# Diagnosis of 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

October 2099

# 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 is as my own.*

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

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

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

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

| | |
|---|---|
| Studentenommer / *Student number* | Handtekening / *Signature* |
| Voorletters en van / *Initials and surname* | Datum / *Date* |

# Abstract


### English

The English abstract.

### Afrikaans

Die Afrikaanse uittreksel.

# Contents

# List of Figures

# List of Tables

# Nomenclature

### Variables and functions

| | |
|---|---|
| $p(x)$ | Probability density function with respect to variable $x$. |
| $P(A)$ | Probability of event $A$ occurring. |
| $\varepsilon$ | The Bayes error. |
| $\varepsilon_u$ | 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 $i^{\text{th}}$ 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 $s$. |
| $\mu$ | Statistical mean vector. |
| $\Sigma$ | 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 $\mu$ and covariance matrix $\Sigma$. |
| $a_{ij}$ | 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

| | |
|---|---|
| AE | Afrikaans English |
| AID | accent identification |
| ASR | automatic speech recognition |
| AST | African Speech Technology |
| CE | Cape Flats English |
| DCD | dialect-context-dependent |
| DNN | deep neural network |
| G2P | grapheme-to-phoneme |
| GMM | Gaussian mixture model |
| HMM | hidden Markov model |
| HTK | Hidden Markov Model Toolkit |
| IE | Indian South African English |
| IPA | International Phonetic Alphabet |
| LM | language model |
| LMS | language model scaling factor |
| MFCC | Mel-frequency cepstral coefficient |
| MLLR | maximum likelihood linear regression |
| OOV | out-of-vocabulary |
| PD | pronunciation dictionary |
| PDF | probability density function |
| SAE | South African English |
| SAMPA | Speech Assessment Methods Phonetic Alphabet |

# Chapter 1

# Introduction

The current trend in industry is to create systems that operate autonomously. There is also a trend seen where multi-agent systems with hundred and thousands of sub-systems work within a larger system. Even though each sub-system can operate independently, the health of the network/system is dependent on each agent within the system operating as desired. Consequently, autonomous systems must be able to detect faults within the system. To attain this, a fault detection, isolation and recovery system is required for each individual agent and the system as a whole.

## 1.1. Background

All systems have the inevitable problem that they will fail. Determining this failure is of a variety of importance for multiple systems. The current tendency is that many systems are more and more integrated. A web of self-driving cars, UAV's for delivery, satellite constellations and more. The problem with any of these systems is if any part of the system is faulty it can have detrimental consequences for a part of the system or the entire system.

Software developers make mistakes, as all humans do. The current industry standard for number of bugs per thousand lines of code is 5 bugs/KLOC at $5 per line of code. NASA works on 0.004 bugs/KLOC at $805 per line of code. Imagine having hundreds of thousands of lines of code and having that same code on thousands of satellites. Each fault detection must usually be checked with if statements and logic. This leads to more lines of code and mistakes on trying to capture mistakes. I propose a solution to this problem by using statistical models and machine learning to use satellites in close proximity to determine the health of the other nearby components within the system.

## 1.2. Problem description

https://www.theverge.com/2020/1/14/21043229/spacex-starlink-satellite-mega-constellation-concerns-astronomy-space-traffic

The most detrimental effect could be on satellite constellations. Satellites are expensive,

$500000 each for Starlink satellite. And this is also based on the mass production of satellites. Satellites cannot merely be accessed and fixed as other systems. Therefore satellites are used as the specific system for FDIR of systems.

The current situation with cubesats is that ground stations will not be able to keep up with the fault detection and diagnosis of cubesats within constellations where thousands of cubesats are within the same height above the earth. As is the current situation with Starlink.

Kessler syndrome is the effect of one satellite, that is unable to control its orientation and position, causing a collision in orbit. This leads to more debris in the orbit and more collisions and this is detrimental if considered in view of Starlink's mega-constellation.

## 1.3. Research hypothesis

I propose a fault detection system for each individual satellite and the constellation as a whole. The satellite will have an on-board FDIR system that also uses the information of the satellites closest to its position to provide feedback of its own "health" and the health of the other satellites in its orbit. If a satellite is determined "unhealthy" by all the nearby satellites then the satellite with go into safe mode until the ground station can determine the problem with current satellite.

## 1.4. Scope and objectives

The following objectives will be pursued in this project/thesis/dissertation:

I To *conduct* a thorough survey of the literature related to:

(a) facility location problems in general,

(b) models for the placement of a network of radio transmitters in particular,

(c) the nature of parameters required to describe effective radio transmission, and

(d) terrain elevation data required to generate an instance of the bi-objective radio transmitter location problem described in the previous section.

II To *establish* an suitable framework for evaluating the effectiveness of a given set of placement locations for a network of radio transmitters in respect of its total area coverage and its mutual area coverage.

III To *formulate* a bi-objective facility location model suitable as a basis for decision support in respect of the location of a network of radio transmitters with a view to identify high-quality trade-offs between maximising total coverage area and maximising mutual coverage area. The model should take as input the parameters

and data identified in Objective I(c)–(d) and function within the context of the framework of Objective II.

IV To *design* a generic *decision support system* (DSS) capable of suggesting high-quality trade-off locations for user-specified instances of the bi-objective radio transmitter location problem described in the previous section. This DSS should incorporate the location model of Objective III.

V To *implement* a concept demonstrator of the DSS of Objective IV in an applicable software platform. This DSS should be flexible in the sense of being able to take as input an instance of the bi-objective radio transmitter location problem described in the previous section via user-specification of the parameters and data of Objectives I(c)–(d) and produce as output a set of high-quality trade-off transmitter locations for that instance.

VI To *verify* and validate the implementation of Objective V according to generally accepted modelling guidelines.

VII To *apply* the concept demonstrator of Objective V to a special case study involving realistic radio transmission parameters and real elevation data for a specified portion of terrain.

VIII To *evaluate* the effectiveness of the DSS and associated concept demonstrator of Objectives IV–VI in terms of its capability to identify a set of high-quality trade-off solutions for a network of radio transmitter locations.

IX To *recommend* sensible follow-up work related to the work in this project which may be pursued in future.

## 1.5. Research methodology

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.

## 1.6. 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.

# Chapter 2

# Literature Study

The implementation of FDIR on satellites have multiple complications with regards to the type of data generated by a satellite and the methodologies that can be implemented within the time and memory constraint of a cube-sat processor.

## 2.1. Anomaly Detection on Satellites

Various methodologies have been tested on different component of satellites. Therefore a summary of these research articles are provided in this section.

### 2.1.1. Analysis and Prediction of Satellite Anomalies

### 2.1.2. Agent-based algorithm for fault detection and recovery of gyroscope's drift in small satellite missions

To ensure that the ADCS of satellites are autonomous every aspect of the control must be able to recover from faults. [**?**] developed an algorithm to evaluate the control of a gyroscope and detect whether drifting exists. If drifting is detected another algorithm is deployed to ensure the recovery of the gyroscope drift by updating the error state vector.

### 2.1.3. Multivariate Anomaly Detection in Discrete and Continuous Telemetry Signals Using a Sparse Decomposition in a Dictionary

[**?**]

### 2.1.4. Fault isolation of reaction wheels onboard three-axis controlled in-orbit satellite using ensemble machine learning

[**?**]

### 2.1.5. Fault tolerant control for satellites with four reaction wheels

[?]

### 2.1.6. Innovative Fault Detection, Isolation and Recovery Strategies On-Board Spacecraft: State of the Art and Research Challenges

[?]

### 2.1.7. Machine learning methods for spacecraft telemetry mining

[?]

### 2.1.8. Machine learning techniques for satellite fault diagnosis

[?]

### 2.1.9. Satellite fault diagnosis using a bank of interacting Kalman filters

[?]

### 2.1.10. A scheme of satellite multi-sensor fault-tolerant attitude estimation

[?] implements a fault tolerant federated Kalman filter with three sub-filters for multi-sensor fault estimation.

### 2.1.11. Detection of satellite attitude sensor faults using the UKF

[?] provides 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.

### 2.1.12. Sensor fault detection and recovery in satellite attitude control

[?]

### 2.1.13. Sensor Failure Detection in Dynamical Systems by Kalman Filtering Methodology

While methods for sensor failure detection in other dynamical systems has also been developed which includes kalman filter methodology [**?**],

### 2.1.14. Sensors Anomaly Detection of Industrial Internet of Things Based on Isolated Forest Algorithm and Data Compression

isolation forests [**?**] and using LSTM on sensor data to detect anomalies on machines

### 2.1.15. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection

[**?**]

### 2.1.16. Sensor fault detection and isolation using adaptive extended Kalman filter

[**?**]

## 2.2. Statistical Methods

### 2.2.1. Pearson Correlation

Vectors of certain sensors are highly correlated. For instance the vector of the earth sensor is highly correlated since the magnitude of the vector remains more or less constant. To detect anomalies the correlation of vectors can be measured and with a specified threshold the correlation can be indicated as a anomaly or nor.

The squared Pearson correlation coefficient (SPCC) for vectors depicted as

$$a = [a_1, a_2, \ldots, a_L]^T,$$
$$b = [b_1, b_2, \ldots, b_L]^T,$$

is defined as [**?**]

$$\rho^2(a,b) = \frac{E^2(a,b)}{E(a^T a)E(b^T b)}. \tag{2.1}$$

The correlation coefficient is proven to be constraint as

$$0 \leq \rho \leq 1, \tag{2.2}$$

where $\rho = 1$ is perfect linear correlation.

### 2.2.2. Variance

Within a sequential data sample of the satellite, the variance of the variables should be within a given threshold if the satellite is in a stable condition. The variance of the data sample is defined as

$$S^2 = \frac{\sum(x_i + \bar{x})^2}{n-1} \tag{2.3}$$

where $x$ defines the variable within the dataset.

### 2.2.3. Kalman-Filter

The Kalman-filter application would require the state-space matrices to be provided in the log file.

### 2.2.4. Multivariate Guassian Distribution

The assumption that the error of our data is generated with a Guassian distribution with a specific mean, $\mu$, and variance, $\sigma^2$, provides the opportunity for using multi-variate Gaussian distribution to determine the probability of a data-sample within a dataset.

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} \tag{2.4}$$

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)} - \mu_j)^2 \tag{2.5}$$

$$p(x) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} exp(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}) \tag{2.6}$$

For multi-variate Guassian distribution [**?**].

$$\sum = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)(x^{(i)} - \mu)^T \tag{2.7}$$

$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}}|\sum|^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu)^T\sum^{-1}(x-\mu)) \tag{2.8}$$

The Anomalies will be classified based on probabilities smaller than a given threshold $p(x) < \epsilon$.

**Algorithm 2.1:** Multi-variate Guassian Distribution Algorithm

1: Determine feature vectors $x_i$
2: Determine threshold probabilty, $\epsilon$
3: Calculate $\mu_j$ with Eq 2.4
4: Calculate $\sigma_j$ with Eq 2.5
5: Calculate $p(x)$ with Eq 2.6
6: **if** $p(x) < \epsilon$ **then**
7: Anomaly $= True$
8: **else**
9: Anomaly $= False$
10: **end if**

### 2.2.5. Kullback-Leibler Divergence

The Kullback-Leibler divergence quantifies the difference between two probability density functions, denoted as $p(x)$ and $q(x)$ [**?**]. Satellites are systems that are predictable within a time-series. The divergence between two sequential data buffers from the satellite will have a very similar probability distribution. Therefore calculating the difference between two datasets can be used to detect an anomaly based on a given threshold.

The difference between the probability distributions from datasets, $a$ and $b$, in Figure 2.1 cannot simply be calculated as the difference in the mean or the difference in the variance. To overcome this, the divergence between the two distributions can be calculated. Intuitively a point $x$ with a high probability in the dataset $a$ should have a high probability in the dataset $b$ if the two datasets have a small divergence.

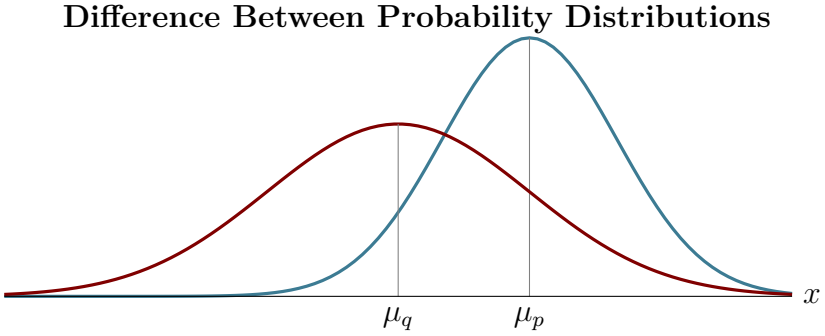



**Figure 2.1:** Guassian Distributions

The divergence can be expressed as

$$KL(P||Q) = \int p(x) \log\left(\frac{q(x)}{p(x)}\right) dx. \tag{2.9}$$

### 2.2.6. Canonical Correlation Analysis

Due to the orbital nature of satellites there exist a correlation between various sensors. For instance the sun sensor, magnetometer and earth sensor are correlated based on the desired

orientation and orbit of the satellite. This correlation might not be of linear nature, but with non-linear correlation methods such as kernel canonical correlation the correlation can be measured.

However, canonical correlation provides the measure of correlation between a multi-dimensional variable with another multi-dimensional variable. Although this seems profitable for satellite fault detection, it will only be applicable for each the comparison between individual sensors. This will indicate the non-linear correlation of the sun sensor with regards to the magnetometer. The problem however, according to [**?**] is to, determine the appropriate threshold for which to classify a fault. [**?**] proposed a method for determining the appropriate threshold on page 5, algorithm 1. [**?**] [**?**]

Python - Pyrcca package

**K-means-based**

**Guassian Mixture Model**

**Just-In-Time-Learning**

[**?**]

## 2.3. Feature Extraction

To https://towardsdatascience.com/feature-extraction-techniques-d619b56e31be

### 2.3.1. Prony's Method

### 2.3.2. Convolutional Networks

### 2.3.3. K-means Clustering

K-clustering: Clustering multiple points with similar features.

### 2.3.4. Partial Least Square

### 2.3.5. Locally Linear Embedding

### 2.3.6. t-Distributed Stochastic Neighbor Embedding

### 2.3.7. Dynamic Mode Decomposition

The proposed method by [**?**] uses Dynamic Mode Decomposition (DMD), which was initially developed by [**?**] and further expanded to include control by [**?**], 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.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k \tag{2.10}$$

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

$$\mathbf{A} = \mathbf{X}\mathbf{X}^\dagger \tag{2.11}$$

This necessitates data of the state vector over time. The article by [**?**] 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.

$$\mathbf{X}_{k+1} = \mathbf{A}\mathbf{X}_k + \mathbf{B}\mathbf{Y}_k \tag{2.12}$$

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 2.12 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 [**?**] the model is adjusted with a Kalman Filter. From $\mathbf{A}$ and $\mathbf{B}$ the Kalman filter can be implemented to predict $\mathbf{X}_{k+1}$

$$\hat{\mathbf{X}}_{k+1} = \mathbf{A}\hat{\mathbf{X}}_k + \mathbf{B}\mathbf{Y}_k + K(\mathbf{X}_k - \hat{\mathbf{X}}_k) \tag{2.13}$$

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

$$\mathbf{V}_k = \frac{1}{N}\sum_{i=k-N}^{k}(\mathbf{X}_i - \hat{\mathbf{X}}_i)(\mathbf{X}_i - \hat{\mathbf{X}}_i)^T \tag{2.14}$$

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 $\mathbf{X}$.

## 2.4. Supervised Learning

Supervised learning consists of models that are trained on labelled data. This is not a problem with simulation, but with the real data, it is a problem and to provide tests on the real data to label it must be proficient. If unsupervised learning and statistical methods are not sufficient in their accuracy, a method for labelling the real data must be provided.

### 2.4.1. Decision Trees

For decision trees in the use of classification we use the Gini score

$$G = \sum_{k=1}^{K} \hat{p}_k(1 - \hat{p}_k) \tag{2.15}$$

### 2.4.2. Random Forests

[?, ?, ?]

### 2.4.3. Long Short Term Memory

Time-series data: LSTM or DLSTM

### 2.4.4. Support Vector Machines

Support Vector Machines

### 2.4.5. Naive Bayes

Naive Bayes

### 2.4.6. K-nearest neighbours

K-nearest neighbours

### 2.4.7. Artificial Neural Networks

Artificial Neural Networks

## 2.5. Unsupervised Learning

Density-based, distance, Clustering

### 2.5.1. Kernel Adaptive Density-based

Kernel adaptive density-based: Is an algorithm that uses the density factor of a data point relative to other data points to determine whether the data point is an outlier or not.

### 2.5.2. Loda

Loda: Is a fast and efficient anomaly detection algorithm that used histograms to evaluate data points to determine whether a data point is an outlier. Loda is an on-line method and not a batch method.

### 2.5.3. Robust-kernel Density Estimation

Robust-kernel density estimation

## 2.6. Reinforcement Learning

Active Anomaly detection with meta-policy (Meta-AAD) is a deep reinforcement learning approach that is based on the actor-critic model. The agent must query data points within the given dataset (where the queried point is the data top 1 data point). The query is given to a human

## 2.7. Summary

# Chapter 3

# Simulation

To implement and research various FDIR systems on satellites an simulation of satellite dynamics and kinematics is developed. The focus of this thesis is on small satellites and more specifically cubesats. For the simulation of the ADCS of the satellite [**?**, **?**, **?**] were referenced during the development of the satellite simulation. The simulation was developed in Python to simulate the dynamics and kinematics during a satellite orbit. The faults for the subsystems are also developed within the simulation and will be discussed within this chapter.

## 3.1. Attitude Determination and Control System

For the mission of the specific satellite in this document the main operational goal of the Attitude Determination and Control System (ADCS) on this specific satellite mission is to control the payload to point towards the centre of the earth.

### 3.1.1. Coordinate Frames

The coordinate frames in aerospace is a fundamental part of the ADCS. To determine the orientation and position of an object, it should be relative to a fixed frame. Consequently, the Earth inertial coordinate (EIC) frame is the fixed frame from which every other frame is relative to.

A coordinate frame consists of three orthogonal vectors which is commonly referred to as x, y, and z. The axis of the coordinate frame is appropriately named as X-axis, Y-axis and Z-axis as seen in Figure... A vector (r) within the current coordinate frame can thus be expressed as

$$\overrightarrow{r} = x\,\overrightarrow{i} + y\,\overrightarrow{j} + z\,\overrightarrow{k} \tag{3.1}$$

where the magnitude of $\overrightarrow{r}$ is denoted as $|\overrightarrow{r}|$ and is equal to

$$|\overrightarrow{r}| = \sqrt{x^2 + y^2 + z^2}. \tag{3.2}$$

The Earth-centered coordinate frames are dived into two, namely the EIC and earth fixed coordinate (EFC) frame. EFC is fixed to the earth and rotates with it. This frame

is important with respect to where the satellite position is with regards to position's on earth, such as the ground station. It is also import for the modelling of the geomagnetic fields.

The EIC is defined as the Z-axis pointing towards the north pole, the X-axis pointing towards the Vernal Equinox, $\Upsilon$, and the Y-axis completing the orthogonal set. The EFC is a copy of the EIC, with the Z-axis being identical, however the EFC 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), $\alpha_G$. With the knowledge of $t$ — the elapsed time since $t_0$, $w_E$ — the angular rate of the earth, and $\alpha_{G,0}$ — the GHA at $t = t_0$, $\alpha_G$ can be calculated as

$$\alpha_G = w_E t + \alpha_{G,0} \tag{3.3}$$

To transform a vector from one coordinate frame to another, a transformation matrix, $\boldsymbol{A}$, is required. For example vector $\overrightarrow{r}_{EFC}$ can be transformed to $\overrightarrow{r}_{EIC}$ with

$$\overrightarrow{r}_{EIC} = \boldsymbol{A}_{EFC}^{EIC} \overrightarrow{r}_{EFC} \tag{3.4}$$

with $\boldsymbol{A}_{EFC}^{EIC}$ being the EFC-to-EIC transformation matrix. Due to the definition of both coordinate frames, $\boldsymbol{A}_{EFC}^{EIC}$ can be defined .

$$\boldsymbol{A}_{EFC}^{EIC} = \begin{bmatrix} cos(\alpha_G) & -sin(\alpha_G) & 0 \\ sin(\alpha_G) & cos(\alpha_G) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

To determine the satellite position, satellite coordinate frames must be used. Three satellite-centred coordinate frames are used, namely the inertial-reference coordinate frame (the satellite does not rotate), the orbit-referenced coordinate (ORC) frame and the satellite body coordinate (SBC) frame. The IRC frame is only acknowledged, since it is the frame that is fixed (as it does not rotate around the centre of the satellite), however it changes position with the orbit of the satellite. This frame is not used to determine the position of the satellite and will not be referenced for the remainder of this document.

The ORC frame changes location as the satellite moves, however the Z-axis is always pointing towards the centre of the earth, with the Y-axis being the 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, $\overrightarrow{r}_{sat}$ and the unit velocity vector, $\overrightarrow{v}_{sat}$ in EIC [**?**].

$$\boldsymbol{A}_{EIC}^{ORC} = \begin{bmatrix} \hat{u} & \hat{v} & \hat{w} \end{bmatrix}^T \tag{3.6}$$

where

$$\hat{w} = -\frac{r_{sat}}{||r_{sat}||} \tag{3.7}$$

$$\hat{v} = -\frac{r_{sat} \times v_{sat}}{||r_{sat} \times v_{sat}||} \tag{3.8}$$

$$\hat{u} = \hat{v} \times \hat{w} \tag{3.9}$$

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. For the transformation of a vector from the ORC to SBC frame, the direct cosine matrix (DCM) also referred to as $\boldsymbol{A}$ or $\boldsymbol{A}_{ORC}^{SBC}$ is used. For the remainder of the document the DCM will be referred to as $\boldsymbol{A}_{ORC}^{SBC}$ to avoid any confusion. The calculation of this transformation matrix will be discussed in §3.1.2.

### 3.1.2. Attitude

To determine the attitude of an object, a 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 axis, that change orientation with the rotation of the object. The three axes, denoted by X, Y and Z rotate with the object as depicted in Figure **??**.

Euler angles and DCM will be acknowledged and explained why they are not applicable. In some literature the first and last quaternion are swapped.

### 3.1.3. Satellite Kinematics and Dynamics

### 3.1.4. Rungka-kutta

The integration method used in the simulation is the 4th order Rungka-Kutta method to solve the differential equations.

**Algorithm 3.2:** Multi-variate Guassian Distribution Algorithm

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

where h is the step height, which is set to $T_s/10$, $T_s = 1$.

## 3.2. Environment

### 3.2.1. Earth Orbit

Earth orbit according to sgp4 and also the placement of the earth sensor.

Show plot of 3D earth orbit...

### 3.2.2. Sun

The calculations for the sun position and also the placement of the coarse and fine sun sensor.

Show graph of sun plot...

### 3.2.3. Geomagnetic field

$$V(r_s, \theta, \lambda) = R_E \sum_{n=1}^{k} \left( \frac{R_E}{r_s}^{n+1} \right) \sum_{m=0}^{n} \left( g_n^m cos(m\lambda) + h_n^m sin(m\lambda) \right) P_n^m(\theta) \tag{3.10}$$

Show graph of geomagnetic plot...

## 3.3. Sensor models

### 3.3.1. Position of Sensors and Field of View

### 3.3.2. Noise

## 3.4. Disturbance models

### 3.4.1. Gravity Gradient

### 3.4.2. Aerodynamic Disturbance

[?]

### 3.4.3. Wheel Imbalance

## 3.5. Attitude Determination

In this section discuss the Kalman filter.

### 3.5.1. Extended Kalman Filter

The implementation of the estimated kalman filter *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 general form for a system model can be expressed as

$$\dot{x_t} = \mathbf{f}(x_t) + s_t \tag{3.11}$$

where $\mathbf{f}(x_t)$ is a non-linear function of $x_t$. The state vector, $x$, for the full 7-state EKF consists of the quaternion vector, $q$ and the inertial-referenced angular velocity, $\omega_B^I$.

$$x = [q, \omega_B^I]^T \tag{3.12}$$

The estimated state vector $x$ will be denoted as $\hat{x}$ and the estimated vector before and after the measurement update will be indicated with a superscript $'-'$ and $'+'$ respectively. To calculate the model update the dynamics and kinematics of the system model is used to calculate both $\omega_B^I$ and $q$. The integration method used in the simulation is the 4th order Rungka-Kutta method to solve the differential equations. The integration method is shown in 3.3 where $(\hat{\omega}_B^I)_k^-$ is calculated for the first step of the model update.

**Algorithm 3.3:** Rungka-Kutta 4th order Algorithm at time $k$

Satellite Body Inertia $\mathbf{J} = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.45 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$
Timestep $(T_s) = 1s$
Number of iterations $(I) = 10$
Step size $h = \frac{T_s}{h}$
Disturbance torques $N_d = N_{gg} + N_{aero} + N_{rw} - N_{gyro}$
Control torques $N_c = N_m - N_w$
$\mathbf{N} = N_c + N_d$
**for** $n \in I$ **do**
    $k_1 = h(\mathbf{J^{-1}N})$
    $k_2 = h(\mathbf{J^{-1}N} + \frac{k1}{2})$
    $k_3 = h(\mathbf{J^{-1}N} + \frac{k_2}{2})$
    $k_4 = h(\mathbf{J^{-1}N} + k_3)$
    $\omega_{n+1} = \omega_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}$
**end for**
$(\hat{\omega}_B^I)_k^- = \omega_{n+1}$
**return** $(\hat{\omega}_B^I)_k^-$

where $N_{gg}$, gravity gradient disturbance torque, $N_{aero}$, aerodynamic disturbance torque,

$N_{rw}$, reaction wheel disturbance torque, $N_{gyro}$, gyroscopic torque, $N_m$, magnetic control torque and $N_w$, reaction wheel control torque, are modelled according to [**?**].

The calculation for the $\hat{q}_k^-$ is done with Eq 3.13 [**?**]

$$\hat{q}_k^- = \left[ cos(k_q)\mathbf{I}_{4x4} + \frac{1}{\|(\hat{\omega}_B^O)_k^-\|} sin(k_q)\mathbf{\Omega}_k^- \right] \hat{q}_{k-1}^+$$

$$\text{where } k_q = \frac{Ts}{2}\|(\hat{\omega}_B^O)_k^-\|$$

$$\begin{aligned} (\hat{\omega}_B^O)_k^- &= (\hat{\omega}_B^I)_k^- - \hat{\mathbf{A}} \begin{bmatrix} 0 & -(\omega_k) & 0 \end{bmatrix}^T \\ &= \begin{bmatrix} \hat{\omega}_{ox} & \hat{\omega}_{oy} & \hat{\omega}_{oz} \end{bmatrix}^T \end{aligned} \tag{3.13}$$

$$\|(\hat{\omega}_B^O)_k^-\| = \sqrt{\hat{\omega}_{ox}^2 + \hat{\omega}_{oy}^2 + \hat{\omega}_{oz}^2}$$

$$\text{and } \mathbf{\Omega}_k^- = \begin{bmatrix} 0 & \hat{\omega}_{oz} & -\hat{\omega}_{oy} & \hat{\omega}_{ox} \\ -\hat{\omega}_{oz} & 0 & \hat{\omega}_{ox} & \hat{\omega}_{oy} \\ \hat{\omega}_{oy} & -\hat{\omega}_{ox} & 0 & \hat{\omega}_{oz} \\ -\hat{\omega}_{ox} & -\hat{\omega}_{oy} & -\hat{\omega}_{oz} & 0 \end{bmatrix}$$

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

$$\hat{x}_k^- = \begin{bmatrix} (\hat{\omega}_B^I)_k^- & \hat{q}_k^- \end{bmatrix} \tag{3.14}$$

$Q_k$ is the covariance matrix representing the discrete system noise and is assumed to be zero-mean and Guassian. $\Phi_k$ is the discrete system perturbation model. $H_k$ is the discrete measurement perturbation Jacobian Matrix. $\mathbf{R}_k$ is the measurement noise covariance matrix. The state covariance matrix $P_k$ can be propagated with Eq 3.15.

$$\mathbf{P}_k^- = \Phi_k \mathbf{P}_{k-1}^+ \Phi_k^T \tag{3.15}$$

The $\mathbf{e}_k$ calculated with Eq 3.16

$$\mathbf{e}_k = v_{meas,k} - \hat{\mathbf{A}}_k^- v_{model,k} \tag{3.16}$$

where $v_{meas,k}$ is the measured vector in **SBC** and $v_{model,k}$ is the modelled **ORC** vector. The gain matrix $\mathbf{K}_k$ is used to determine the the influence of $\mathbf{e}_k$ on updated state vector, $\hat{x}_k^+$. $\mathbf{K}_k$ can be calculated as

$$\mathbf{K}_k = \mathbf{P}_k^-(\mathbf{H}_k^-)^T \left[ \mathbf{H}_k^- \mathbf{P}_k^- (\mathbf{H}_k^-)^T + \mathbf{R}_k \right]^{-1} \tag{3.17}$$

after which the updated state vector can be calculated with Eq 3.18

$$\hat{x}_k^+ = \hat{x}_k^- + \mathbf{K}_k \mathbf{e}_k \tag{3.18}$$

The state covariance matric can then be updated as

$$\mathbf{P}_k^+ = \left[\mathbf{I}_{7\times7} - \mathbf{K}_k \mathbf{H}_k^+\right] \mathbf{P}_k \left[\mathbf{I}_{7\times7} - \mathbf{K}_k \mathbf{H}_k^+\right] + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \tag{3.19}$$

During the measurement update of $\hat{x}_k^-$ with the error of $v_{meas,k}$ and $v_{model,k}$, the $\hat{x}_k^+$ is largely affected by anomalous behaviour in the sensor measurements. The sensitivity of the Kalman filter to this behaviour as well as related work will be discussed in section **??**.

## 3.6. Attitude Control

The attitude command vector during nadir-pointing in the SBC frame is $\mathbf{u}_c = [0, 0, 1]$, since the SBC frame $z$ coordinate should line up with the ORC frame. During the sun following phase, the attitude command according to [**?**] can be calculated as

$$\mathbf{u}_c = \frac{\mathbf{u}_{sp}^{SBC} \times \mathbf{s}_o}{\left\| \mathbf{u}_{sp}^{SBC} \times \mathbf{s}_o \right\|} \tag{3.20}$$

Where $\mathbf{s}_o$ is the measured unit sun vector in ORC, and the main solar panel's position is denoted as a unit vector, $\mathbf{u}_{sp}^{SBC}$. The angle between $\mathbf{u}_{sp}^{SBC}$ and $\mathbf{s}_o$, $\delta$, can be calculated with the vector dot-product. The command quaternion $\mathbf{q}_c$ can then be calculated

$$\mathbf{q}_c = \begin{bmatrix} \mathbf{u}_c sin(\frac{\delta}{2}) \\ cos(\frac{\delta}{2}) \end{bmatrix} \tag{3.21}$$

This can then be used as the reference for the control. The reference $\boldsymbol{\omega}_b^I$ is always $[0, 0, 0]$. The

### 3.6.1. Momentum-Dumping

Momentum dumping is crucial to ensure that the wheel disturbance does not cause the system to become unstable.

### 3.6.2. Quaternion Feedback Controller

Reaction wheel control during normal operation

# Chapter 4

# Implementation of Methods on Actual Satellite

## 4.1. Fault Detection, Isolation and Recovery

## 4.2. Feature Extraction

## 4.3. Detection

## 4.4. Isolation

## 4.5. Recovery

## 4.6. Summary

# Chapter 5

# Anomalies

## 5.1. Reflection of Solar Panels on Sun Sensor

The reflection anomaly is modelled for the specific shape and design of the CubeSat as shown in Figure 5.1.

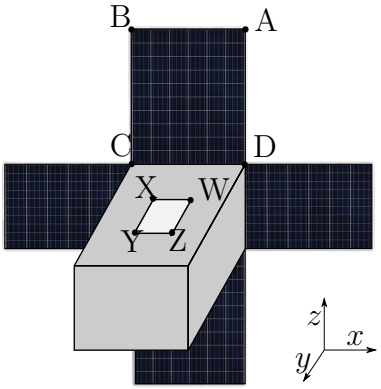



**Figure 5.1:** Cube Sat

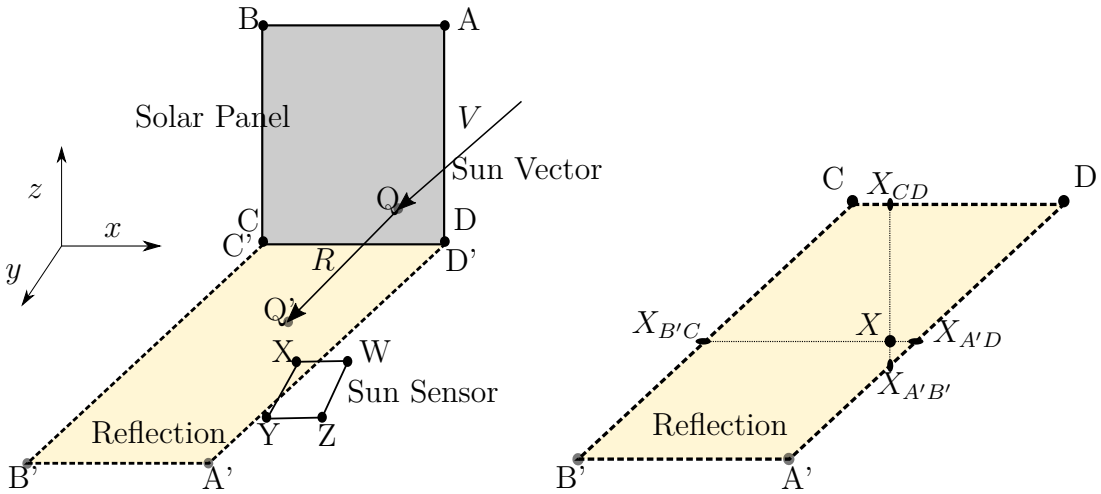



**Figure 5.2:** Reflection

The assumption is that the solar panel can be modelled as a geometric plane. Therefore, light from the solar panel will reflect as from a perfectly smooth mirror. It is further assumed 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. Therefore the intensity of the light vector is disregarded. The reflected sun vector, $r$, can be calculated as

$$\mathbf{r} = \mathbf{v} - 2\mathbf{n}^T(\mathbf{v} \cdot \mathbf{n}). \tag{5.1}$$

Where $\mathbf{v}$ is the incoming sun vector and $\mathbf{n}$ is the average vector to the plane $ABCD$ of the solar panel, as seen in Figure 5.1. To calculate the intersection of the reflected vector with the plane $XWYZ$ of the sun sensor, the equation of the plane, $XWYZ$, the reflected vector, $r$, and the point of origin is required. The reflection of the sun vector, $\mathbf{v}$ is illustrated in Figure 5.2 The equation for a plane can be denoted as

$$\mathbf{p} = ax + by + cz + d. \tag{5.2}$$

Where $x$, $y$ and $z$ are the dimensions in the SBC frame. The reflected unit vector can also be translated to

$$\begin{aligned} x &= \alpha t \\ y &= \beta t \\ z &= \zeta t \end{aligned} \tag{5.3}$$

Where the coefficients, $\alpha$, $\beta$ and $\zeta$ are the values of the reflected unit vector in each respective dimension. Since we can calculate the coefficients for Eq 5.3 from the reflected vector, we can calculate $t$, by substituting $x$, $y$ and $z$ into Eq 5.2. This is possible, because we determine the equation of the plane for the surface $XYZW$ based on our design.

After that, the intersecting point with the plane $XYZW$ can be calculated as

$$P(x, y, z) = (o_1 + \alpha t, o_2 + \beta t, o_3 + \zeta t) \tag{5.4}$$

where $o_1, o_2, o_3$ is the point of origin. Which in this case is the position of reflection from the solar panel. Therefore, if the sun vector $\mathbf{v}$ reflected from the solar panel as $\mathbf{r}$, the point of intersection $Q'$ on Figure 5.2 can be calculated as

$$Q'(x, y, z) = (Q_x + \alpha t, Q_y + \beta t, Q_z + \zeta t) \tag{5.5}$$

To model reflection from the solar panels to the sun sensor, only two corners of the solar panel and two corners of the sun sensor are to be considered. From Figure 5.2 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 $XYZW$ using Eq 5.4. 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 $XYWZ$ 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

$$y_{A'B'} = mx_{A'B'} + c \tag{5.6}$$

from the coordinates of $A'$ and $B'$, the corresponding $X_{A'B',y}$ can be calculated by substituting $X_x$ into Eq 5.6. 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 is shown in Figure 5.3. During the modelling of the reflection, the reflection also affects the estimation and, therefore, the attitude control of the satellite. In the figures of this article, the grey zones indicate the eclipse period, as seen in Figure 5.3.

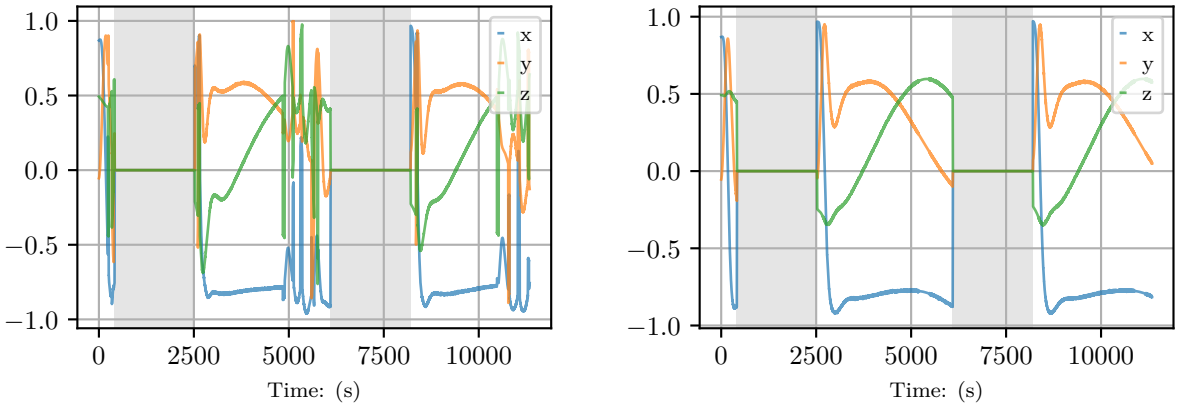



**Figure 5.3:** Comparison of Sun Vector with and without Reflection

## 5.2. Moon and Sun in Field of View of Nadir Sensor

An anomaly that can be experienced by an infra-red nadir sensor is the moon overlapping the horizon of the earth in die nadir sensor's FOV. This is shown in Figure 5.6. This influences the edge detection and circular fit algorithm [?,?] 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.

### 5.2.1. Simulating Nadir Sensor Infra-red Image

Firstly the vectors of both the satellite to earth and the moon the earth 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 already been done. These vectors are shown in Figure 5.4. 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

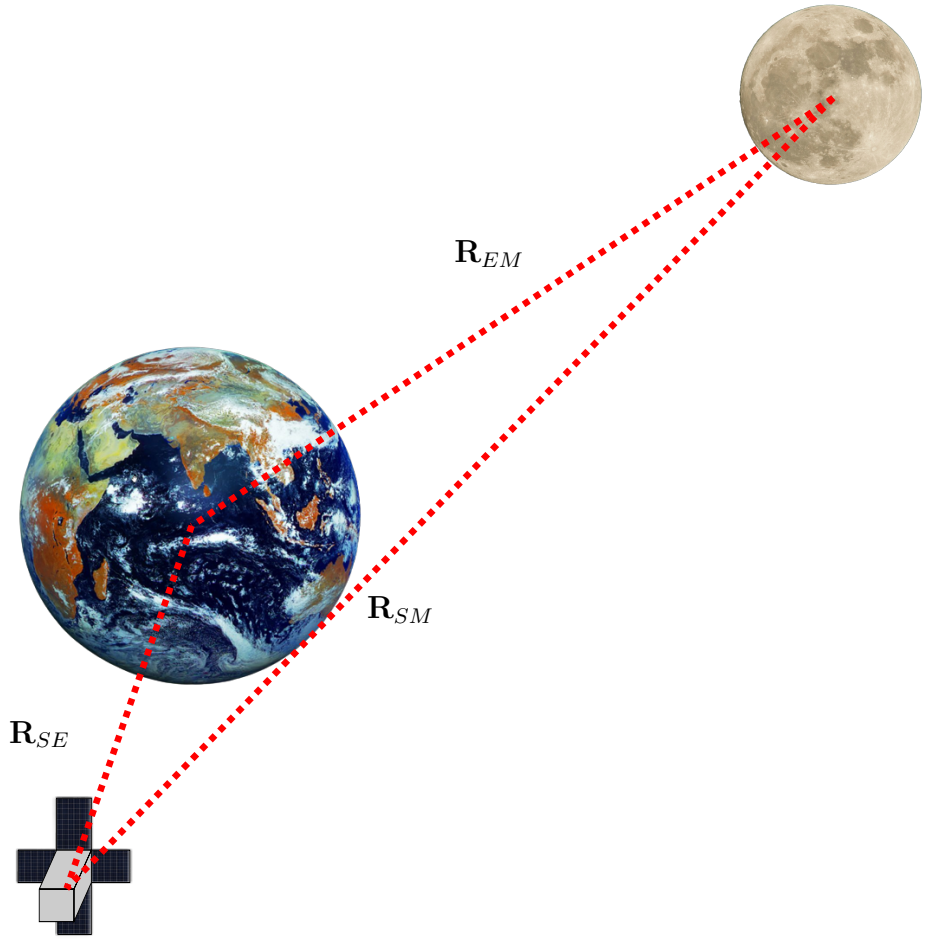



**Figure 5.4:** Earth to Moon and Earth to Satellite Vectors

$$P_{earth} = [x_0, y_0, z_0] \tag{5.7}$$

and

$$\mathbf{R}_{SE} = [n_x, n_y, n_z]. \tag{5.8}$$

Therefore with the equation for the 3D plane defined as

$$Ax + By + Cy = D \tag{5.9}$$

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

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} n_x \\ n_y \\ n_z \\ n_x x_0 + n_y y_0 + n_z z_0 \end{bmatrix} \tag{5.10}$$

This 3D plane slicing the earth in half as seen from the satellite is shown in Figure 5.5. The moon and earth can both be projected unto the 3D plane to determine the image seen by the nadir sensor. Therefore the nadir vector must also be projected unto the 3D plane. A circle can be drawn for the earth. moon and the nadir sensor FOV. The radius, of the moon as projected on the 3D plane can be calculated as

$$R_{moon} = \|R_{SE}\| \frac{r_{moon}}{\|R_{SM}\|} \tag{5.11}$$

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

$$R_{FOV} = \|R_{SE}\| \tan\theta \tag{5.12}$$

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

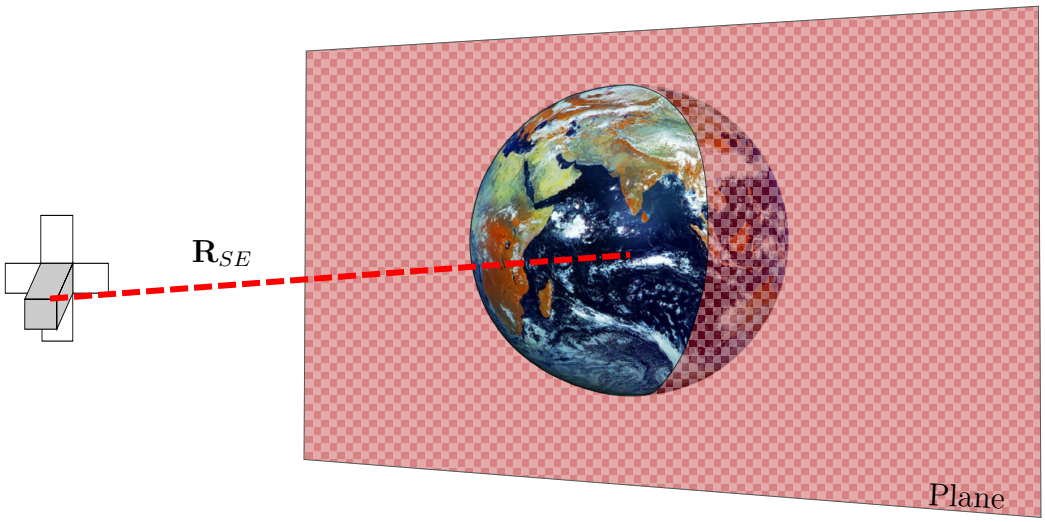



**Figure 5.5:** Plane perpendicular to $\mathbf{R}_{SE}$ and at center of earth

Firstly the edges of the moon and earth are discretely determined 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 projected earth and moon unto the 3D plane is shown in Figure 5.5. The discrete edges of the earth that is within the FOV and will be used for the algorithm discussed in section 5.2.2 are determined with the following logical statements.

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 is 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 the creates the array of points that will be used in the algorithm used to calculate the centre of the earth.

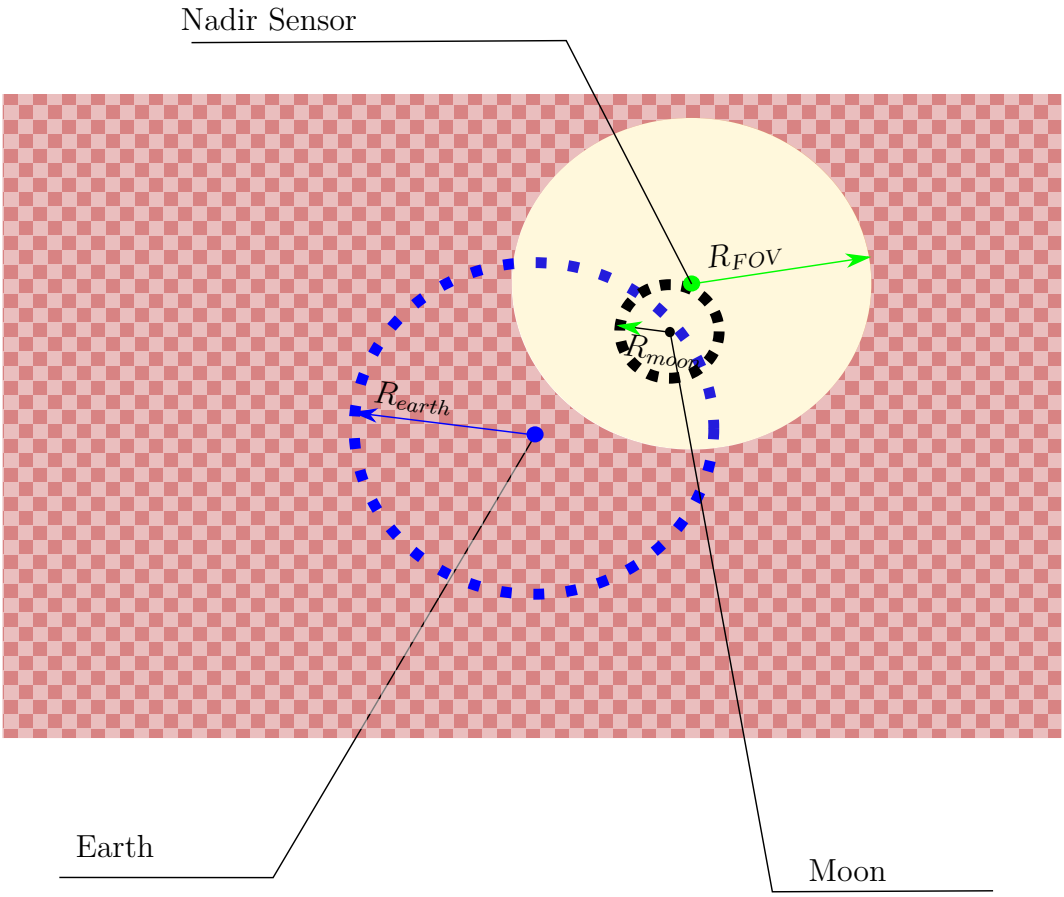



**Figure 5.6:** Projection of moon and earth on plane

## 5.2.2. Calculating the Centre of the Earth

The edges of the earth are detected based on a gradient between the lowest temperature and the highest temperature within the IR 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 it's lowest temperature being warmer than that of space.

With this assumption the circular fit algorithm as shown in Figure 5.7 can now be used to determine the centre of the earth on the plane [**?**]. For this calculation the 3D plane is transformed to a 2D plane and all the coordinates is also transformed.

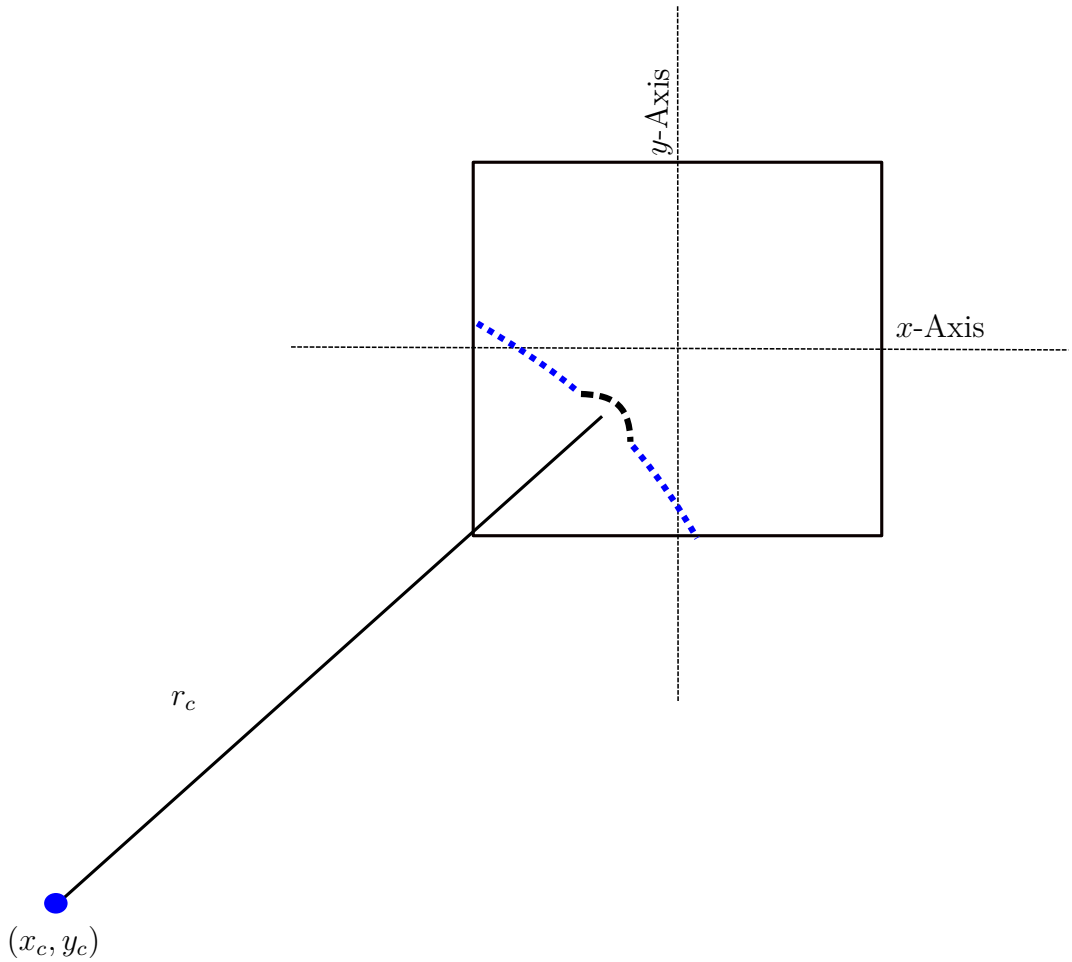



**Figure 5.7:** Circular Fit Algorithm

Firstly the curvature is described as

$$ax + bx + c = x^2 + y^2 \tag{5.13}$$

where

$$\begin{aligned} a &= 2x_c \\ b &= 2y_c \\ c &= r_c^2 - \sqrt{x_c^2 + y_c^2} \end{aligned} \tag{5.14}$$

Therefore using all the discrete edges within the image $a$, $b$ and $c$ can be calculated as

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x_0^2 + y_0^2 \\ x_1^2 + y_1^2 \\ \vdots \\ x_n^2 + y_n^2 \end{bmatrix} \tag{5.15}$$

where $(x_0, y_0)$ to $(x_n, y_n)$ are the coordinates of the discrete edges. 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. The other anomaly in this section 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 [**?**].

### 5.2.3. Influence of anomaly on estimation

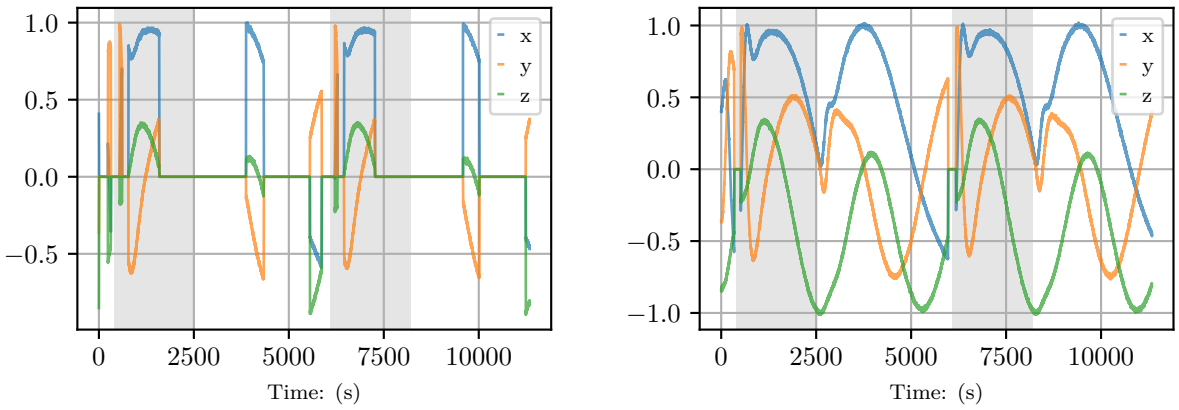



**Figure 5.8:** Comparison of Earth Vector with and without moon on the horizon

## 5.3. Magnetometer

[**?**]

### 5.3.1. Magnetic Moment Disturbance from Satellite Bus

$$m = AI \tag{5.16}$$

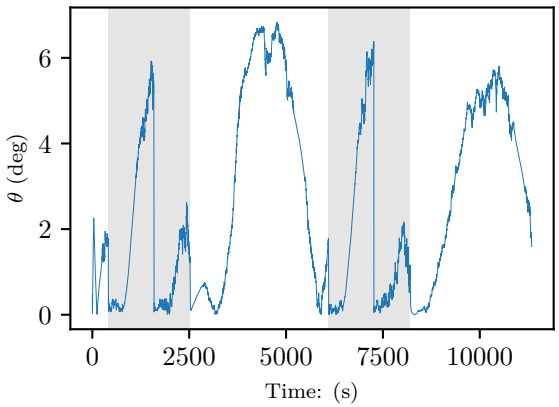



**Figure 5.9:** Earth Values

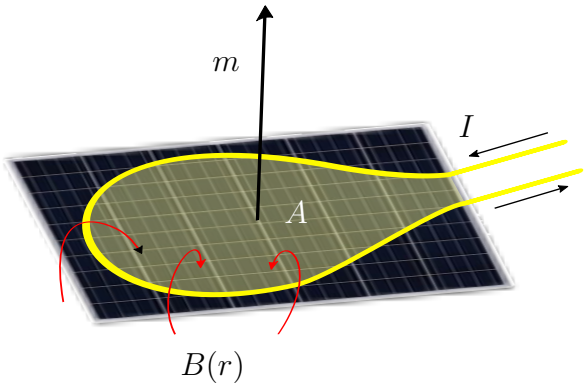



**Figure 5.10:** Dipole Moment from circular loop in solar panel

where $A$ is the area within the loop and $I$ is the current.

$$I = I_{max} cos(\theta) \tag{5.17}$$

where $\theta$ is the angle between the normal vector to the solar panel and the incoming sun vector.

The resulting torque can be expressed as

$$\tau = m \times B \tag{5.18}$$

where $B$ is the magnetic field of the earth.

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

$$B(r) = \frac{\mu_0}{4\pi} \frac{3\hat{r}(\hat{r} \cdot m) - m}{\|r\|^3} \tag{5.19}$$

The vector $r$ between the position of the magnetometer and the solar panel influences the magnetic field significantly. The magnetic field for each solar panel will therefore be different.

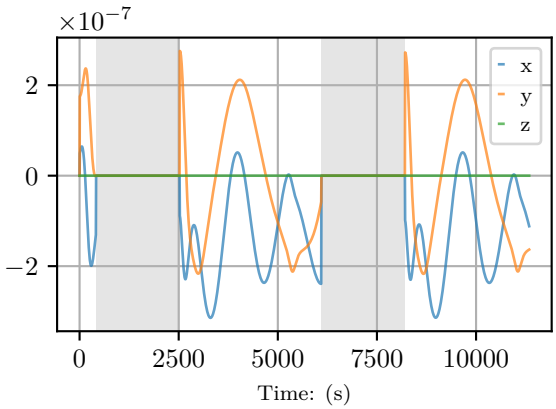



**Figure 5.11:** Solar Panel Disturbance Torques

The resulting measured vector by the magnetometer is the summation of the earth's magnetic field and the magnetic field produced by the coil in the solar panel.

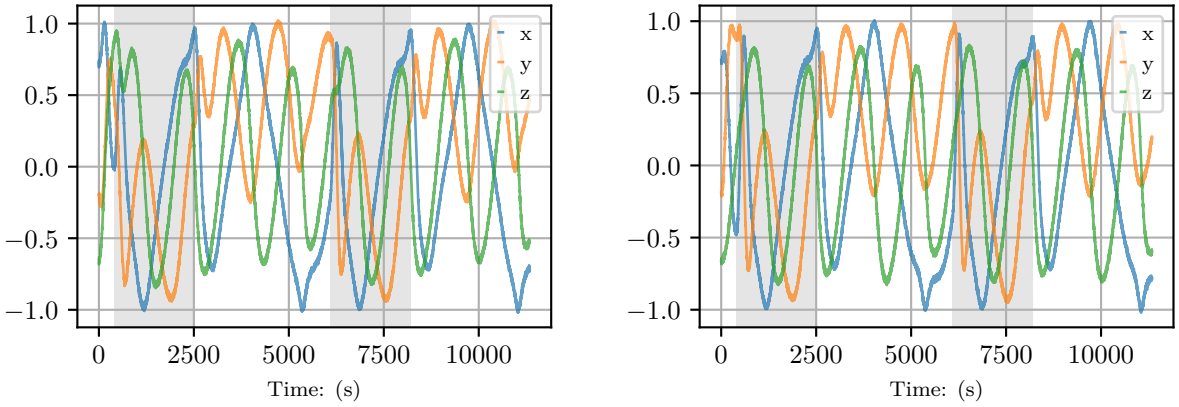



**Figure 5.12:** Comparison of Magnetic field Vector with and without solar Panel magnetic field

## 5.4. Reaction wheels

When an actuator fails it influences all the sensor measurements. A common anomaly is when reaction wheels failure. The anomaly will be modelled as a sudden failure in the actuator when it does not react to inputs.

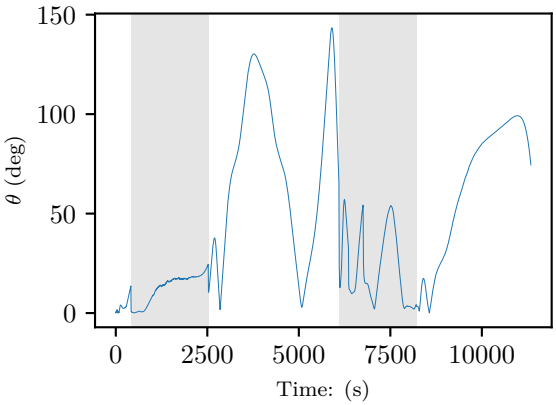



**Figure 5.13:** Estimation Metric with failure of Reaction Wheels

# Chapter 6

# Feature Extraction

## 6.1. Binary Feature Extraction

The classes hyperparameter will be set to 2. The only problem with this, is a very evident 2 classes is eclipse and non-eclipse. Consequently each feature extraction model will be trained twice. Once for the eclipse and once for the non-eclipse period. The features extracted will be used with the detection algorithms.

### 6.1.1. Linear Regression

### 6.1.2. Local Outlier Factor

### 6.1.3. K-means Clustering

### 6.1.4. Prony's Method

### 6.1.5. Partial Least Square

### 6.1.6. t-Distributed Stochastic Neighbor Embedding

## 6.2. Multiple Feature Extraction

The classes hyperparameter will be set to the number of anomalies (plus 1). The features will be used in the isolation algorithms. The only problem with this, is a very evident 2 classes is eclipse and non-ecplise. Consequently each feature extraction model will be trained twice. Once for the eclipse and once for the non-eclipse period.

### 6.2.1. Linear Regression

### 6.2.2. Local Outlier Factor

### 6.2.3. K-means Clustering

### 6.2.4. Prony's Method

### 6.2.5. Partial Least Square

### 6.2.6. t-Distributed Stochastic Neighbor Embedding

## 6.3. Summary

# Chapter 7

# Recovery

## 7.1. Analysis

## 7.2. EKF-ignore

## 7.3. EKF-combination

## 7.4. EKF-reset

## 7.5. EKF-top2

# Chapter 8

# Detection

## 8.1. Analysis

## 8.2. Supervised Learning

### 8.2.1. Random Forests

### 8.2.2. Decision Trees

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 with Eq 8.1.

$$GI = 1 - \sum_{i=1}^{n} (P_i)^2 \tag{8.1}$$

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 optimise the decision tree, which also considers the most prominent information gain to construct the decision tree. Figure 8.1 is a graphical representation of the decision tree developed to classify anomalies. 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 optimise the efficiency and accuracy of the decision tree.

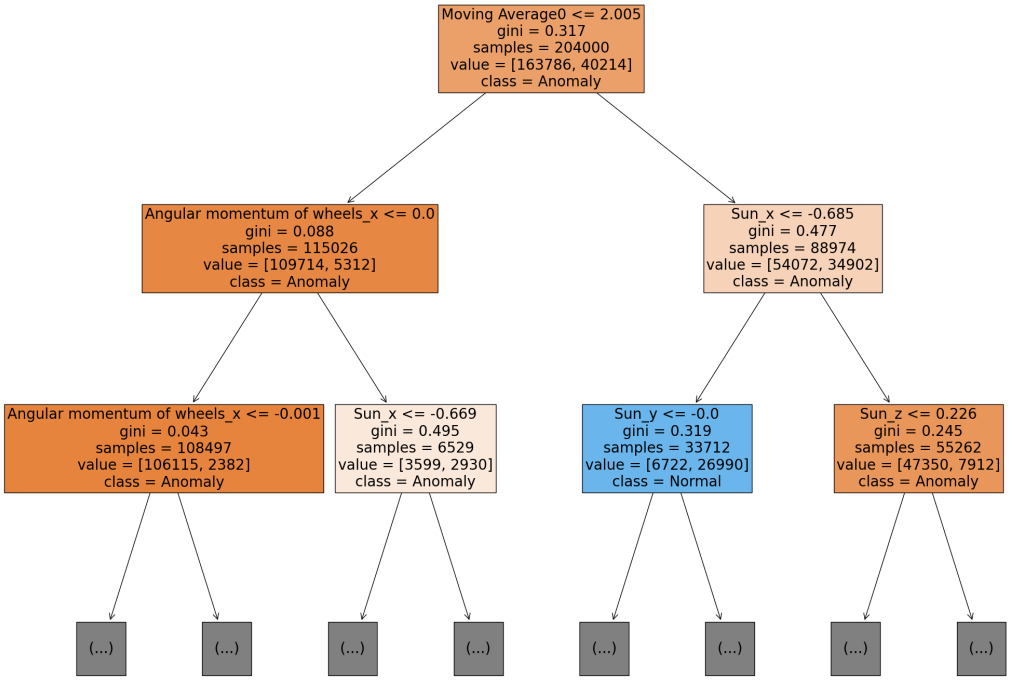



**Figure 8.1:** Decision Tree

### 8.2.3. Support Vector Machines

## 8.3. Unsupervised Learning

### 8.3.1. Isolation Forests

This unsupervised learning method is based on the principle of isolating data points by slicing the data with random conditions [**?**]. 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 [**?**]. 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 8.2 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 calculated the average distance for the data point. Using a threshold for the distance, the data point is classified as anomalous or not.

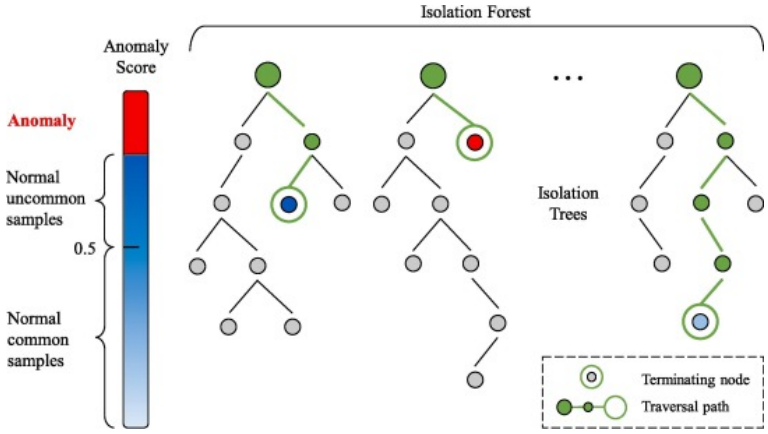



**Figure 8.2:** Isolation Forests [?]

The anomaly score is calculated with Eq 8.2

$$s(x, n) = 2^{-E(h(x))/c(n)} \tag{8.2}$$

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 [?] 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

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}. \tag{8.3}$$

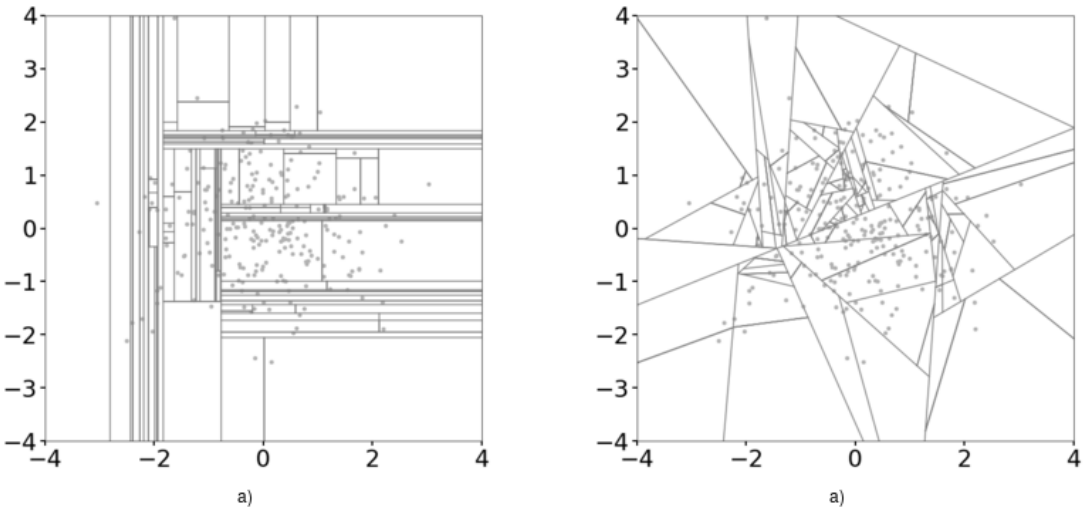
$H(i)$ in Eq 8.3 is the harmonic number and is estimated with Euler's constant as

$$H(i) \approx ln(i) + 0.5772156649. \tag{8.4}$$

Isolation Forests, however have multiple issues, since it splits data in rectangles as seen in Figure a. 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 [?] 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 b.

It is evident that applying an angle of 0° to all the slices the general algorithm of the




**(a)** Isolation Forest Slicing example **(b)** Extended Isolation Forest Slicing example

**Figure 8.3:** The slicing of Isolation Forest vs Extended Isolation Forest

extended isolation forest produces the standard isolation forest algorithm where planes or slices are perpendicular to the axis of the randomly selected feature, $x$.

### 8.3.2. Local Outlier Factor

Most algorithms for anomaly detection are based on a metric which accounts for the entire dataset [**?**]. However, many anomalies are identifiable in relation to the local neighbourhood of data points and not the overall dataset. Therefore, [**?**] developed the local outlier factor $LOF$ algorithm that provides a measure of a data point's "outlierness". 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 8.4 where the data points which are clustered together have smaller LOF's than data points which are removed from the highly dense areas.

To calculate the LOF, the $k$-distance must be calculated and also the local reachability density $lrd$. The $k$-distance, is the $k^{th}$ ranked $distance(o, p_i)$. Where $distance(o, p_i)$ is the distance between data point $o$ 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 $MinPts$. To reduce fluctuations in the $distance(o, p_i)$ the distance between $o$ and $p_i$ is replaced with

$$max\{distance(o, p_i), k\text{-distance}\} \tag{8.5}$$

and will henceforth be referred to as the reachability distance [**?**]. The $lrd$ of a data point,

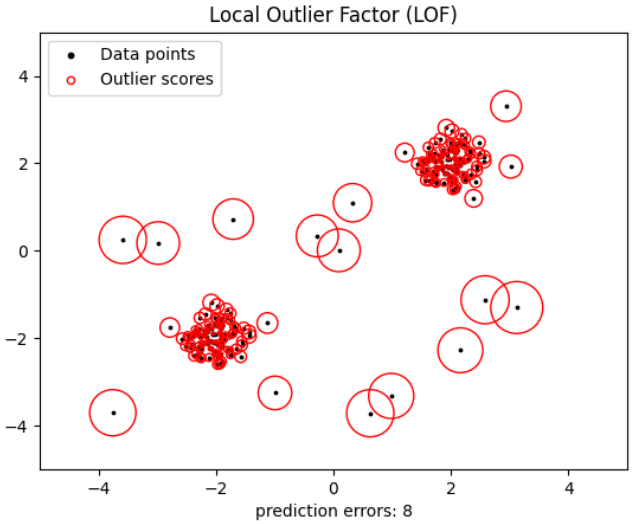



**Figure 8.4:** LOF measure

$p$, is calculated as

$$lrd_{MinPts}(p) = 1/\left(\frac{\sum_{o \in N_{MinPts}(p)} reachdist_{MinPts}(p,o)}{|N_{MinPts}(p)|}\right) \tag{8.6}$$

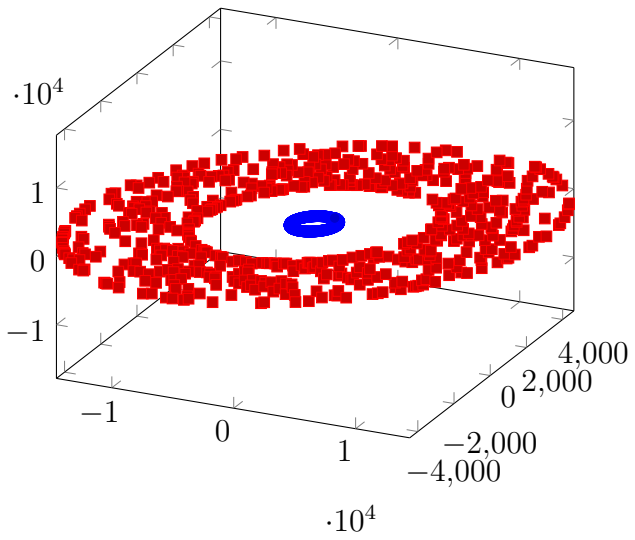
and denotes "the inverse of the average reachability distance based on the $MinPts$-nearest neighbours of the $p$" — [**?**]. Eq 8.6 enables the calculation for the $LOF$ of point $p$ as shown in Eq 8.7

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|} \tag{8.7}$$

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. 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 measurements of earth sensors, sun sensors and magnetometers that drastically change from the previous orbital data. For example in Fig **??** it is evident that the LOF will be comparatively larger for the red data points, which are anomalies, to the blue data points that are the normal orbit of the satellite.

## 8.4. Summary

Earth Sensor During Multiple Orbits



# Chapter 9

# Isolation

## 9.1. Analysis

## 9.2. Random Forests

## 9.3. Decision Trees

## 9.4. Support Vector Machines

## 9.5. Summary

# Chapter 10

# Results

## 10.1. Perfect Extended Kalman Filter

For all these results the best recovery method will be demonstrated

## 10.2. Unsupervised Detection and Supervised Isolation

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

## 10.3. Supervised Detection and Isolation

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

## 10.4. Summary

# Chapter 11

# 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.

## 11.1. 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.

## 11.2. 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.

## 11.3. 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.

## 11.4. 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.

# Appendix A

# Project Planning Schedule

This is an appendix.

# Appendix B

# Outcomes Compliance

This is another appendix.