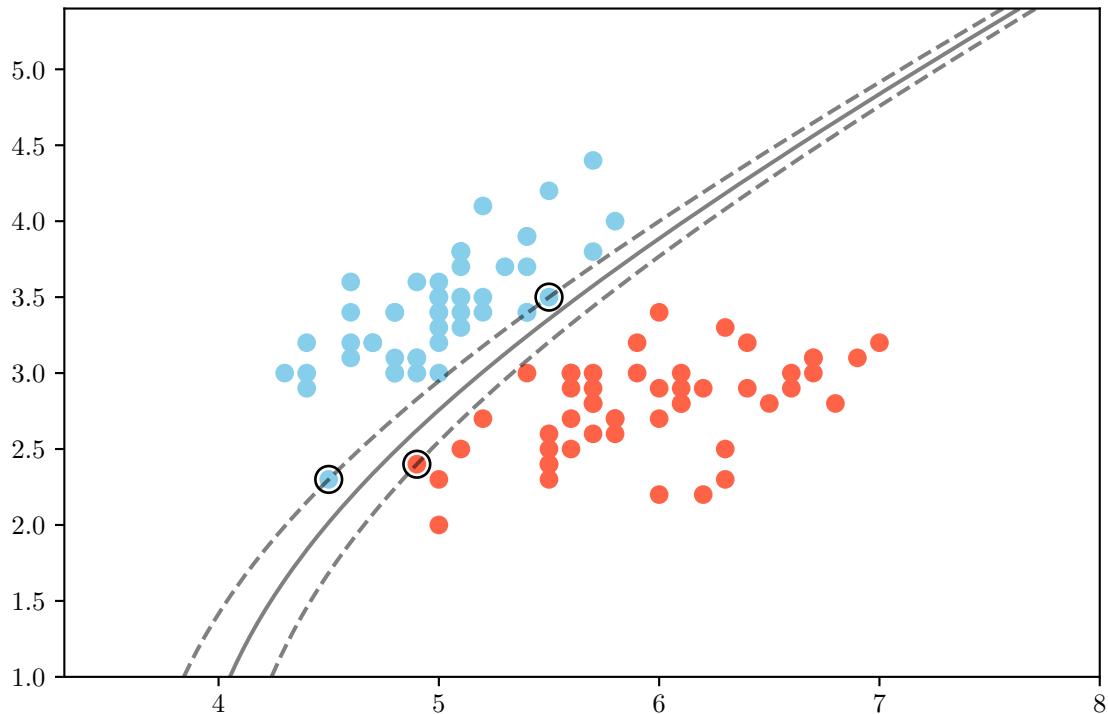


Figure 9.4: Support Vector Machine Polynomial Example

9.2. Unsupervised Learning

Supervised learning is a subgroup of machine learning which is based on unlabelled data. Local Outlier factor as discussed in Section 7.1 can also be implemented as an unsupervised learning method.

9.2.1. Isolation Forests

This unsupervised learning method is based on the principle of isolating data points by slicing the data with random conditions [62]. The data is randomly split into specified sample sizes with a randomly selected dimension and a randomly selected cut-off value. For each sample size the data must be split until each data point within the sample is isolated from all other data points. Training of a single tree is completed when all the data points are isolated and this training must be repeated for all the data samples, however many are predefined.

The distance measured from the first split the *tree top* to the isolated data point is used to determine whether a data point is anomalous or not [63]. The logical reasoning for support of this algorithm is that data points which are non-anomalous will be more closely related and hence have more splits to separate the data points until isolation is achieved. Therefore, the distance from the tree top for non-anomalous data points will be longer than anomalous data points which will have a shorter distance from the tree top. Therefore non-anomalous data points are closer to the *root*.

Figure 9.5 demonstrates the splitting of the data points until isolated. Each split or *branch* only splits the data into two groups. After training multiple trees, a single data point is "sent through the forest" and the distance from the tree top for each tree is calculated and the average of all the trees are used to calculate the average distance for the data point. Using a threshold for the distance, the data point is classified as anomalous or not.

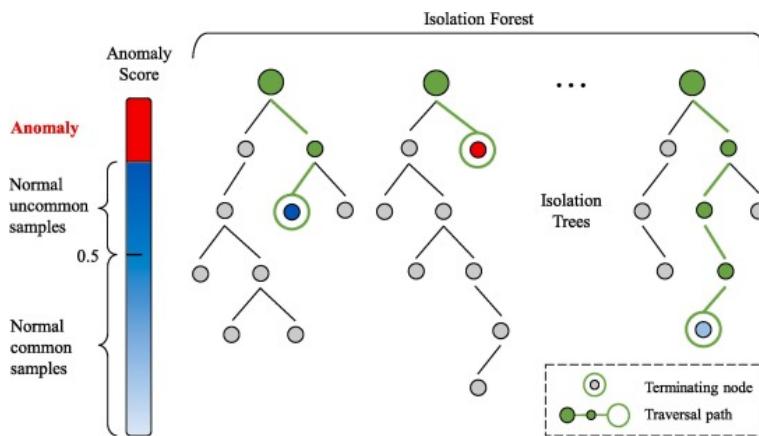


Figure 9.5: Isolation Forests [64]

An example of a trained isolation forest is provided in Figure 9.6. The samples in white are the training samples of which all are considered normal data. Therefore the tree is constructed and the branches are created until all the data samples in the training data set are divided into individual nodes. Thereafter, the new data set is classified and is evident that data samples far removed from the normal training samples are isolated within the anomaly threshold and are therefore considered anomalous (depicted as red).

TODO: Ensure that isolation forest is done trained with no anomaly samples.

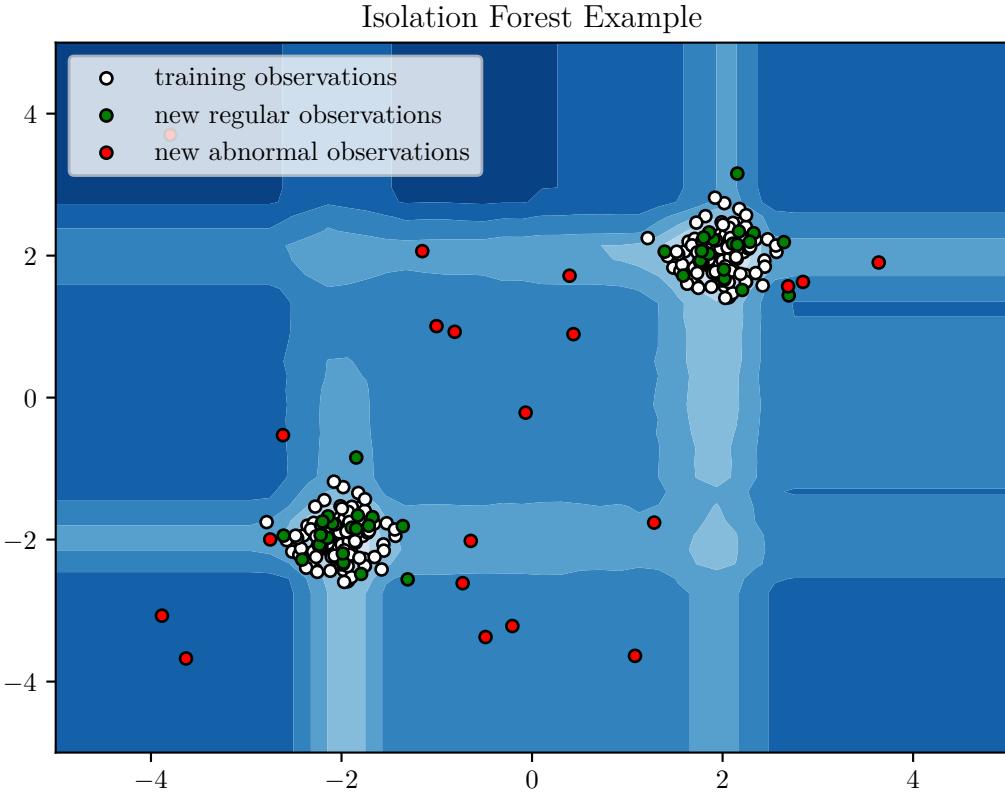


Figure 9.6: Isolation Forest

The anomaly score is calculated as

$$s(x, n) = 2^{-E(h(x))/c(n)}, \quad (9.11)$$

where $E(h(x))$ is the average value of the distance measured from the tree top for a single data point in all the trees [63] and n is the size of a data sample used to train a single tree. For the distance to be normalized, $c(n)$ — the mean distance from the tree top in an unsuccessful search in a *Binary Search Tree* (BST) — is used and is calculated as

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}. \quad (9.12)$$

$H(i)$ in Eq 9.12 is the harmonic number and is estimated with Euler's constant as

$$H(i) \approx \ln(i) + 0.5772156649. \quad (9.13)$$

Isolation Forests, however have multiple issues, since it splits data in rectangles as seen in Figure 9.6 and Figure 9.7a. This is due to the slicing algorithm selecting a feature, x

and a cut-off value, v . Consequently, the data is either split vertically or horizontally — if seen as a two dimensional dataset. This split method is unable to categorise complex data structures. These issues however are addressed by [63] and led to the *Extended Isolation Forest* algorithm.

The extended isolation forest algorithm generalises the isolation forest algorithm by applying a slope to each slice. Data points are therefore divided into two groups depending on the "side" of the plane or slice as seen in Figure 9.7b.

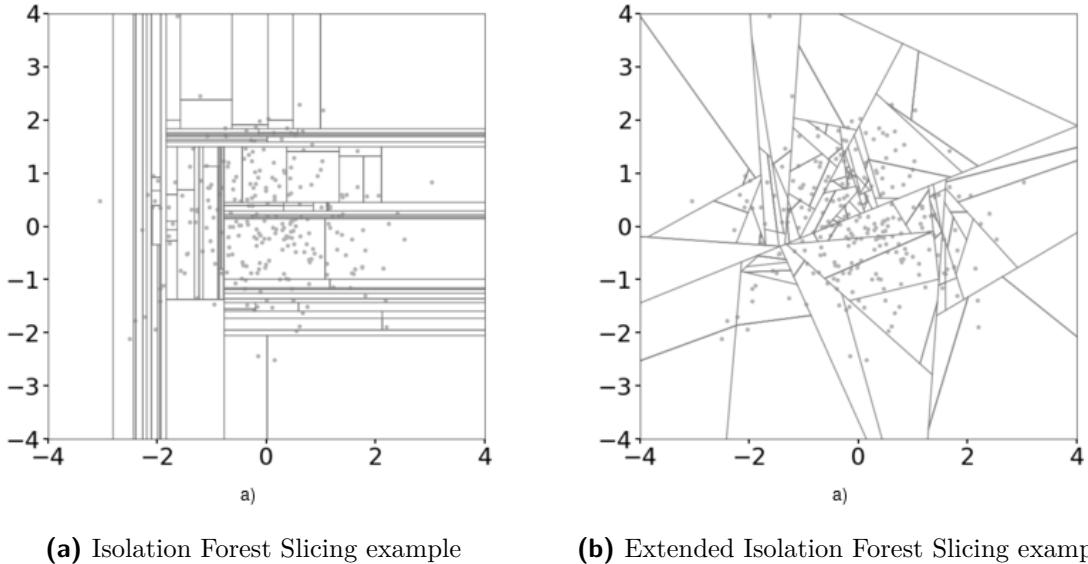


Figure 9.7: The slicing of Isolation Forest vs Extended Isolation Forest

It is evident that applying an angle of 0° to all the slices the general algorithm of the extended isolation forest produces the standard isolation forest algorithm where planes or slices are perpendicular to the axis of the randomly selected feature, x .

9.3. Analysis

TODO: Provide analysis of methods with perfect isolation and recovery ignore.

9.4. Summary

For the detection of anomalies different binary classification methods are discussed. This includes supervised learning methods such as decision trees, random forests and support vector machines as well as unsupervised learning methods such as isolation forests and local outlier factor which is only discussed in Section 7.1. Both supervised and unsupervised learning methods for detection only provide a binary split between anomalous and normal data points, however supervised learning methods can also be implemented for isolation, multi-class classification.

Chapter 10

Isolation

TODO: Analysis of each method with perfect detection

10.1. Decision Trees

10.2. Random Forests

10.3. Support Vector Machines

10.4. Summary

Chapter 11

Results

11.1. Perfect Extended Kalman Filter

For all these results the best recovery method will be demonstrated

11.2. Unsupervised Detection and Supervised Isolation

Use the best unsupervised learning algorithm from chapter Detection and the best supervised learning algorithm from chapter Isolation

11.3. Supervised Detection and Isolation

Use the best unsupervised learning algorithm from chapter Detection and the best supervised learning algorithm from chapter Isolation

11.4. Summary

Chapter 12

Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

12.1. Project/thesis/dissertation summary

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

12.2. Appraisal of project/thesis/dissertation contributions

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

12.3. Suggestions for future work

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

12.4. What the student has learnt during this project

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Bibliography

- [1] M. Tafazoli, “A study of on-orbit spacecraft failures,” *Acta Astronautica*, vol. 64, no. 2-3, pp. 195–205, 2009.
- [2] H. E. Koskinen and E. K. Kilpua, “Radiation belts and their environment,” in *Physics of Earth’s Radiation Belts*. Springer, 2022, pp. 1–25.
- [3] S. A. Jacklin, “Small-satellite mission failure rates,” 2019.
- [4] M. Swartwout, “Cubesat database,” URL: <https://sites.google.com/a/slue.edu/swartwout/home/cubesat-database> (visitado 12-06-2022), 2015.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] E. Sobhani-Tehrani and K. Khorasani, *Fault diagnosis of nonlinear systems using a hybrid approach*. Springer Science & Business Media, 2009, vol. 383.
- [9] 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.
- [10] O. Ciftcioglu and E. Turkcan, “Sensor failure detection in dynamical systems by kalman filtering methodology,” Netherlands Energy Research Foundation, Tech. Rep., 1991.
- [11] 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.

- [12] 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.
- [13] 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.
- [14] 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.
- [15] A. Wander and R. Förstner, *Innovative fault detection, isolation and recovery strategies on-board spacecraft: state of the art and research challenges*. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, 2013.
- [16] S. K. Ibrahim, A. Ahmed, M. A. E. Zeidan, and I. E. Ziedan, “Machine learning methods for spacecraft telemetry mining,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 4, pp. 1816–1827, 2018.
- [17] ———, “Machine learning techniques for satellite fault diagnosis,” *Ain Shams Engineering Journal*, vol. 11, no. 1, pp. 45–56, 2020.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] S. S. Nasrolahi and F. Abdollahi, “Sensor fault detection and recovery in satellite attitude control,” *Acta Astronautica*, vol. 145, pp. 275–283, 2018.
- [22] 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.
- [23] 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.

- [24] 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.
- [25] 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.
- [26] J. H. Wessels, “Infrared horizon sensor for cubesat implementation,” Ph.D. dissertation, Stellenbosch: Stellenbosch University, 2018.
- [27] *Sun sensor*, Sputnix, May 2020, v13.
- [28] *AAC SpaceQuest MAG-3*, AAC Clyde Space, 2022, three Axis Satellite Magnetometer.
- [29] *Microsatellite Reaction Wheels (RW3-0.06)*, Sinclair Interplanetary, 2014.
- [30] *Medium CubeTorquer*, CubeSpace, 2014.
- [31] J. Auret, “Design of an aerodynamic attitude control system for a cubesat,” Ph.D. dissertation, Stellenbosch: Stellenbosch University, 2012.
- [32] W. H. Jordaan, “Spinning solar sail: the deployment and control of a spinning solar sail satellite,” Ph.D. dissertation, Stellenbosch: Stellenbosch University, 2016.
- [33] C. Jéger, “Determination and compensation of magnetic dipole moment inapplication for a scientific nanosatellite mission,” 2017.
- [34] 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.
- [35] J. Diebel, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.
- [36] J. R. Wertz, *Spacecraft attitude determination and control*. Springer Science & Business Media, 2012, vol. 73.
- [37] D. Vallado, P. Crawford, R. Hujak, and T. Kelso, “Revisiting spacetrack report# 3,” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2006, p. 6753.
- [38] B. Rhodes. Python package index - sgp4. [Online]. Available: <https://pypi.org/project/sgp4/>

- [39] D. A. Vallado, *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001, vol. 12.
- [40] F. E. Ross, “The sun’s mean longitude,” *The Astronomical Journal*, vol. 29, pp. 152–156, 1916.
- [41] 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.
- [42] 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.
- [43] 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.
- [44] 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.
- [45] B. Bialke, “High fidelity mathematical modeling of reaction wheel performance,” in *21st Annual American Astronautical Society Guidance and Control Conference*, 1998.
- [46] W. Steyn, “AODCS EKF Estimation,” 2004.
- [47] 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.
- [48] W. H. Steyn, “An attitude control system for sumbandilasat an earth observation satellite,” in *ESA 4S Symposium*, 2008, pp. 1–12.
- [49] 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.
- [50] H. M. Van Rensburg, “An infrared earth horizon sensor for a leo satellite,” Ph.D. dissertation, Stellenbosch: University of Stellenbosch, 2008.
- [51] 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.

- [52] 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.
- [53] P. J. Schmid, L. Li, M. P. Juniper, and O. Pust, “Applications of the dynamic mode decomposition,” *Theoretical and Computational Fluid Dynamics*, vol. 25, no. 1, pp. 249–259, 2011.
- [54] J. L. Proctor, S. L. Brunton, and J. N. Kutz, “Dynamic mode decomposition with control,” *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.
- [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. Colagrossi and M. Lavagna, “Fault tolerant attitude and orbit determination system for small satellite platforms,” *Aerospace*, vol. 9, no. 2, p. 46, 2022.
- [60] 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.
- [61] D. Meyer and F. Wien, “Support vector machines,” *The Interface to libsvm in package e1071*, vol. 28, p. 20, 2015.
- [62] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” pp. 413–422, 2008.
- [63] S. Hariri, M. C. Kind, and R. J. Brunner, “Extended isolation forest,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1479–1489, 2019.
- [64] H. Chen, H. Ma, X. Chu, and D. Xue, “Anomaly detection and critical attributes identification for products with multiple operating conditions based on isolation forest,” *Advanced Engineering Informatics*, vol. 46, p. 101139, 2020.

Appendix A

System Perturbation Matrix

This section will show the derivation of the Jacobian matrix \mathbf{F}_t that must be constructed in the execution of the full state EKF, as described in Section 5.4.

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

$$\mathbf{F}_t = \begin{bmatrix} \frac{\partial \dot{\omega}_B^I}{\partial \omega_B^I} & \frac{\partial \dot{\omega}_B^I}{\partial \mathbf{q}} \\ \frac{\partial \dot{\mathbf{q}}}{\partial \omega_B^I} & \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} \end{bmatrix}_{\omega_B^I = \hat{\omega}_B^I, \mathbf{q} = \hat{\mathbf{q}}}$$

Note that the subscript ' t ' indicating the time domain has been dropped from Equation 5.4.5 to simplify the derivation. The non-linear function $\mathbf{f}(\mathbf{x})$ can be separated into two parts: a non-linear function describing $\dot{\omega}_B^I$ and a non-linear function describing $\dot{\mathbf{q}}$. The continuous non-linear system equation with regards to $\dot{\omega}_B^I$ is the Euler dynamic equation, or

$$\dot{\omega}_B^I = \mathbf{J}^{-1} \left(\mathbf{N}_c + \mathbf{N}_d - \boldsymbol{\omega}_B^I \times (\mathbf{J} \boldsymbol{\omega}_B^I + \mathbf{h}_w) \right)$$

The individual components of Equation B.2 can also be expressed as

$$\begin{aligned} \dot{\omega}_{xi} &= \frac{1}{I_{xx}} (N_{cx} + N_{dx} - \omega_{yi} (I_{zz}\omega_{zi} + h_z) + \omega_{zi} (I_{yy}\omega_{yi} + h_y)) \\ \dot{\omega}_{yi} &= \frac{1}{I_{yy}} (N_{cy} + N_{dy} - \omega_{zi} (I_{xx}\omega_{xi} + h_x) + \omega_{xi} (I_{zz}\omega_{zi} + h_z)) \\ \dot{\omega}_{zi} &= \frac{1}{I_{zz}} (N_{cz} + N_{dz} - \omega_{xi} (I_{yy}\omega_{yi} + h_y) + \omega_{yi} (I_{xx}\omega_{xi} + h_x)). \end{aligned}$$

Using Equations B.3 to B.5, $\frac{\partial \dot{\omega}_B^I}{\partial \omega_B^I}$ can be determined by taking each individual partial derivative, which delivers

$$\frac{\partial \dot{\omega}_B^I}{\partial \omega_B^I} = \begin{bmatrix} 0 & \frac{\omega_{zi}(I_{yy}-I_{zz})-h_z}{I_{xx}} & \frac{\omega_{yi}(I_{yy}-I_{zz})+h_y}{I_{xx}} \\ \frac{\omega_{zi}(I_{zz}-I_{xx})+h_z}{I_{yy}} & 0 & \frac{\omega_{xi}(I_{zz}-I_{xx})-h_x}{I_{yy}} \\ \frac{\omega_{yi}(I_{xx}-I_{yy})-h_y}{I_{zz}} & \frac{\omega_{xi}(I_{xx}-I_{yy})+h_x}{I_{zz}} & 0 \end{bmatrix}.$$

$\frac{\partial \dot{\omega}_B^I}{\partial \mathbf{q}}$ is however much more difficult to determine. The first step is to determine which components of Equation B.2 are dependent on the attitude quaternion of the satellite.

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

$$\mathbf{N}_d \approx \mathbf{N}_{gg}$$

\mathbf{N}_d can thus easily be expressed in terms of quaternions using Equation 2.3.3 as

$$\begin{aligned}\mathbf{N}_{dx} &\approx k_{gx} (2[q_2 q_3 + q_1 q_4]) (-q_1^2 - q_2^2 + q_3^2 + q_4^2) \\ \mathbf{N}_{dy} &\approx k_{gy} (2[q_1 q_3 - q_2 q_4]) (-q_1^2 - q_2^2 + q_3^2 + q_4^2) \\ \mathbf{N}_{dz} &\approx k_{gz} (2[q_1 q_3 - q_2 q_4]) (2[q_2 q_3 + q_1 q_4])\end{aligned}$$

$\frac{\partial \dot{\omega}_B^I}{\partial \mathbf{q}}$ can now be calculated as

$$\frac{\partial \dot{\omega}_B^I}{\partial \mathbf{q}} = \mathbf{J}^{-1} \left[\frac{\partial \mathbf{N}_d}{\partial \mathbf{q}} \right] = \mathbf{K} \left[\begin{array}{cccc} \mathbf{d}_1 & \mathbf{d}_2 & \mathbf{d}_3 & \mathbf{d}_4 \end{array} \right]$$

where

$$\mathbf{K} = \begin{bmatrix} 2k_{gx} & 0 & 0 \\ 0 & 2k_{gy} & 0 \\ 0 & 0 & 2k_{gz} \end{bmatrix}$$

and

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

$$\dot{\mathbf{q}} = \frac{1}{2} \boldsymbol{\Omega}(\omega_B^O) \mathbf{q},$$

where

$$\boldsymbol{\Omega}(\omega_B^O) = \begin{bmatrix} 0 & \omega_{zo} & -\omega_{yo} & \omega_{xo} \\ -\omega_{zo} & 0 & \omega_{xo} & \omega_{yo} \\ \omega_{yo} & -\omega_{xo} & 0 & \omega_{zo} \\ -\omega_{xo} & -\omega_{yo} & -\omega_{zo} & 0 \end{bmatrix}$$

The relationship between ω_B^I and ω_B^O is given by

$$\omega_B^O = \omega_B^I - \mathbf{A} \begin{bmatrix} 0 \\ -\omega_o \\ 0 \end{bmatrix} = \begin{bmatrix} \omega_{xi} + \omega_o A_{12} \\ \omega_{yi} + \omega_o A_{22} \\ \omega_{zi} + \omega_o A_{32} \end{bmatrix}$$

which means that $\frac{\partial \dot{\mathbf{q}}}{\partial \omega_B^I}$ can be determined as

$$\frac{\partial \dot{\mathbf{q}}}{\partial \omega_B^I} = \begin{bmatrix} \frac{\partial \dot{q}_1}{\partial \omega_{xi}} & \frac{\partial \dot{q}_1}{\partial \omega_{yi}} & \frac{\partial \dot{q}_1}{\partial \omega_{zi}} \\ \frac{\partial \dot{q}_2}{\partial \omega_{xi}} & \frac{\partial \dot{q}_2}{\partial \omega_{yi}} & \frac{\partial \dot{q}_2}{\partial \omega_{zi}} \\ \frac{\partial \dot{q}_3}{\partial \omega_{xi}} & \frac{\partial \dot{q}_3}{\partial \omega_{yi}} & \frac{\partial \dot{q}_3}{\partial \omega_{zi}} \\ \frac{\partial \dot{q}_4}{\partial \omega_{xi}} & \frac{\partial \dot{q}_4}{\partial \omega_{yi}} & \frac{\partial \dot{q}_4}{\partial \omega_{zi}} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \hat{q}_4 & -\hat{q}_3 & \hat{q}_2 \\ \hat{q}_3 & \hat{q}_4 & -\hat{q}_1 \\ -\hat{q}_2 & \hat{q}_1 & \hat{q}_4 \\ -\hat{q}_1 & -\hat{q}_2 & -\hat{q}_3 \end{bmatrix}$$

$\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}}$ can be determined by substituting Equation B.13 into Equation B.12, which delivers

$$\begin{aligned} \dot{q}_1 &= \frac{1}{2} (q_2 (\omega_{zi} - \omega_o A_{32}) - q_3 (\omega_{yi} - \omega_o A_{22}) + q_4 (\omega_{xi} - \omega_o A_{12})) \\ \dot{q}_2 &= \frac{1}{2} (-q_1 (\omega_{zi} - \omega_o A_{32}) + q_3 (\omega_{xi} - \omega_o A_{12}) + q_4 (\omega_{yi} - \omega_o A_{22})) \\ \dot{q}_3 &= \frac{1}{2} (q_1 (\omega_{yi} - \omega_o A_{22}) - q_2 (\omega_{xi} - \omega_o A_{12}) + q_4 (\omega_{zi} - \omega_o A_{32})) \\ \dot{q}_4 &= \frac{1}{2} (-q_1 (\omega_{xi} - \omega_o A_{12}) - q_2 (\omega_{yi} - \omega_o A_{22}) - q_3 (\omega_{zi} - \omega_o A_{32})) \end{aligned}$$

$$\begin{aligned} \mathbf{d}_1 &= \begin{bmatrix} \frac{-q_1 A_{23} + q_4 A_{33}}{I_{xx}} \\ \frac{-q_1 A_{13} + q_3 A_{33}}{I_{yy}} \\ \frac{q_3 A_{23} + q_4 A_{13}}{I_{zz}} \end{bmatrix} & \mathbf{d}_2 &= \begin{bmatrix} \frac{-q_2 A_{23} + q_3 A_{33}}{I_{xx}} \\ \frac{-q_2 A_{13} - q_4 A_{33}}{I_{yy}} \\ \frac{-q_4 A_{23} + q_3 A_{13}}{I_{zz}} \end{bmatrix} \\ \mathbf{d}_3 &= \begin{bmatrix} \frac{q_3 A_{23} + q_2 A_{33}}{I_{xx}} \\ \frac{q_3 A_{13} + q_1 A_{33}}{I_{yy}} \\ \frac{q_1 A_{23} + q_2 A_{13}}{I_{zz}} \end{bmatrix} & \mathbf{d}_4 &= \begin{bmatrix} \frac{q_4 A_{23} + q_1 A_{33}}{I_{xx}} \\ \frac{q_4 A_{13} - q_2 A_{33}}{I_{yy}} \\ \frac{-q_2 A_{23} + q_1 A_{13}}{I_{zz}} \end{bmatrix}. \end{aligned}$$

By partially deriving Equations B.15 to B.18 and performing some mathematical manipulation, $\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}}$ can be calculated as

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

Appendix B

Measurement Perturbation Jacobian Matrix

This section will show the derivation of the Jacobian matrix \mathbf{H}_k that must be constructed in the execution of the full state EKF, as described in Section 5.4.

The discrete measurement perturbation matrix \mathbf{H}_k from Equation 5.4.12 can be determined by partially deriving the non-linear function $\mathbf{h}(\mathbf{x}_k)$, which is given by Equation 5.4.15 as

$$\mathbf{h}(\mathbf{x}_k) = \mathbf{A}\mathbf{v}_{\text{model},k}.$$

Since \mathbf{A} is constructed from \mathbf{q} only, Equation B.20 suggests that $\mathbf{h}(\mathbf{x}_k)$ is independent of ω_B^I , thus

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

\mathbf{H}_k can thus be calculated as

$$\mathbf{H}_k = \left[\mathbf{0}_{[3 \times 3]} \quad \mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3 \quad \mathbf{h}_4 \right],$$

$$\text{where } \mathbf{h}_1 = 2 \begin{bmatrix} q_1 & q_2 & q_3 \\ q_2 & -q_1 & q_4 \\ q_3 & -q_4 & -q_1 \end{bmatrix} \mathbf{v}_{\text{model},k},$$

$$\mathbf{h}_2 = 2 \begin{bmatrix} -q_2 & q_1 & -q_4 \\ q_1 & q_2 & q_3 \\ q_4 & q_3 & -q_2 \end{bmatrix} \mathbf{v}_{\text{model},k}$$

$$\mathbf{h}_3 = 2 \begin{bmatrix} -q_3 & q_4 & q_1 \\ -q_4 & -q_3 & q_2 \\ q_1 & q_2 & q_3 \end{bmatrix} \mathbf{v}_{\text{model},k}$$

$$\text{and } \mathbf{h}_4 = 2 \begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \end{bmatrix} \mathbf{v}_{\text{model},k}$$

Appendix C

System Noise Covariance Matrix

This section will show the derivation of the covariance matrix \mathbf{Q}_k that must be constructed in the execution of the full state EKF, as described in Section 5.4.

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 Φ_k to be approximated using only two terms without significant inaccuracies, thus $\Phi_k \approx \mathbf{I} + [T_s \mathbf{F}_t]_{t=k_s}$.
- The angular rate noise for each axis is equal, thus $\sigma_{\omega x} = \sigma_{\omega y} = \sigma_{\omega z} = \sigma_\omega$.

Given the above-mentioned assumptions, the angular rate noise covariance matrix $\mathbf{Q}_{\omega,t}$ is given as

$$\mathbf{Q}_{\omega,t} = \begin{bmatrix} \sigma_\omega^2 & 0 & 0 \\ 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & \sigma_\omega^2 \end{bmatrix}$$

\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 ($\mathbf{Q}_{q,t}$) is simply a zero matrix, or

$$\mathbf{Q}_{q,t} = \mathbf{0}_{[4 \times 4]}$$

\mathbf{Q}_t can be formed from Equations B.23 and B.24 as

$$\begin{aligned} \mathbf{Q}_t &= \begin{bmatrix} \mathbf{Q}_{\omega,t} & \mathbf{0}_{[3 \times 4]} \\ \mathbf{0}_{[4 \times 3]} & \mathbf{Q}_{q,t} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{Q}_{\omega,t} & \mathbf{0}_{[3 \times 4]} \\ \mathbf{0}_{[4 \times 3]} & \mathbf{0}_{[4 \times 4]} \end{bmatrix} \end{aligned}$$

\mathbf{F}_t can also be express in the form of Equation B.26 as

$$\mathbf{F}_t = \begin{bmatrix} \mathbf{F}_{11[3 \times 3]} & \mathbf{F}_{12[3 \times 4]} \\ \mathbf{F}_{21[4 \times 3]} & \mathbf{F}_{22[4 \times 4]} \end{bmatrix}$$

\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

$$\mathbf{Q}_k = T_s \mathbf{S}_1 + \frac{1}{2} T_s^2 \mathbf{S}_2 + \frac{1}{3} T_s^3 \mathbf{S}_3$$

where $\mathbf{S}_1 = \mathbf{Q}_t$

$$\begin{aligned} \mathbf{S}_2 &= \left[\begin{array}{cc} \mathbf{Q}_{\omega,t} \mathbf{F}_{11}^T + \mathbf{F}_{11} \mathbf{Q}_{\omega,t} & \mathbf{Q}_{\omega,t} \mathbf{F}_{21}^T \\ \mathbf{F}_{21} \mathbf{Q}_{\omega,t} & \mathbf{0}_{[4 \times 4]} \end{array} \right]_{t=kT_s} \\ \mathbf{S}_3 &= \left[\begin{array}{cc} \mathbf{F}_{11} \mathbf{Q}_{\omega,t} \mathbf{F}_{11}^T & \mathbf{F}_{11} \mathbf{Q}_{\omega,t} \mathbf{F}_{21}^T \\ \mathbf{F}_{21} \mathbf{Q}_{\omega,t} \mathbf{F}_{11}^T & \mathbf{F}_{21} \mathbf{Q}_{\omega,t} \mathbf{F}_{21}^T \end{array} \right]_{t=kT_s} \end{aligned}$$

The computational load of calculating \mathbf{Q}_k can be reduced if the assumption is made that $\mathbf{F}_{11} \ll \mathbf{F}_{21}$ [21]. \mathbf{Q}_k can then be simplified to

$$\mathbf{Q}_k = \left[\begin{array}{cc} T_s \mathbf{Q}_{\omega,t} & \frac{1}{2} T_s^2 \mathbf{Q}_{\omega,t} \mathbf{F}_{21}^T \\ \frac{1}{2} T_s^2 \mathbf{F}_{21} \mathbf{Q}_{\omega,t} & \frac{1}{3} T_s^3 \mathbf{F}_{21} \mathbf{Q}_{\omega,t} \mathbf{F}_{21}^T \end{array} \right]_{t=kT_s}$$

Appendix D

Measurement Noise Covariance Matrix

This section will show the derivation of the covariance matrix \mathbf{R}_k that must be constructed in the execution of the full state EKF, as described in Section 5.4.

The relationship between the true measured vector $\bar{\mathbf{v}}_{\text{meas},k}$ and the true modelled vector $\bar{\mathbf{v}}_{\text{model},k}$ is given by Equation 5.4.13 as

$$\bar{\mathbf{v}}_{\text{meas},k} = \mathbf{A}(\mathbf{q}_k) \bar{\mathbf{v}}_{\text{model},k}.$$

It should be noted that $\mathbf{A}_k = \mathbf{A}(\mathbf{q}_k)$. The added notation, which merely implies that \mathbf{A} 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

$$\begin{aligned} \mathbf{v}_{\text{meas},k} &= \bar{\mathbf{v}}_{\text{meas},k} + \mathbf{m}_{\text{meas},k} \\ \text{and } \mathbf{v}_{\text{model},k} &= \bar{\mathbf{v}}_{\text{model},k} + \mathbf{m}_{\text{model},k} \end{aligned}$$

If $\Delta\mathbf{q}_k$ is defined as the difference between the true quaternion \mathbf{q}_k and the estimated quaternion $\hat{\mathbf{q}}_k$,

$$\Delta\mathbf{q}_k = \mathbf{q}_k - \hat{\mathbf{q}}_k$$

then Equation 5.4.13 can also be expressed as

$$\mathbf{v}_{\text{meas},k} - \mathbf{m}_{\text{meas},k} = \mathbf{A}(\hat{\mathbf{q}}_k + \Delta\mathbf{q}_k)(\mathbf{v}_{\text{model},k} - \mathbf{m}_{\text{model},k})$$

The Taylor series expansion from Equations 5.4.4 and 5.4.11 can also be used to approximate $\mathbf{A}(\hat{\mathbf{q}}_k + \Delta\mathbf{q}_k)$ as

$$\begin{aligned} \mathbf{A}(\hat{\mathbf{q}}_k + \Delta\mathbf{q}_k) &\approx \mathbf{A}(\hat{\mathbf{q}}_k) + \mathbf{C}_k \Delta\mathbf{q}_k \\ \text{where } \mathbf{C}_k &= \left[\frac{\partial \mathbf{A}(\hat{\mathbf{q}}_k)}{\partial \hat{\mathbf{q}}_k} \right] \end{aligned}$$

Substituting Equation B.37 into Equation B.36 delivers

$$\mathbf{v}_{\text{meas},k} - \mathbf{m}_{\text{meas},k} = (\mathbf{A}(\hat{\mathbf{q}}_k) + \mathbf{C}_k \Delta\mathbf{q}_k)(\mathbf{v}_{\text{model},k} - \mathbf{m}_{\text{model},k})$$

Given that the innovation \mathbf{e}_k is defined by Equation 5.4.17 as

$$\mathbf{e}_k = \mathbf{v}_{\text{meas},k} - \mathbf{A}(\hat{\mathbf{q}}_k) \mathbf{v}_{\text{model},k},$$

substitution can be used to manipulate Equation B.39 into

$$\mathbf{e}_k = (\mathbf{C}_k \mathbf{v}_{\text{model},k} - \mathbf{C}_k \mathbf{m}_{\text{model},k}) \Delta \mathbf{q}_k + \mathbf{m}_{\text{meas},k} - \mathbf{A}(\hat{\mathbf{q}}_k) \mathbf{m}_{\text{model},k}$$

If it is assumed that $\mathbf{m}_{\text{model},k}$ and $\Delta \mathbf{q}_k$ are extremely small compared respectively to $\mathbf{v}_{\text{model},k}$ and \mathbf{q}_k , then

$$\mathbf{m}_{\text{model},k} \Delta \mathbf{q}_k \approx 0$$

Equation B. 40 can thus be simplified to

$$\mathbf{e}_k = [\mathbf{0}_{[3 \times 3]} \mathbf{C}_k \mathbf{v}_{\text{model},k}] \Delta \mathbf{x}_k + \mathbf{m}_k,$$

$$\text{where } \mathbf{m}_k = \mathbf{m}_{\text{meas},k} - \mathbf{A}(\hat{\mathbf{q}}_k) \mathbf{m}_{\text{model},k}$$

The covariance matrix \mathbf{R}_k of \mathbf{m}_k is defined as

$$\begin{aligned} \mathbf{R}_k &= \mathbb{E} \left\{ (\mathbf{m}_k) (\mathbf{m}_k)^T \right\} \\ &= \mathbb{E} \left\{ (\mathbf{m}_{\text{meas},k} - \mathbf{A}(\hat{\mathbf{q}}_k) \mathbf{m}_{\text{model},k}) (\mathbf{m}_{\text{meas},k} - \mathbf{A}(\hat{\mathbf{q}}_k) \mathbf{m}_{\text{model},k})^T \right\} \\ &= \mathbb{E} \left\{ \mathbf{m}_{\text{meas},k} \mathbf{m}_{\text{meas},k}^T - \right. \\ &\quad \mathbf{m}_{\text{meas},k} \mathbf{m}_{\text{model},k}^T \mathbf{A}(\hat{\mathbf{q}}_k)^T - \\ &\quad \mathbf{A}(\hat{\mathbf{q}}_k) \mathbf{m}_{\text{model},k} \mathbf{m}_{\text{meas},k}^T + \\ &\quad \left. \mathbf{A}(\hat{\mathbf{q}}_k) \mathbf{m}_{\text{model},k} \mathbf{m}_{\text{model},k}^T \mathbf{A}(\hat{\mathbf{q}}_k)^T \right\} \\ &= \mathbb{E} \left\{ \mathbf{m}_{\text{meas},k} \mathbf{m}_{\text{meas},k}^T \right\} - \\ &\quad \mathbb{E} \left\{ \mathbf{m}_{\text{meas},k} \mathbf{m}_{\text{model},k}^T \mathbf{A}(\hat{\mathbf{q}}_k)^T \right\} - \\ &\quad \mathbb{E} \left\{ \mathbf{A}(\hat{\mathbf{q}}_k) \mathbf{m}_{\text{model},k} \mathbf{m}_{\text{meas},k}^T \right\} + \\ &\quad \mathbb{E} \left\{ \mathbf{A}(\hat{\mathbf{q}}_k) \mathbf{m}_{\text{model},k} \mathbf{m}_{\text{model},k}^T \mathbf{A}(\hat{\mathbf{q}}_k)^T \right\} \end{aligned}$$

where \mathbb{E} indicates the expected value operator. The last term of Equation B.44 can be simplified to

$$\mathbf{A}(\hat{\mathbf{q}}_k) \mathbf{m}_{\text{model},k} \mathbf{m}_{\text{model},k}^T \mathbf{A}(\hat{\mathbf{q}}_k)^T = \mathbf{m}_{\text{model},k} \mathbf{m}_{\text{model},k}^T$$

since $\mathbf{m}_{\text{model},k} \mathbf{m}_{\text{model},k}^T$ is a scalar value and $\mathbf{A}(\hat{\mathbf{q}}_k) \mathbf{A}(\hat{\mathbf{q}}_k)^T = 1$. 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 B. 44 becomes

$$\begin{aligned}\mathbf{R}_k &= \mathbb{E} \left\{ \mathbf{m}_{\text{meas},k} \mathbf{m}_{\text{meas},k}^T \right\} + \\ &\quad \mathbb{E} \left\{ \mathbf{m}_{\text{model},k} \mathbf{m}_{\text{model},k}^T \right\} \\ &= (\sigma_{\text{meas}}^2 + \sigma_{\text{model}}^2) \mathbf{I}_{3 \times 3}\end{aligned}$$

where σ_{meas} and σ_{model} are the respective standard deviations of $\mathbf{m}_{\text{meas},k}$ and $\mathbf{m}_{\text{model},k}$. It is also assumed that σ_{meas} and σ_{model} are constant.