begindocument/before
shipout/backgroundshipout/foreground

file/translations-basic-dictionary-english.trsl/after

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

Declaration ii

Abstract iii

List of Figures viii

List of Tables x

Nomenclature xi

1. Introduction 1

1.1. Background 1

1.2. Project definitions 2

1.3. Thesis outline 3

2. Literature Study 4

2.1. Anomaly Detection on Satellites 4

2.1.1. Analysis and Prediction of Satellite Anomalies 4

2.1.2. Agent-based algorithm for fault detection and recovery of gyroscope's drift in small satellite missions 4

2.2. Statistical Methods 5

2.2.1. Pearson Correlation 5

2.2.2. Variance 5

2.2.3. Kalman-Filter 6

2.2.4. Multivariate Guassian Distribution 6

2.2.5. Kullback-Leibler Divergence 7

2.2.6. Canonical Correlation Analysis 7

2.3. Feature Extraction 8

2.3.1. Prony's Method 8

2.3.2. Convolutional Networks 8

2.3.3. K-means Clustering 8

2.3.4. Partial Least Square 8

2.3.5. Locally Linear Embedding 8

2.3.6. t-Distributed Stochastic Neighbor Embedding 8

2.4. Supervised Learning 8

2.4.1. Decision Trees . . . 8
2.4.2. Random Forests . . . 9
2.4.3. Long Short Term Memory . . . 9
2.4.4. Support Vector Machines . . . 9
2.4.5. Naive Bayes . . . 9
2.4.6. K-nearest neighbours . . . 9
2.4.7. Artificial Neural Networks . . . 9
2.5. Unsupervised Learning . . . 9
2.5.1. Kernel Adaptive Density-based . . . 9
2.5.2. Loda . . . 9
2.5.3. Robust-kernel Density Estimation . . . 9
2.6. Reinforcement Learning . . . 10
2.7. Summary . . . 10

3. Simulation **11**
3.1. Attitude Determination and Control System . . . 11
3.1.1. Coordinate Frames . . . 11
3.1.2. Attitude . . . 13
3.1.3. Satellite Kinematics and Dynamics . . . 14
3.2. Environment . . . 15
3.2.1. Orbit Propagation . . . 15
3.2.2. Sun . . . 16
3.2.3. Geomagnetic field . . . 16
3.3. Sensor models . . . 16
3.3.1. Sun Sensor . . . 17
3.3.2. Nadir Sensor . . . 17
3.3.3. Magnetometer . . . 17
3.4. Disturbance models . . . 17
3.4.1. Gravity Gradient . . . 17
3.4.2. Aerodynamic Disturbance . . . 18
3.4.3. Wheel Imbalance . . . 18
3.5. Attitude Determination . . . 18
3.5.1. Extended Kalman Filter . . . 19
3.6. Attitude Control . . . 21
3.6.1. Quaternion Feedback Controller . . . 21
3.6.2. Momentum Dumping . . . 22

4. Implementation of Methods on Actual Satellite **24**
4.1. Fault Detection, Isolation and Recovery . . . 24
4.2. Feature Extraction . . . 24

4.3. Detection . . . 24
4.4. Isolation . . . 24
4.5. Recovery . . . 24
4.6. Summary . . . 24

5. Anomalies **25**
5.1. Reflection of Solar Panels on Sun Sensor . . . 25
5.1.1. Influence of anomaly on estimation . . . 27
5.2. Moon and Sun in Field of View of Nadir Sensor . . . 28
5.2.1. Simulating Nadir Sensor Infra-red Image . . . 28
5.2.2. Calculating the Centre of the Earth . . . 31
5.2.3. Influence of anomaly on estimation . . . 33
5.3. Magnetic Moment Disturbance from Satellite Bus . . . 33
5.3.1. Influence of anomaly on estimation . . . 36
5.4. Reaction wheels . . . 37
5.5. Summary . . . 37

6. Feature Extraction **41**
6.1. Binary Feature Extraction . . . 41
6.1.1. Local Outlier Factor . . . 41
6.1.2. Dynamic Mode Decomposition . . . 41
6.2. Multiple Feature Extraction . . . 42
6.2.1. Linear Regression . . . 42
6.2.2. Local Outlier Factor . . . 42
6.2.3. K-means Clustering . . . 42
6.2.4. Prony's Method . . . 42
6.2.5. Partial Least Square . . . 42
6.2.6. t-Distributed Stochastic Neighbor Embedding . . . 42
6.3. Summary . . . 42

7. Recovery **43**
7.1. Analysis . . . 43
7.2. EKF-ignore . . . 43
7.3. EKF-combination . . . 43
7.4. EKF-reset . . . 43
7.5. EKF-top2 . . . 43

8. Detection **44**
8.1. Analysis . . . 44
8.2. Supervised Learning . . . 44

8.2.1. Random Forests 44
8.2.2. Decision Trees 44
8.2.3. Support Vector Machines 45
8.3. Unsupervised Learning 45
8.3.1. Isolation Forests 45
8.3.2. Local Outlier Factor 47
8.4. Summary 48

9. Isolation **50**
9.1. Analysis 50
9.2. Random Forests 50
9.3. Decision Trees 50
9.4. Support Vector Machines 50
9.5. Summary 50

10. Results **51**
10.1. Perfect Extended Kalman Filter 51
10.2. Unsupervised Detection and Supervised Isolation 51
10.3. Supervised Detection and Isolation 51
10.4. Summary 51

11. Conclusion **52**
11.1. Project/thesis/dissertation summary 52
11.2. Appraisal of project/thesis/dissertation contributions 53
11.3. Suggestions for future work 53
11.4. What the student has learnt during this project 53

Bibliography **54**

A. Project Planning Schedule **55**

B. Outcomes Compliance **56**

List of Figures

1.1. System Diagram 2

2.1. Guassian Distributions 7

3.1. Euler angles 13

5.1. Cube Sat 25
5.2. Reflection 26
5.3. Comparison of Sun Vector with and without Reflection 28
5.4. Estimation Metric with reflection on sun sensor 28
5.5. Earth to Moon and Earth to Satellite Vectors 29
5.6. Plane perpendicular to $\mathbf{R}_{SE}$ and at center of earth 30
5.7. Projection of moon and earth on plane 31
5.8. Circular Fit Algorithm 32
5.9. Comparison of Earth Vector with and without moon on the horizon 33
5.10. Earth Values 34
5.11. Dipole Moment from circular loop in solar panel 34
5.12. Shadow created by CubeSat body on Solar Panels 35
5.13. Solar Panel Disturbance Torques 35
5.14. Comparison of Magnetic field Vector with and without solar Panel magnetic field 36
5.15. Estimation Metric with induced dipole moment 36
5.16. Estimation Metric with failure of Reaction Wheels 37
5.17. Estimation Metric with failure of Reaction Wheels 38
5.18. Estimation Metric with failure of Reaction Wheels 38
5.19. Estimation Metric with failure of Reaction Wheels 38
5.20. Estimation Metric with failure of Reaction Wheels 39
5.21. Estimation Metric with failure of Reaction Wheels 39
5.22. Estimation Metric with failure of Reaction Wheels 39
5.23. Estimation Metric with failure of Reaction Wheels 40
5.24. Estimation Metric with failure of Reaction Wheels 40

8.1. Decision Tree 45
8.2. Isolation Forest 46
8.3. Slicing of Isolation Forest 47

8.4. Local outlier measure 48

List of Tables

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 ith 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

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

1.1. Background

For many satellite missions, attitude determination is of high importance. It is required to effectively control the attitude to fulfill the required mission requirements. The control performance is also limited by the estimation accuracy and performance. A classical satellite attitude requirement is to be earth-following during eclipse, point the payload to a target and otherwise point and track the sun for solar charging. Good attitude estimation throughout the entire orbit is required 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 Extended Kalman Filter (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 be either just reduced estimation accuracy or even divergence and instability. It is good practice to develop appropriate tests to protect the EKF against incorrect measurements.

Data driven vs knowledge driven.

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. 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 detect, classify (isolate) and recover from the anomaly autonomously to ensure safe operation during orbit. A model for the fault detection, isolation and recovery (FDIR) of the EKF is provided in figure 1.1.

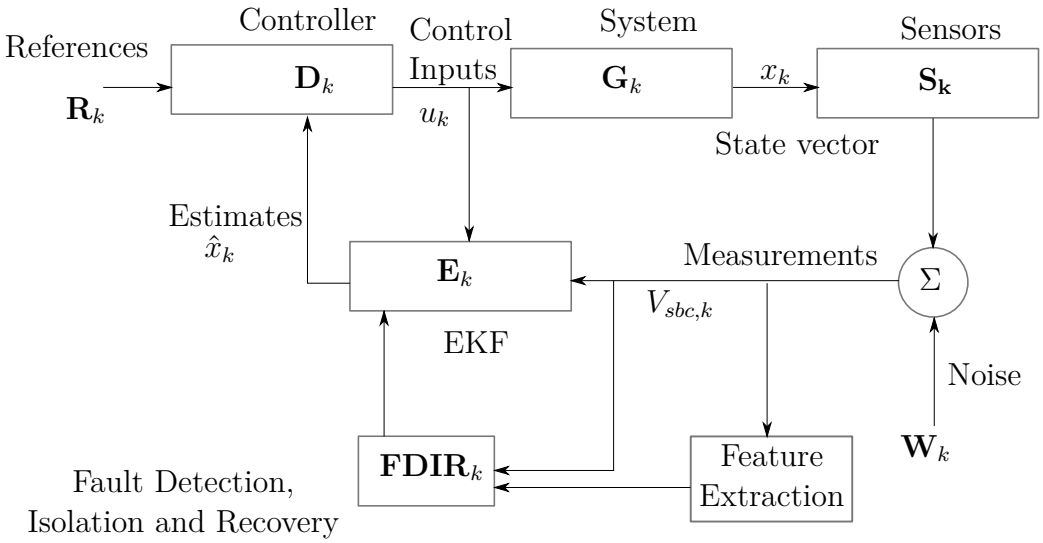


Figure 1.1: System Diagram

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 horizon and a magnetometer with magnetic disturbances influenced 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 7.

1.2. Project definitions

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 data 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 model accuracy depending on training data from generic sensor anomalies vs practical anomalies should also be discussed. A thorough analysis of the methods and the results should be used to discuss the difference in estimation accuracy based on the model's training on generic or specific anomalies.

1.3. 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 the subject of FDIR.
Chapter 3 demonstrated the development and implementation of the simulation environment.
Chapter 4
Chapter 5 provides the mathematical models of the specific anomalies and the effects thereof on the satellite.
Chapter 6 describes the feature extraction methods used in this thesis to enhance the accuracy of the detection and classification models.
Chapter 7 provides various recovery methods and demonstrates the theoretical possibility of the methods based on perfect prediction accuracy.
Chapter 8 describes the different algorithms and methods used to detect an anomaly in the system.
Chapter 9 describes the different algorithms and methods used to classify an anomaly in the system.
Chapter 10 provides a summary of all the results for the combination of best methods as provided in chapter 7, chapter 8 and chapter 9.
Chapter 11 discusses the influence of modelling specific anomalies on the prediction accuracy and robustness of a EKF.

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.

Multivariate Anomaly Detection in Discrete and Continuous Telemetry Signals Using a Sparse Decomposition in a Dictionary [?]

Fault isolation of reaction wheels onboard three-axis controlled in-orbit satellite using ensemble machine learning [?]

Fault tolerant control for satellites with four reaction wheels [?]

Innovative Fault Detection, Isolation and Recovery Strategies On-Board Spacecraft: State of the Art and Research Challenges [?]

Machine learning methods for spacecraft telemetry mining [?]

Machine learning techniques for satellite fault diagnosis [?]

Satellite fault diagnosis using a bank of interacting Kalman filters [?]

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.

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.

Sensor fault detection and recovery in satellite attitude control [?]

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 [?],

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

LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection [?]

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, μ, and variance, σ^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

```
Determine feature vectors x_i
Determine threshold probabilty, ϵ
Calculate μ_j with Eq 2.4
Calculate σ_j with Eq 2.5
Calculate p(x) with Eq 2.6
if p(x) < ϵ then
    Anomaly = True
else
    Anomaly = False
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.

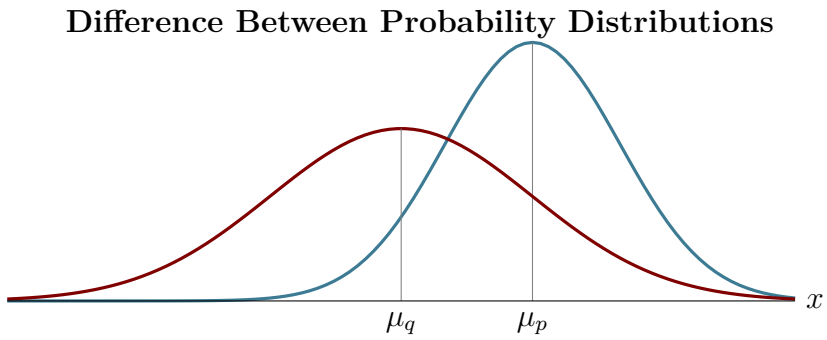


Figure 2.1: Guassian Distributions

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

2.4. 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

$$G = \sum_{k=1}^{K} \hat{p}_k (1 - \hat{p}_k) \tag{2.10}$$

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 during eclipse and point the solar panels towards the sun during the sunlit phase.

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 ($\mathbf{r}$) within the current coordinate frame can thus be expressed as

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

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

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

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, Υ, 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), α_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 $t = t_0$, α_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 $\mathbf{r}_{EFC}$ can be transformed to $\mathbf{r}_{EIC}$ with

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

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, $\mathbf{r}_{sat}$ and the unit velocity vector, $\mathbf{v}_{sat}$ in EIC [**?**].

$$
\begin{aligned}
\boldsymbol{A}_{EIC}^{ORC} &= \begin{bmatrix}\mathbf{u} & \mathbf{v} & \mathbf{w}\end{bmatrix}^T \\
\text{where} \quad \mathbf{w} &= -\frac{r_{sat}}{||r_{sat}||} \\
\mathbf{v} &= -\frac{r_{sat} \times v_{sat}}{||r_{sat} \times v_{sat}||} \\
\mathbf{u} &= \hat{v} \times \hat{w}
\end{aligned}
\tag{3.6}
$$

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. 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 **??**. **TODO: The figures must be in theta, phi and psi**.

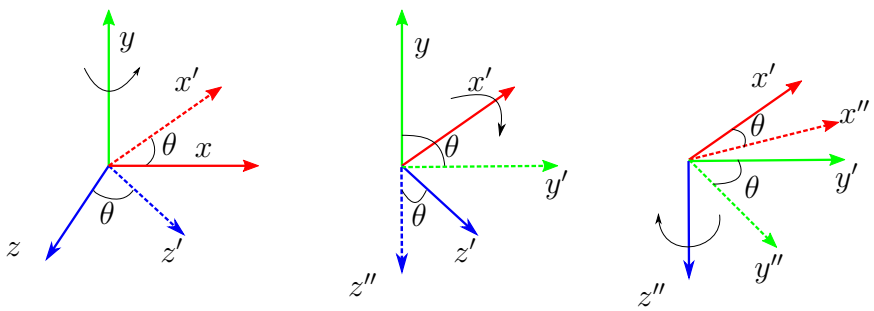


Figure 3.1: Euler angles

The $\boldsymbol{A}_{ORC}^{SBC}$ can be used to calculate the attitude transformation from given Euler angle rotations. This is done by multiplying representing each individual Euler angle rotation as a transformation matrix. These matrices form the $\boldsymbol{A}_{ORC}^{SBC}$ as given by

$$
\begin{aligned}
\mathbf{A} &= \mathbf{A}_\psi \mathbf{A}_\phi \mathbf{A}_\theta \\
&= \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix}
\end{aligned}
\tag{3.7}
$$

Euler angles however is not always a suitable method in determining the attitude of a satellite. This is because of singularities that can occur such as the gimbal-lock effect. Where two rotational axis, coincide to form a single rotational axis. Consequently, not all $3D$ rotations can then be described with the Euler angles, since effectively only two rotations can occur in stead of three [**?**]. Therefore the method of describing 3D rotation with quaternions is more often used and more convenient.

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

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \tag{3.8}$$

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 $\mathbf{e} = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^T$ and the angle of the Euler rotation is Φ then $\mathbf{q}$ can be expressed as

$$\mathbf{q} = \begin{bmatrix} e_1 sin(\frac{\Phi}{2}) \\ e_2 sin(\frac{\Phi}{2}) \\ e_3 sin(\frac{\Phi}{2}) \\ cos(\frac{\Phi}{2}) \end{bmatrix} \tag{3.9}$$

It is difficult to visualize a quaternion, however the most simplistic method of understanding it is shown in Figure **TODO: Provide figure of quaternion**. A quaternion is consequently a unit vector protruding from the centre point of an object as well as the angle of rotation of that object around that unit vector. As seen in Figure **??** the angle θ is the angle of rotation around the y-axis, for quaternions the angle of rotation is the same principle, however the axis around which the object is rotating, is the unit vector. Therefore, q_4 provides the angle of rotation while, q_1, q_2, q_3 represents the unit vector, however with the condition of Eq 3.8.

$\boldsymbol{A}_{ORC}^{SBC}$ can also be transformed as a function of $\mathbf{q}$ [**?**] through

$$\mathbf{A} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}. \tag{3.10}$$

Since the quaternion is used for attitude determination it will also be used for the attitude control. Therefore, a error between the commanded quaternion, $\mathbf{q}_c$ and the current quaternion, $\mathbf{q}$, must be calculated. This is discussed in section 3.6.1.

3.1.3. Satellite Kinematics and Dynamics

The conservation of momentum dominates the dynamics of a satellite. This is dominated by 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, $\dot{\boldsymbol{\omega}}_B^I$. The control torques used in this design are only reaction wheel torques, $\mathbf{N}_w$ and magnetic torques, $\mathbf{N}_m$. The disturbance torques are discussed in detail in section 3.4, therefore it can 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

$$\begin{aligned} \mathbf{I}\dot{\boldsymbol{\omega}}_B^I &= \mathbf{N_c} + \mathbf{N_d}, \\ \text{where} \quad \mathbf{N_d} &\approx \mathbf{N}_{aero} - \mathbf{N}_{gyro} + \mathbf{N}_{gg} + \mathbf{N}_{rw}, \\ \text{and} \quad \mathbf{N_c} &= \mathbf{N}_m - \mathbf{N}_w. \end{aligned} \tag{3.11}$$

This is the overarching equation that will be used to determine the control torque as well as the model update of the extended Kalman Filter (EKF). The integration method used in the simulation is the 4th order Rungka-Kutta method to solve the differential equations. This is demonstrated with algorithm 3.2.

Algorithm 3.2: Multi-variate Guassian Distribution Algorithm

```
Definitions: Ts - Timestamp;
h = Ts/N
for n ← 1 to N do
    k1 = hf(xn, yn)
    k2 = hf(xn + h/2, yn + k1/2)
    k3 = hf(xn + h/2, yn + k2/2)
    k4 = hf(xn + h, yn + k3)
    yn+1 = yn + k1/6 + k2/3 + k3/3 + k4/6
end for
```

where h is the step size, which is set to $T_s/10$, where the time step, T_s, is equal to one second. With this procedure the dynamics and kinematics of the satellite can be simulated after each time step.

3.2. Environment

3.2.1. 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 of the simplified general perturbations (SGP) model, namely SGP4, is used as orbit propagator of the satellite after each time step [**?**].

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, which 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 elapse 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 by the SGP4 model.

TODO: Show plot of 3D earth orbit...

The SGP4 outputs the $\mathbf{r}_{sat_k}$ and $\mathbf{v}_{sat_k}$ in the EIC reference frame. Therefore, $\boldsymbol{A}_{EIC}^{ORC}$ can now be calculated according to Eq-3.6.

3.2.2. Sun

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

$$
\begin{aligned}
\mathbf{r}_{sun} &= r_{\oplus} \begin{bmatrix} cos(\lambda_e) \\ cos(\epsilon)sin(\lambda_e) \\ sin(\epsilon)sin(\lambda_e) \end{bmatrix}, \\
\text{where} \quad r_{\oplus} &= 1.000\,140\,612 - 0.016\,708\,617\,cos(M_{\oplus}) - 0.001\,395\,89\,cos(2M_{\oplus}), \\
M_{\oplus} &= 357.527\,723\,300^o + 35\,999.050\,340\,T_{JC}, \\
\lambda_e &= \lambda_{M_{\oplus}} + 1.914\,666\,471\,sin(M_{\oplus}) + 0.019\,994\,643\,sin(2M_{\oplus}), \\
\lambda_{M_{\oplus}} &= 280.460\,618\,400^o + 36\,000.770\,053\,610\,T_{JC}, \\
\epsilon &= 23.439\,291^o - 0.013\,004\,200\,T_{JC} \\
\text{and} \quad T_{JC} &= \frac{J_t - 2\,451\,545}{36\,525}.
\end{aligned}
\tag{3.12}
$$

$$
S_{EIC} = \mathbf{r}_{sun} - \mathbf{r}_{sat} \tag{3.13}
$$

TODO: Show drawing of sun relative to earth and satellite...

TODO: Show graph of sun plot...

3.2.3. Geomagnetic field

$$
V(r_s, \theta, \lambda) = R_E \sum_{n=1}^{k} \left(\frac{R_E}{r_s}^{n+1} \right) \sum_{m=0}^{n} \left(g_n^m cos(m\lambda) + h_n^m sin(m\lambda) \right) P_n^m(\theta) \tag{3.14}
$$

TODO: Show graph of geomagnetic plot...

3.3. Sensor models

Sun sensor, nadir, magnetometer.

$$\mathbf{v}_b = \mathbf{A}_{ORC}^{EIC}\mathbf{v}_{ORC} + \mathbf{m}_v \tag{3.15}$$

3.3.1. Sun Sensor

TODO: Show position of sun sensor...

3.3.2. Nadir Sensor

TODO: Show position of nadir sensor...

3.3.3. Magnetometer

TODO: Show position of magnetometer...

3.4. Disturbance models

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

$$\mathbf{N}_{gyro} = \dot{\boldsymbol{\omega}}_B^I \times (\mathbf{I}\dot{\boldsymbol{\omega}}_B^I + \mathbf{h}_w), \tag{3.16}$$

where h_w is the angular momentum of the reaction wheels. The other disturbance torques are discussed in more detail below.

3.4.1. Gravity Gradient

$$\boldsymbol{N}_{gg} = 3\,\omega_o^2(\mathbf{z}_o^B \times \mathbf{I}\mathbf{z}_o^B) \tag{3.17}$$

$$\mathbf{z}_o^B = \mathbf{A}\begin{bmatrix}0 & 0 & 1\end{bmatrix}^T \tag{3.18}$$

$$\begin{aligned}
\mathbf{N}_{gg} &= \begin{bmatrix} k_{gx}A_{23}A_{33} \\ k_{gy}A_{13}A_{33} \\ k_{gz}A_{13}A_{23} \end{bmatrix} \\
\text{where} \quad k_{gx} &= 3\,\omega_o^2(\mathbf{I}_{zz} - \mathbf{I}_{yy}), \\
k_{gy} &= 3\,\omega_o^2(\mathbf{I}_{xx} - \mathbf{I}_{zz}), \\
\text{and} \quad k_{gz} &= 3\,\omega_o^2(\mathbf{I}_{yy} - \mathbf{I}_{xx}),
\end{aligned} \tag{3.19}$$

3.4.2. Aerodynamic Disturbance

[?]

$$\mathbf{N}_{aero} = \sum_{i=1}^{n} \Big(\rho \left\| \mathbf{v}_A^B \right\|^2 A_i H\{cos(\alpha_i)\} \, cos(\alpha_i) \Big(\sigma_t (\mathbf{r}_{pi} \times \bar{\mathbf{v}}_A^B) + \Big[\sigma_n S + (2 - \sigma_n - \sigma_t) cos(\alpha_i) \Big] (\mathbf{r}_p i \times \bar{\mathbf{n}}_i) \Big) \Big) \tag{3.20}$$

$$\rho = \rho_o e^{-\frac{h-h_o}{H}} \tag{3.21}$$

$$\mathbf{v}_A^{EIC} = \begin{bmatrix} 0 \\ 0 \\ \omega_E \end{bmatrix} \times \mathbf{r}_{sat} - \mathbf{v}_{sat} \tag{3.22}$$

TODO: Show drawing of aerodynamic disturbance...

3.4.3. Wheel Imbalance

$$\mathbf{F}_{xs} = U_s \omega^2 \begin{bmatrix} sin(\omega t + \phi_s) \\ cos(\omega t + \phi_s) \\ 0 \end{bmatrix} \tag{3.23}$$

$$\mathbf{N}_{xs} = w_z + F_{xs} \tag{3.24}$$

$$F_{xd} = mr\omega^2 \tag{3.25}$$

$$\mathbf{N}_{xs} = U_d \omega^2 \begin{bmatrix} sin(\omega t + \phi_d) \\ cos(\omega t + \phi_d) \\ 0 \end{bmatrix}, \tag{3.26}$$

where $U_d = mrd$ is the dynamic imbalance. **TODO: Get values of Us and Ud used in simulation**

TODO: Show drawing of static and dynamic 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

$$\dot{x}_t = \mathbf{f}(x_t) + s_t \tag{3.27}$$

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, ω_B^I.

$$x = [q, \omega_B^I]^T \tag{3.28}$$

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 ω_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 $N_d = N_{gg} - N_{gyro}$
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 3.29 [**?**]

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

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

$$\begin{aligned}(\hat{\omega}_B^O)_k^- &= (\hat{\omega}_B^I)_k^- - \hat{\mathbf{A}}\begin{bmatrix}0 & -(\omega_k) & 0\end{bmatrix}^T \\ &= \begin{bmatrix}\hat{\omega}_{ox} & \hat{\omega}_{oy} & \hat{\omega}_{oz}\end{bmatrix}^T\end{aligned} \tag{3.29}$$

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

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

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

$$\hat{x}_k^- = \begin{bmatrix}(\hat{\omega}_B^I)_k^- & \hat{q}_k^-\end{bmatrix} \tag{3.30}$$

Q_k is the covariance matrix representing the discrete system noise and is assumed to be zero-mean and Guassian. Φ_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 3.31.

$$\mathbf{P}_k^- = \Phi_k \mathbf{P}_{k-1}^+ \Phi_k^T \tag{3.31}$$

The $\mathbf{e}_k$ calculated with Eq 3.32

$$\mathbf{e}_k = v_{meas,k} - \hat{\mathbf{A}}_k^- v_{model,k} \tag{3.32}$$

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

$$\mathbf{K}_k = \mathbf{P}_k^-(\mathbf{H}_k^-)^T\left[\mathbf{H}_k^-\mathbf{P}_k^-(\mathbf{H}_k^-)^T + \mathbf{R}_k\right]^{-1} \tag{3.33}$$

after which the updated state vector can be calculated with Eq 3.34

$$\hat{x}_k^+ = \hat{x}_k^- + \mathbf{K}_k \mathbf{e}_k \tag{3.34}$$

The state covariance matrix can then be updated as

$$\mathbf{P}_k^+ = \left[\mathbf{I}_{7\times7} - \mathbf{K}_k\mathbf{H}_k^+\right]\mathbf{P}_k\left[\mathbf{I}_{7\times7} - \mathbf{K}_k\mathbf{H}_k^+\right] + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{3.35}$$

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 **??**.

TODO: Produce graphs for the estimation (Estimation Metric) with the eclipse phases.

3.6. 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 control 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.

3.6.1. Quaternion Feedback Controller

To ensure that the satellite is in the desired orientation the reaction wheels are used. To ensure stable control in all three axis [**?**] the quaternion feedback reaction wheel controller is implemented. The controller is designed from the estimated orientation and angular velocity, while ignoring disturbance torques. Eq **??** without the disturbance torques becomes

$$\mathbf{J}\dot{\boldsymbol{\omega}}_B^I = -\mathbf{N}_w - \mathbf{N}_{gyro} \tag{3.36}$$

To calculate the required torque, $\mathbf{N}_{w,req}$, the definition according to [**?**] for all cases at time step, k, is given as

$$\mathbf{N}_w(k) = K_{PI}\mathbf{J}\mathbf{q}_{err}(k) + K_{DI}\mathbf{I}\hat{\boldsymbol{\omega}}_B^O - \hat{\boldsymbol{\omega}}_B^I(k) \times [\mathbf{J}\hat{\boldsymbol{\omega}}_B^I(k) + \mathbf{h}_w(k)] \tag{3.37}$$

where $\mathbf{h}_w(k)$ is the measured angular momentum of the wheels, the control gains are defined as

$$\begin{aligned} K_{PI} &= 2\omega_n^2 \\ K_{DI} &= 2\zeta\omega_n \end{aligned} \tag{3.38}$$

the quaternion error is calculated with the quaternion difference operator, Θ, as

$$\begin{aligned} \mathbf{q}_{err}(k) &= \mathbf{q_c}(k)\Theta\hat{\mathbf{q}}(k) \\ \begin{bmatrix} q_{1e} \\ q_{2e} \\ q_{3e} \\ q_{4e} \end{bmatrix} &= \begin{bmatrix} q_{4c} & q_{3c} & -q_{4c} & -q_{4c} \\ -q_{3c} & q_{4c} & q_{1c} & -q_{2c} \\ q_{2c} & -q_{1c} & q_{4c} & -q_{3c} \\ q_{1c} & q_{2c} & q_{3c} & q_{4c} \end{bmatrix} \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \\ \hat{q}_4 \end{bmatrix} \end{aligned} \tag{3.39}$$

where $\hat{\mathbf{q}}$ is the current estimated quaternion and q_c is the command quaternion which is $\begin{Bmatrix} 0 & 0 & 0 & 1 \end{Bmatrix}^T$ during eclipse and during the sun following phase, the attitude command according to [**?**] can be calculated as

$$\mathbf{q}_c = \begin{bmatrix} \mathbf{u}_c sin(\frac{\delta}{2}) \\ cos(\frac{\delta}{2}) \end{bmatrix}, \tag{3.40}$$

where

$$\mathbf{u}_c = \frac{\mathbf{u}_{sp}^{SBC} \times \mathbf{s}_o}{\left\| \mathbf{u}_{sp}^{SBC} \times \mathbf{s}_o \right\|}. \tag{3.41}$$

$\mathbf{s}_o$ 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 $\mathbf{s}_o$, δ, can be calculated with the vector dot-product. This can then be used as the reference for the control. The reference $\boldsymbol{\omega}_b^I$ is always $[0, 0, 0]$.

TODO: Produce graphs for the control (Pointing accuracy) with the eclipse phases.

3.6.2. 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

$$\mathbf{M} = \frac{\mathbf{e} \times \mathbf{B}}{\|\mathbf{B}_b\|^2} \tag{3.42}$$

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

$$\mathbf{e} = -K_w(\mathbf{h_w} - \mathbf{h_w}, \mathbf{ref}) \tag{3.43}$$

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.

TODO: Show graph of momentum dumping plot (Reaction wheels momentum with and without momentum dumping)...

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

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 a inaccurate model update, since the control torque and the torque implemented is not the same. This anomaly will aslo be predicted based on the sensor readings, since the effect will be evident on all the sensors.

5.1. Reflection of Solar Panels on Sun Sensor

Sun reflection from solar panels unto sun sensors is a very probable anomaly, if the sun sensor is not placed in a position where the reflection will not influence the sensor. Reflection on the sun sensor is modelled to determine whether the reflection has a significant effect on the ADCS. The reflection anomaly is modelled for the specific shape and design of the CubeSat as shown in Figure 5.1. The CubeSat will also be the same design for all of the modelled anomalies.

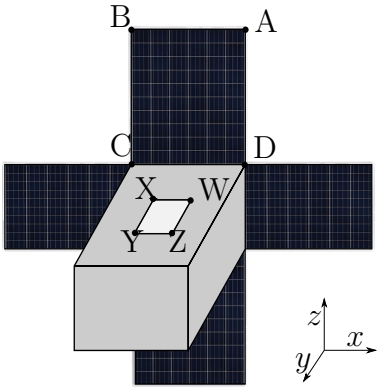


Figure 5.1: Cube Sat

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

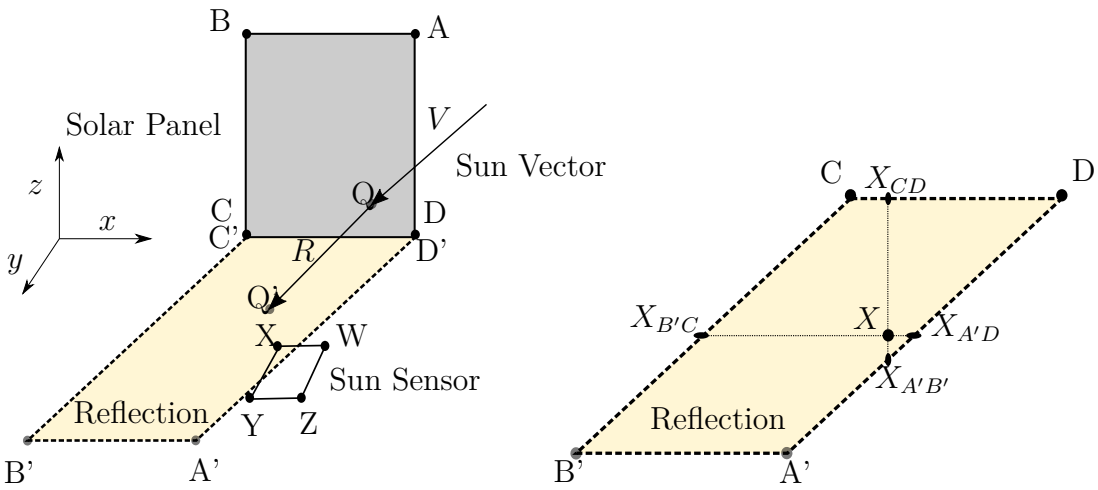


Figure 5.2: Reflection

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, α, β and ζ 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

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

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.

5.1.1. Influence of anomaly on estimation

To determine whether the reflection on the sun sensor has a influence on the ADCS, the estimation metric is shown in Figure 5.4. 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. 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.

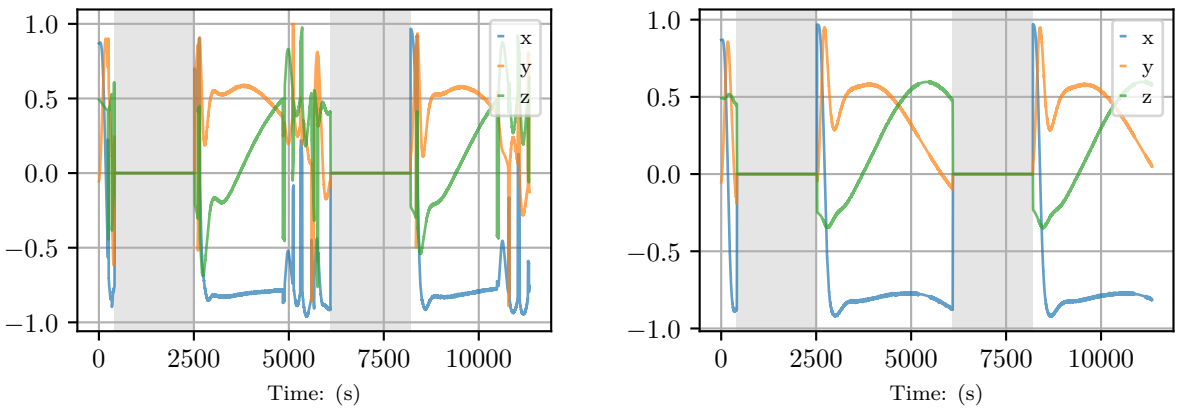


Figure 5.3: Comparison of Sun Vector with and without Reflection

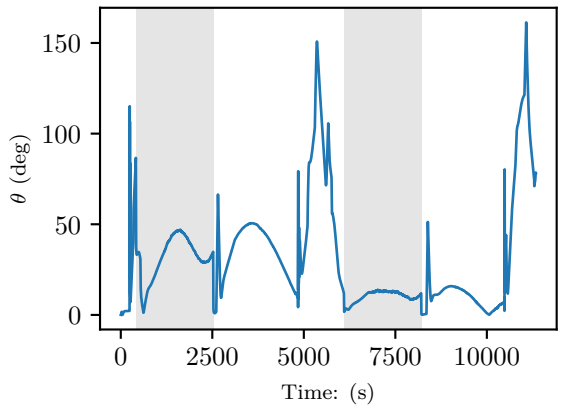


Figure 5.4: Estimation Metric with reflection on sun sensor

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 5.7. 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 5.5. 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

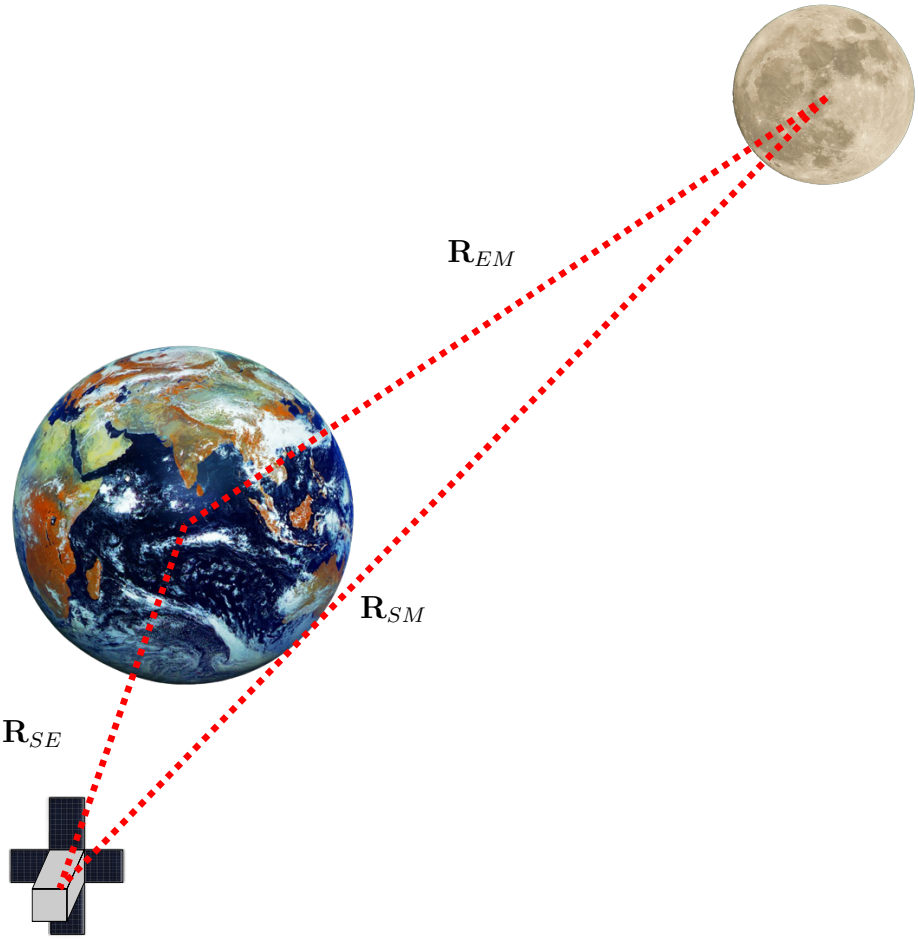


Figure 5.5: Earth to Moon and Earth to Satellite Vectors

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

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 5.6.

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.

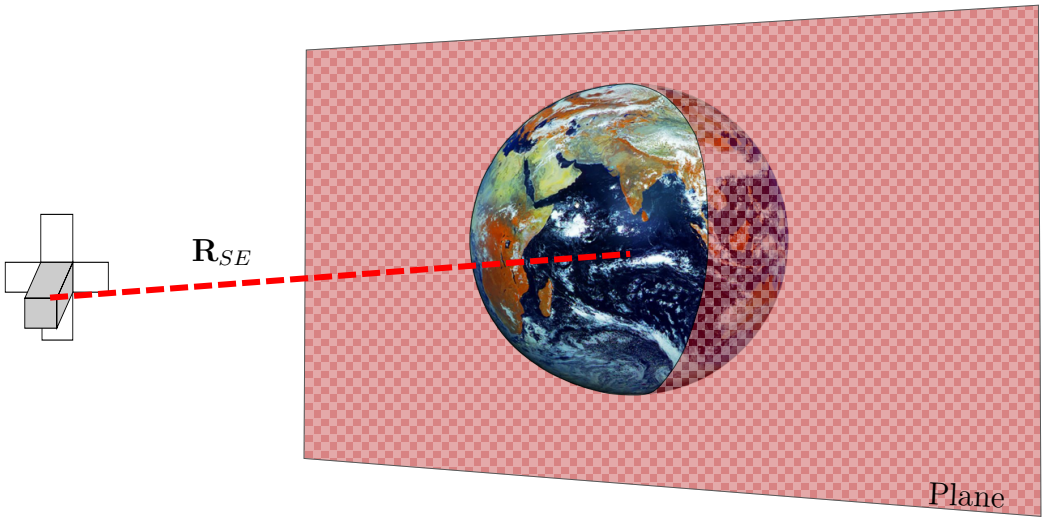


Figure 5.6: Plane perpendicular to $\mathbf{R}_{SE}$ and at center of earth

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 5.6. 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.

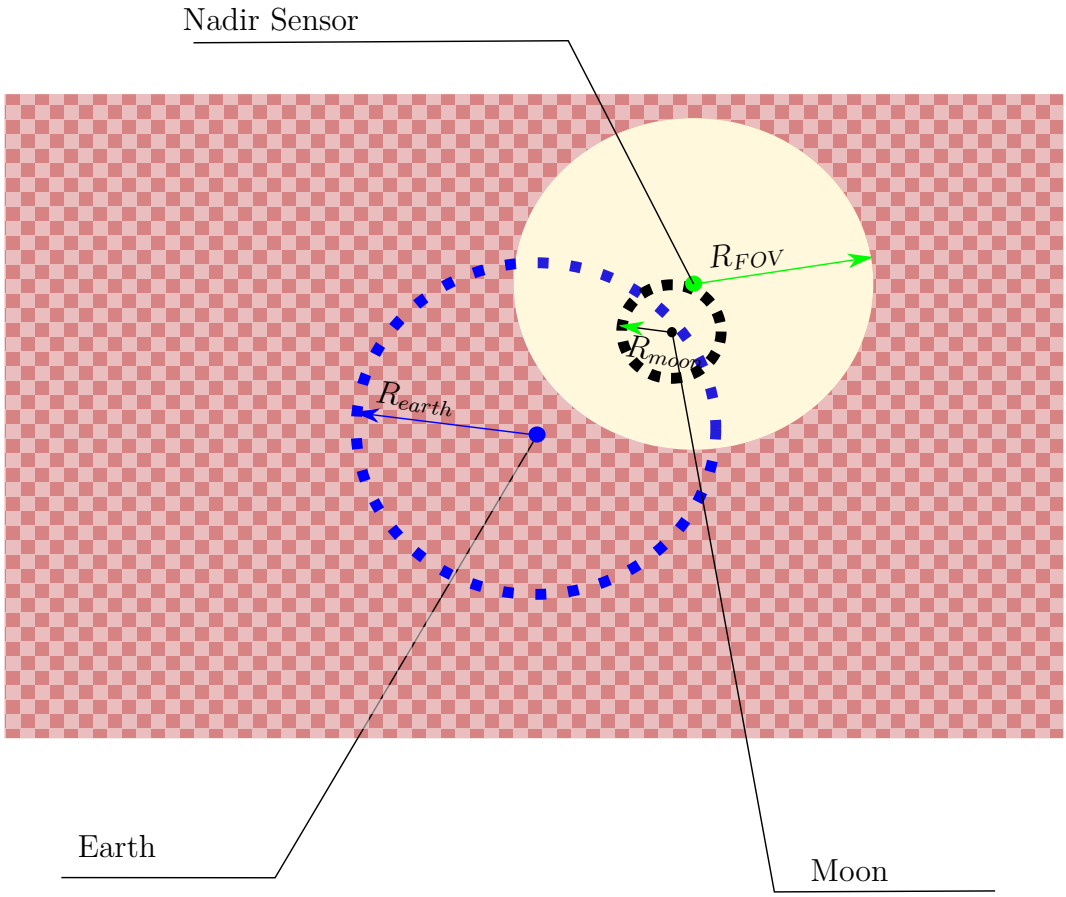


Figure 5.7: Projection of moon and earth on plane

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 5.8 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.

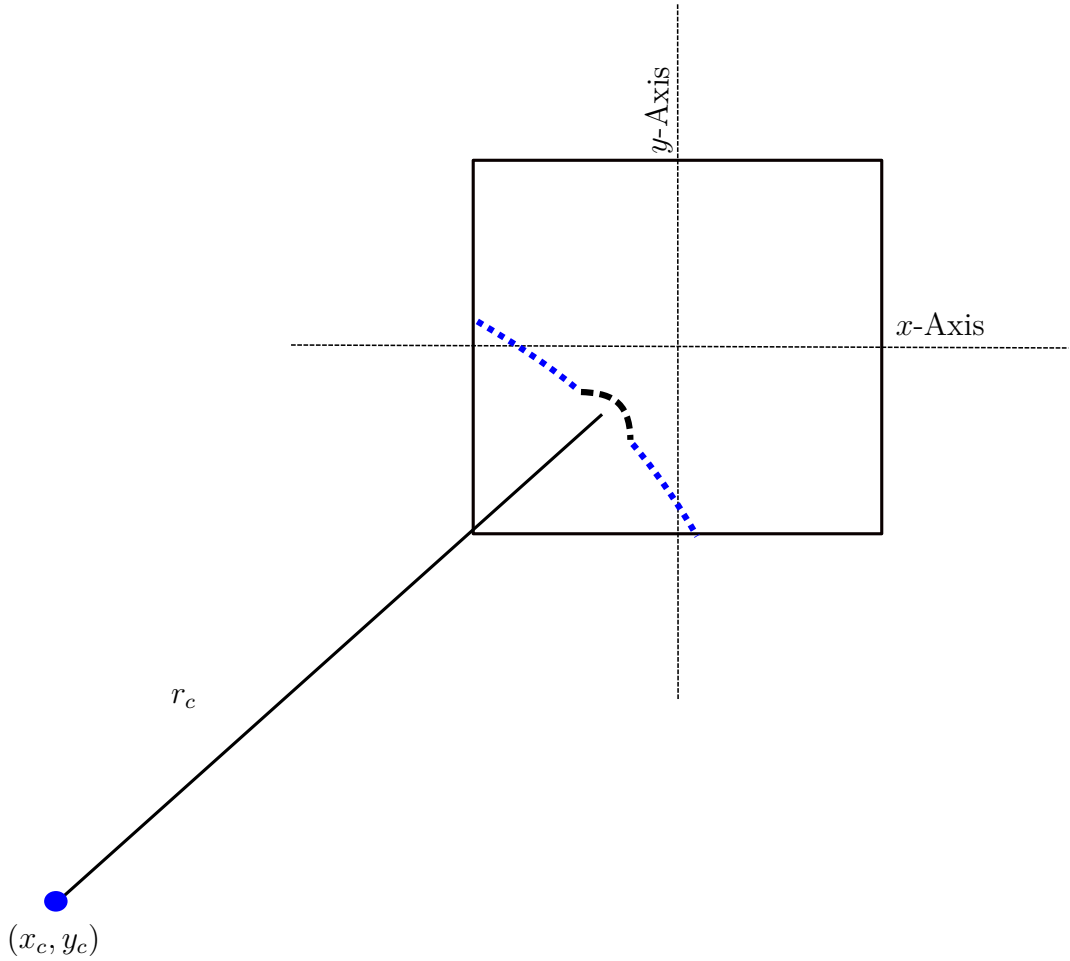


Figure 5.8: Circular Fit Algorithm

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

To determine the effect of the moon in the circular fit algorithm the measured earth vector with and without the moon on horizon anomaly is shown in Figure 5.9. It is clear from Figure 5.9 that the anomaly has no visible effect on the earth vector. It is also evident in Figure 5.10 that the estimation metric is also not influenced negatively by this anomaly. Therefore, this anomaly is not included in the FDIR development, since there is no evident difference due to the anomaly.

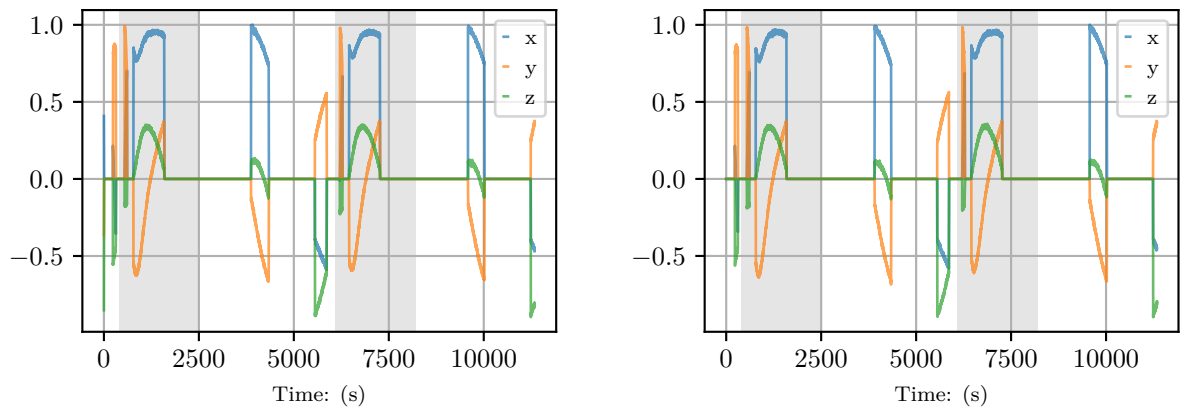


Figure 5.9: Comparison of Earth Vector with and without moon on the horizon

5.3. Magnetic Moment Disturbance from Satellite Bus

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 the coil of the solar panel [**?**, **?**]. According to [**?**] 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

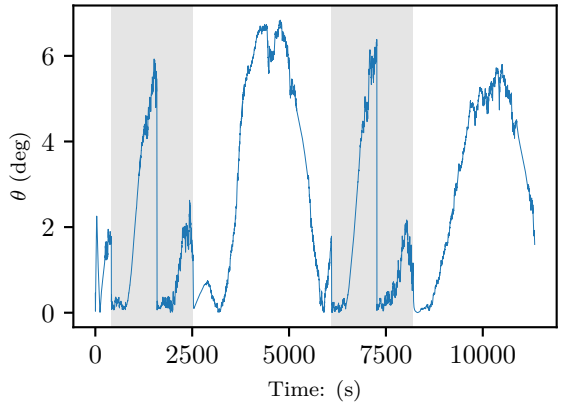


Figure 5.10: Earth Values

modelled for the specific size of the CubeSat model in Figure 5.1. The coil in the solar panel and the resulting magnetic field, $B(r)$, as well as the resulting dipole moment, m, is shown in Figure 5.11. The inner area of the coil is assumed to be the same as the surface area of the solar panel.

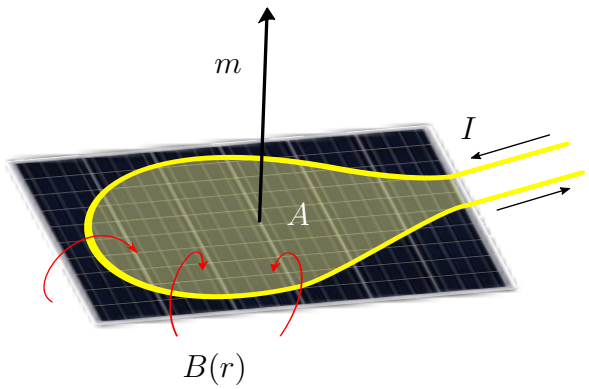


Figure 5.11: Dipole Moment from circular loop in solar panel

The dipole moment is calculated as

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

where A is the area within the loop and I is the current within the coil. 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 as demonstrated in Figure 5.12. Therefore the current, I, can be calculated with

$$I = I_{max} \frac{A_{total}}{A_{illuminated}} cos(\theta) \tag{5.17}$$

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

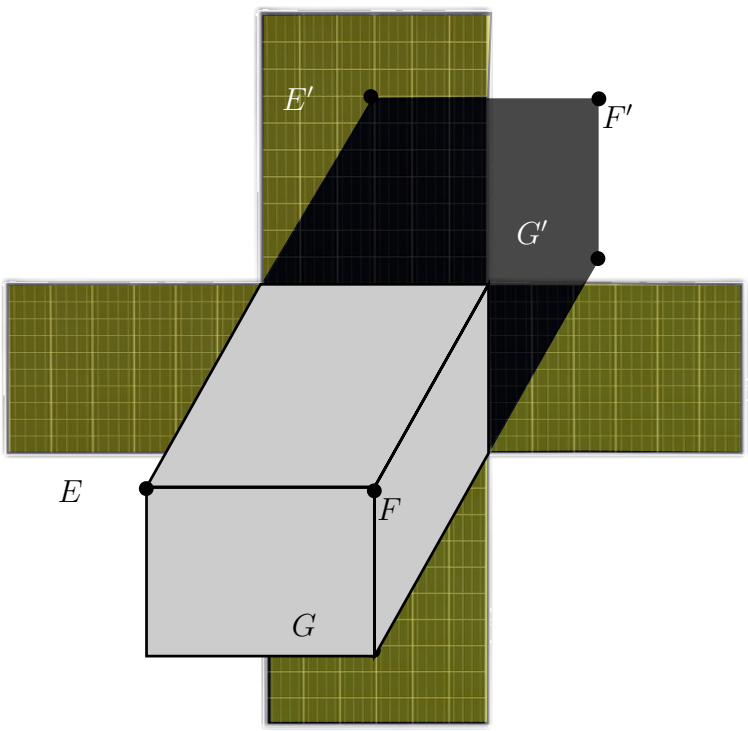


Figure 5.12: Shadow created by CubeSat body on Solar Panels

disturbance torque on the CubeSat. With the resulting torque expressed as

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

where B is the magnetic field of the earth. Since the only external magnetic field that create a resulting torque is that of the earth. The resulting torque for five orbits are shown in Figure 5.13.

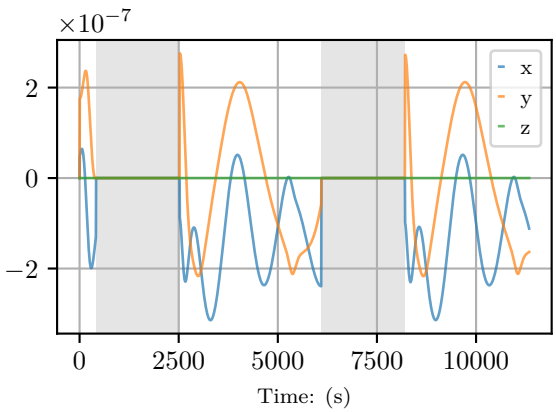


Figure 5.13: Solar Panel Disturbance Torques

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

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

5.3.1. Influence of anomaly on estimation

The vector r 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 and the magnetic field produced by the coils in the solar panels. The resulting magnetometer measurement with and without the induce moment is shown in Figure 5.14. From Figure 5.15 is is evident that this anomaly has a significant effect on the estimation, but not as large as the reflection on the sun sensor.

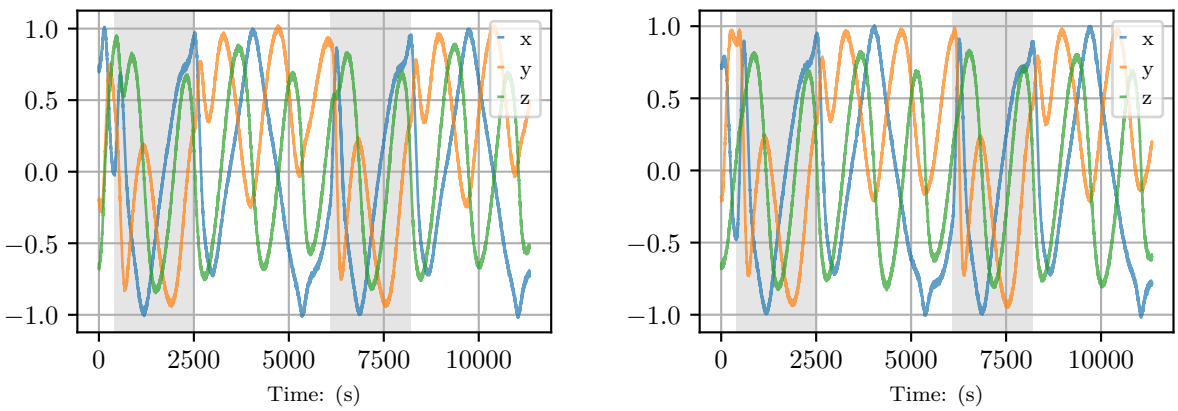


Figure 5.14: Comparison of Magnetic field Vector with and without solar Panel magnetic field

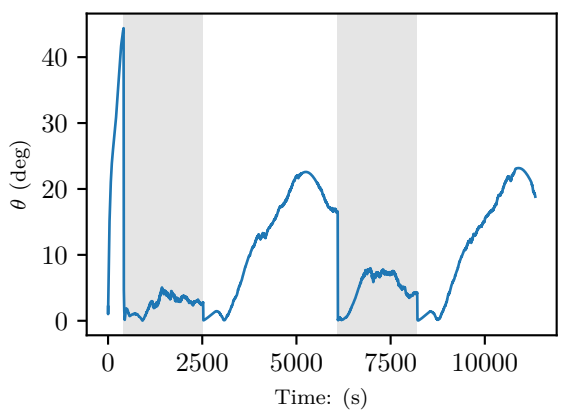


Figure 5.15: Estimation Metric with induced dipole moment

5.4. Reaction wheels

When an actuator fails it influences all the sensor measurements. The anomaly will be modelled as a sudden failure in the actuator when it does not react to inputs. 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 it often occurs and a lot of research is done in this. The recovery of this will however will not be within the specification of this thesis, but the model update for the estimation will be modified based on a modified torque vector. The resulting estimation metric for this anomaly is shown in Figure 5.24 and it is evident that this anomaly has a large negative effect on the EKF. **maybe use the angular moment sensors to update the torque**.

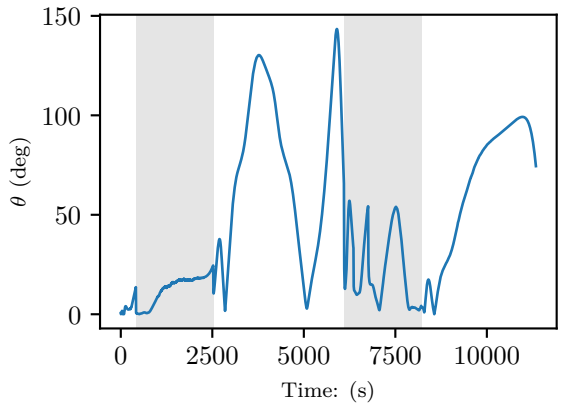


Figure 5.16: Estimation Metric with failure of Reaction Wheels

5.5. Summary

Based on the influence of the modelled anomalies on the estimation metric. Therefore the reflection of the solar panels on the sun sensor, the induced magnetic dipole and the failed reaction wheel are used for the FDIR development.

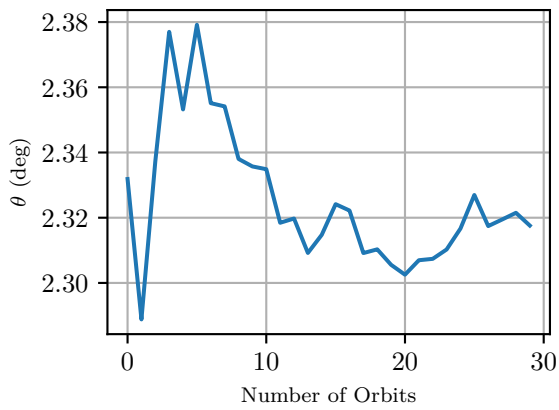


Figure 5.17: Estimation Metric with failure of Reaction Wheels

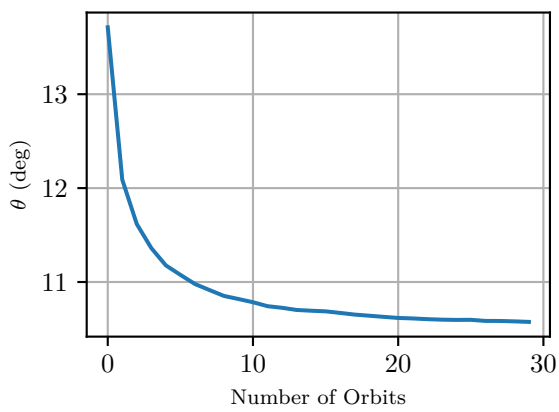


Figure 5.18: Estimation Metric with failure of Reaction Wheels

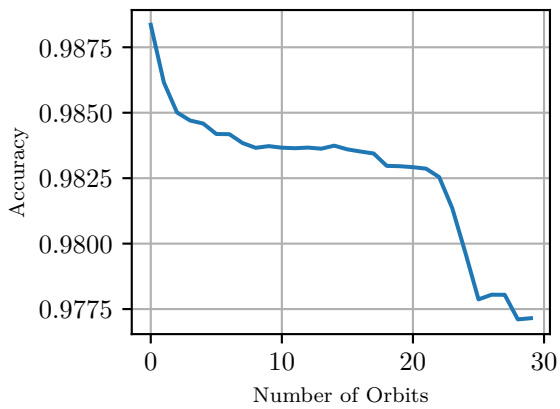


Figure 5.19: Estimation Metric with failure of Reaction Wheels

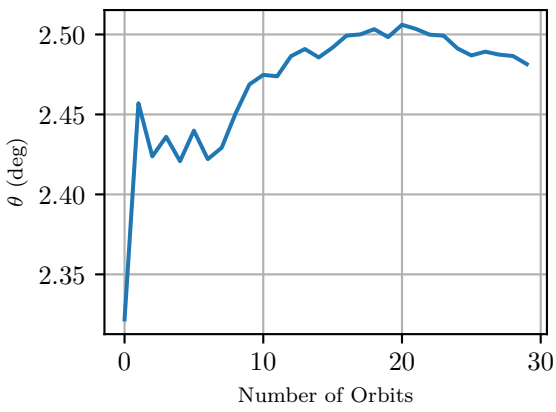


Figure 5.20: Estimation Metric with failure of Reaction Wheels

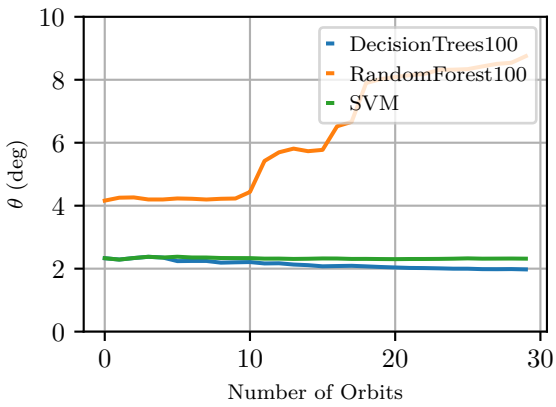


Figure 5.21: Estimation Metric with failure of Reaction Wheels

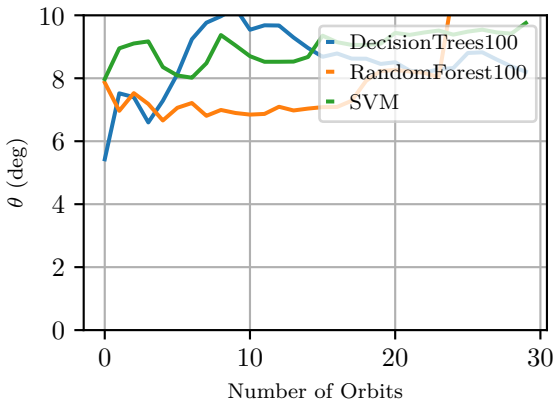


Figure 5.22: Estimation Metric with failure of Reaction Wheels

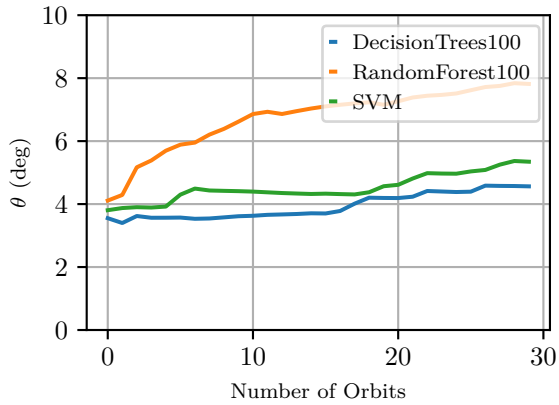


Figure 5.23: Estimation Metric with failure of Reaction Wheels

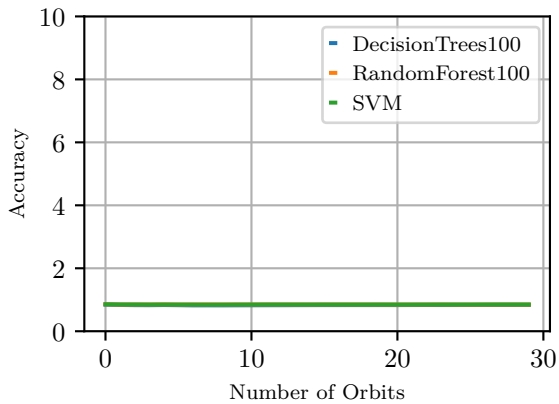


Figure 5.24: Estimation Metric with failure of Reaction Wheels

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. Local Outlier Factor

6.1.2. 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.

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

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

$$\mathbf{A} = \mathbf{X}\mathbf{X}^\dagger \tag{6.2}$$

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.

$$\mathbf{X}_{k+1} = \mathbf{A}\mathbf{X}_k + \mathbf{B}\mathbf{Y}_k \tag{6.3}$$

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 6.3 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 $\mathbf{X}_{k+1}$

$$\hat{\mathbf{X}}_{k+1} = \mathbf{A}\hat{\mathbf{X}}_k + \mathbf{B}\mathbf{Y}_k + K(\mathbf{X}_k - \hat{\mathbf{X}}_k) \tag{6.4}$$

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

$$\mathbf{V}_k = \frac{1}{N}\sum_{i=k-N}^{k}(\mathbf{X}_i - \hat{\mathbf{X}}_i)(\mathbf{X}_i - \hat{\mathbf{X}}_i)^T \tag{6.5}$$

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 $\mathbf{X}$.

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.

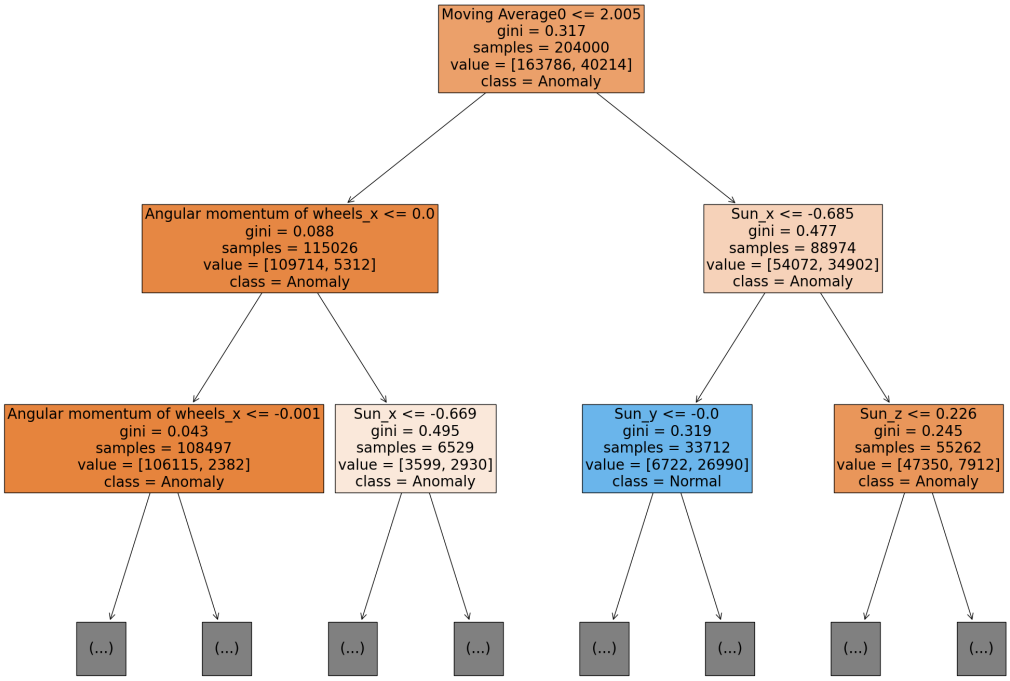


Figure 8.1: Decision Tree

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.

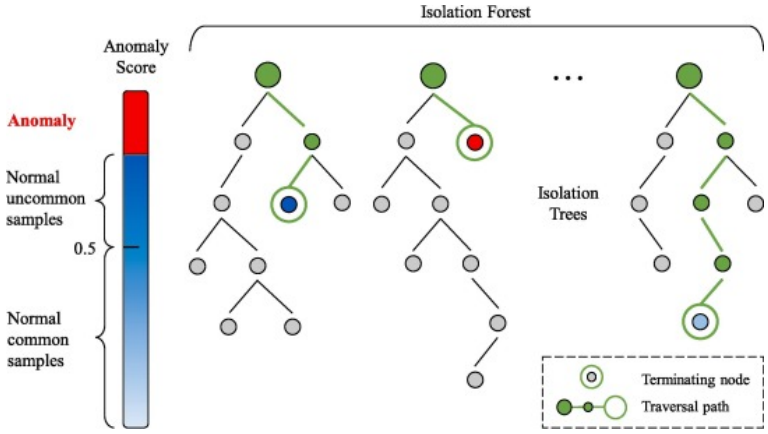


Figure 8.2: Isolation Forests [?]

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

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

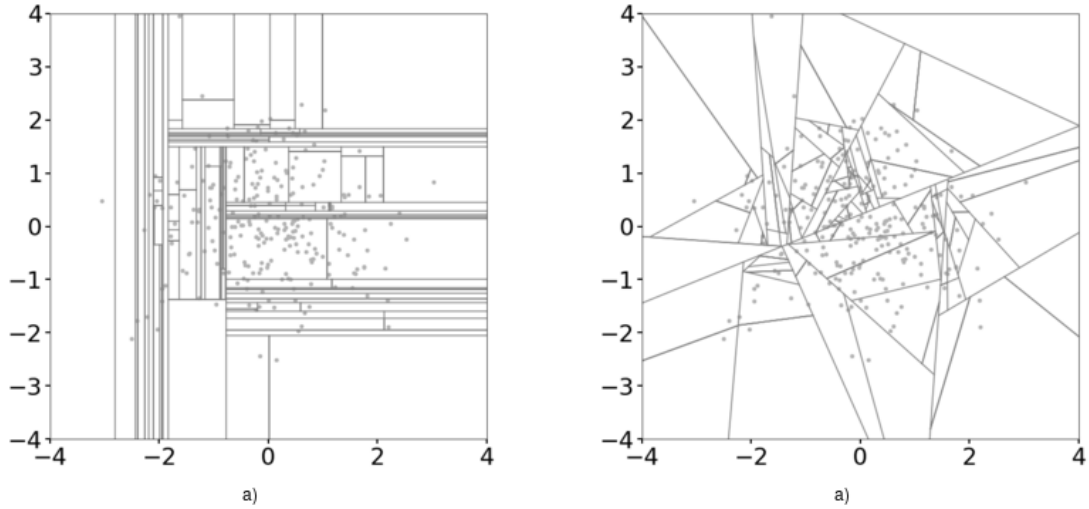
$H(i)$ in Eq 8.3 is the harmonic number and is estimated with Euler's constant as

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

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 kth 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,

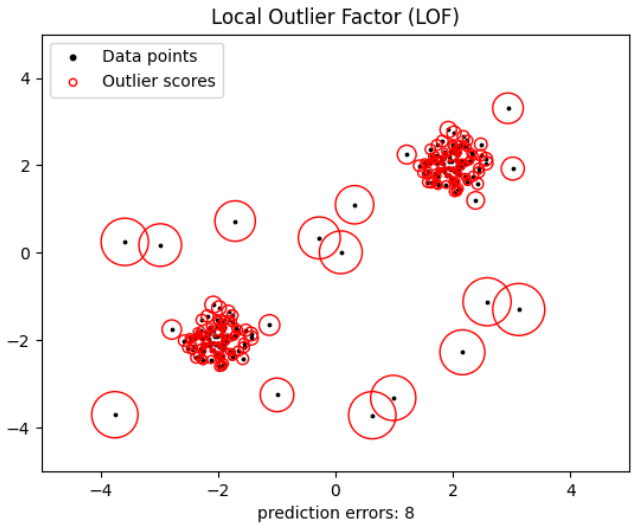


Figure 8.4: LOF measure

p, is calculated as

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

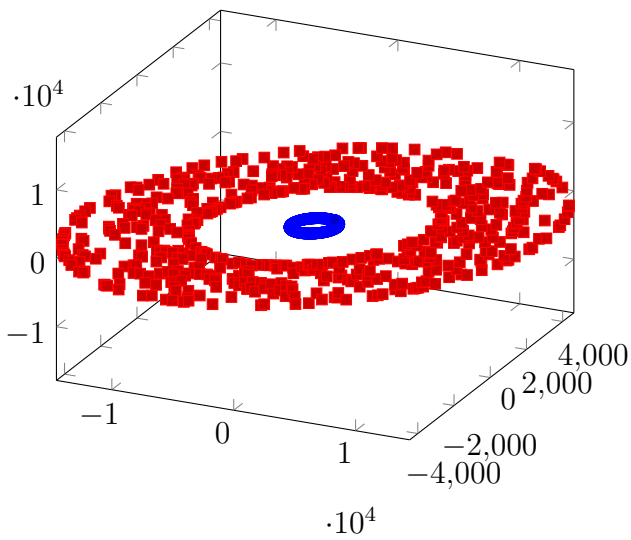
and denotes "the inverse of the average reachability distance based on the $MinPts$-nearest neighbours of the p" — [**?**]. Eq 8.6 enables the calculation for the LOF of point p as shown in Eq 8.7

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

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.

Bibliography

Appendix A

Project Planning Schedule

This is an appendix.

Appendix B

Outcomes Compliance

This is another appendix.