ATTITUDE DETERMINATION USING ASYNCHRONOUS MULTISENSOR FUSION

ANIKET AMOL PRABHUDESAI

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

GRADUATE PROGRAM IN MECHANICAL ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO

JANUARY 2022

© ANIKET AMOL PRABHUDESAI, 2022

ABSTRACT

Satellite failures are a growing concern for amateur missions, specifically for missions involved in the development of CubeSats. Many of these failures stem from the chosen attitude determination algorithm. Hence, this thesis seeks to produce an attitude determination system standard for nanosatellites. To do so, a thorough analysis of attitude and orbit determination, and orbit propagation techniques are discussed and analyzed. The ESSENCE CubeSat is the primary focus of this study, as it is an amateur student led mission. Various sensors used on the ESSENCE 3U CubeSat mission are examined to achieve the goal of this thesis. The utilized sensors are explicitly chosen to be of low cost and affordable for most CubeSat missions.

The thesis proves its functionality while conducting attitude and orbit determination, and orbit propagation techniques. There are two environments, where testing and analysis took place to meet the objectives defined for this thesis. The simulated environment made use of a LEO satellite simulator to examine attitude and orbit determination algorithms. Moreover, a system analysis was done which explored various forms of failure experienced while conducting attitude estimation. Lastly, a hardware in the loop experiment was conducted using a cellphone's inertial measurement unit. The rotational movements experienced in a LEO CubeSat are imitated to validate the outcomes presented in the simulated environment.

This thesis provides a thorough analysis under nominal and failure conditions. Failure outcomes are provided which are critical for the ESSENCE CubeSat missions to consider. The outcome of the thesis produces an algorithm that is chosen as the standard for attitude determination for nanosatellites and is the recommended algorithm for the ESSENCE mission.

ACKNOWLEDGMENTS

I would not have been able to complete this thesis if it were not for several key individuals in my life who supported me throughout this journey. First and foremost, I would like to extend my deepest gratitude to my supervisor Professor Zheng Hong (George) Zhu. His constant guidance and support have allowed me to obtain an ultimate learning experience. It has been a great privilege to be given the opportunity to work with and be a part of the Zhu laboratory. Thank you for being a fantastic mentor. Moreover, I would like to thank the Canadian Space Agency, as the Canadian CubeSat Project (CCP) coined the topic of this thesis.

I would also like to thank Professor Franz Newland. His guidance and recommendations were something I heavily relied on when faced with adversity. He has a unique teaching approach which I highly appreciate, as it has increased my knowledge in space engineering. He has also allowed to me to be involved in many of his projects, which has opened multiple opportunities for me in the space industry. Thank you for your constant dedication and guidance.

I would like to thank my colleagues; Mike Alger and Rajika Pati Arambage, for their unbelievable knowledge, ability to teach and excellent advice. Whenever my back was against the wall, Mike and Rajika were always there to help me solve the problem. Thank you for providing so much valuable input on my thesis through our meetings and discussions. Thank you for being incredibly supportive and I wish you both nothing but success in all your future endeavours.

Finally, I would like to thank my best friend; Gurnoor, who stood by me, trusted me, and inspired me, especially in the moments when I found myself stressed beyond belief. She has taught me many valuable lessons and her nature to help others is a rare trait in today's world. I am truly thankful for having you in my life.

This thesis is dedicated to my amazing parents, Amol and Anagha Prabhudesai, who never stop believing in me and loving me. All my accomplishments are because of the sacrifices you both have made for me. I made a promise to you two at our dinner table, and I fully intend to keep it. If I were to personify hard work, selflessness, and growth; it would be the both of you. Thank you for your support and for sheltering me during my setbacks. Words are not enough to express how much you both mean to me. Thank you for everything.

TABLE OF CONTENTS

Abstract ii
Acknowledgments iii
Table Of Contents iv
List of Tables vii
List of Figures viii
List of Abbreviations xii
Chapter 1 Introduction and Motivation 1
1.1 Background 1
1.2 ESSENCE Mission Overview and Objectives 1
1.3 Discussions on Existing CubeSat Failures 3
1.3.1 CubeSat Failure Causes: Electrical Subsystem 3
1.3.2 CubeSat Failure Causes: Command and Data handling Subsystem 4
1.3.3 CubeSat Failure Causes: Communication Subsystem 5
1.3.4 CubeSat Failure Causes: Attitude Determination and Control Subsystem 6
1.4 Analysis of Common Practices and Standards used in Engineering Domain 8
1.4.1 European Cooperation for Space Standardization (ECSS) 9
1.4.2 PC104 standardization 11
1.4.3 CubeSat Deployers 12
1.4.4 Onboard Software Development kit 13
1.5 Research Question 14
1.6 Research Objectives 15
1.7 Thesis Layout 16
Chapter 2 Literature Review 18
2.1 Optimized TRIaxial Attitude Determination (TRIAD) Method 18
2.2 Davenport's Q Method 21
2.3 Quaternion Estimator (QUEST) Method 23
2.4 Multiplicative Extended Kalman Filter (MEKF) 26
2.4.1 MEKF- Dynamics and Kinematics based System Model 27
2.4.2 MEKF-Kinematics based System Model 29
2.4.3 Asynchronous Sensor Fusion 31
2.5 Conclusion 33

Chapter 3 Overview of Attitude Determination and Control and Reference Frames 34
3.1 Overview of Attitude Determination and Control Subsystem (ADCS) 34
3.2 Coordinate Frames 36
3.2.1 Body Fixed Frame (BFF) 36
3.2.2 Perifocal Reference Frame (PQW) 37
3.2.3 True Equator Mean Equator (TEME) Frame 38
3.2.4 January 2000 (J2000) frame (ECI) 39
3.2.5 Earth Centered Earth Fixed (ECEF) Frame 40
3.2.6 North East Down (NED) Frame 41
3.2.7 Spherical Coordinates with respect to ECEF 41
3.3 Equations of Motion 42
3.3.1 Attitude Dynamics 42
3.3.2 Disturbance Torques 44
3.3.3 Attitude Kinematics – Quaternions 48
Chapter 4 Overview of Attitude and Orbit Determination Sensors 51
4.1 Three-axis MEMS Gyroscope 54
4.2 Fibre Optic Gyroscope (FOG) 58
4.3 Coarse Sun Sensor (CSS) 61
4.4 Fine Sun Sensor (FSS) 64
4.5 Magnetometer 67
4.6 Global Navigation Satellite System (GNSS) Receiver 69
Chapter 5 Orbit Determination/Propagation 73
5.1 Overview of Time Domain 73
5.2 Simplified General Perturbation (SGP) Model 76
5.3 Onboard GPS receiver centric Kalman Filter 93
5.4 Environmental Models 101
5.4.1 Sun Position Model 101
5.4.2 Magnetic Field Vector Model 104
Chapter 6 Static Attitude Determination Algorithms Results and Analysis 111
6.1 Attitude Estimation using Optimized TRIAD Algorithm 111
6.1.1 Optimized TRIAD Algorithm: Simulated Results and Analysis 114
6.2 Attitude Estimation using Davenport's Q-method Algorithm 117
6.2.1 Davenport's Q-method Algorithm: Simulated Results and Analysis 120
6.3 Attitude Estimation using QUEST Algorithm 123
6.3.1 Quest Algorithm: Simulated Results and Analysis 126

Chapter 7 Online Attitude Determination Algorithms Analysis and Results 130
7.1 Attitude Estimation using MEKF Algorithm: Dynamics and Kinematics 130
7.2 Asynchronous Sensor Fusion Techniques 140
7.2.1 Zero Order Hold (ZOH) Technique: Results and Analysis 141
7.2.2 Frequency Based Update Algorithm (FBUA): Results and Analysis 145
7.3 System Analysis: Dynamics and Kinematics based MEKF 147
7.4 Attitude Estimation using MEKF Algorithm: Kinematics Only 160
7.4.1 System Analysis: Kinematics only based MEKF 166
Chapter 8 Hardware in the Loop Experiment 174
8.1 Experiment: Introduction and Setup 174
8.2 Reference Frames and Vector Definition for HIL Experiment 177
8.2.1 Reference Frame Definition 178
8.2.2 Local Tangent Frame Magnetic field and Acceleration Vectors 180
8.3 Sensor Calibration 183
8.4 MEKF Results and Analysis 186
8.4.1 Dynamics and Kinematics System Model: Results and Analysis 187
8.4.2 Kinematics System Model: Results and Analysis 192
Chapter 9 Conclusion and Future Work 197
9.1 Thesis Conclusions 197
9.1.1 Orbit Determination/Propagation Conclusions 197
9.1.2 Static Attitude Determination Conclusions 198
9.1.3 Online Attitude Determination Conclusions 199
9.1.4 Hardware in the Loop Experiment Conclusions 200
9.2 Thesis Contributions 200
9.3 Future Work 202
References 204

LIST OF TABLES

Table 3.1: ESSENCE CubeSat dimensions and COM breakdown..... 43
Table 4.1: Parameters of the MEMS gyroscope integrated in the Nanomind A3200 [73] [74]. .. 56
Table 4.2: Parameters of Fizoptika VGA091A in conjunction with ADS1115 ADC [76] [77]... 59
Table 4.3: NanoPower P110 parameters relevant via datasheet [78]. 62
Table 4.4: Nanosense Fine Sun Sensor parameters relevant via datasheet [79] [80]. 66
Table 4.5: Honeywell HMC5843 parameters relevant via datasheet [73] [81]. 68
Table 4.6: Parameters of WGS-84 ellipsoid [82]..... 69
Table 4.7: Shows the RMS error in position and velocity for the GNSS Receiver [84]. 71
Table 5.1: Shows the Time difference parameters from Bulletin A on the IERS website [86].... 75
Table 5.2: Orbital elements and other parameters parsed and derived from the NORAD TLEs. 76
Table 5.3: Initial parameters defined for the GPS- based Kalman Filter..... 98
Table 6.1: Optimized TRIAD and Davenport's Q-Method Algorithm parameters..... 114
Table 6.2: QUEST Algorithm parameters. 126
Table 6.3: RMS Error expressed in Roll, Pitch and Yaw for all static attitude algorithms..... 128
Table 7.1: MEKF steps used for online attitude determination. 131
Table 7.2: Shows the frequency ranges for the attitude sensors used on the ESSENCE CubeSat. 140
Table 7.3: Shows the frequencies used for the simulated testing environment. 141
Table 7.4: Outlines the initial parameters used for the Dynamics and Kinematics based MEKF. 142
Table 7.5: Shows the angles used for a 3-2-1 rational sequence to produce a misaligned matrix. 154
Table 7.6: Shows the RMS error in the Euler angles for all three operational frequencies..... 159
Table 7.7: Shows the RMS error values in the angular velocity for all three operational frequencies. 160
Table 7.8: Shows the nominal conditions for the kinematics based MEKF. 162
Table 7.9: Shows the error values in the Euler angles for all three operational frequencies..... 173
Table 8.1: Shows the error values of the sensors found in the android phone's IMU [99] [100]. 175
Table 8.2: Geodetic parameters used for HIL experiment..... 178
Table 8.3: Show the magnetic field results of the IGRF-13 Model, WMM and the difference between the two models..... 182
Table 8.4: Shows the Hard Iron bias and the Soft Iron Distortion matrix using "*magcal*"..... 184
Table 8.5: Android device dimensions, mass, and moment of inertia matrix. 188
Table 8.6: Initial state vectors, covariance and the process noise matrix for the Dynamics and Kinematics based MEKF. 189
Table 8.7: Initial state vectors, covariance and the process noise matrix for the Dynamics and Kinematics based MEKF. 194

LIST OF FIGURES

Figure 1.1: ESSENCE 3U-CubeSat bus dimensions via SolidWorks 2
Figure 1.2: Proton detector Payload for the ESSENCE and DESCENT CubeSat missions. (A) Proton detector Payload. (B) Optical payload. 3
Figure 1.3: PC104 stack used for a custom board used for ESSENCE Burn Wire Mechanism via SolidWorks. 12
Figure 1.4: NanoRacks CubeSat Deployer used for DESCENT and ESSENCE missions [5]. ... 13
Figure 3.1: Summation of ADCS for a typical satellite [66]. 35
Figure 3.2: Body Fixed Frame for the ESSENCE 3U-CubeSat. 37
Figure 3.3: PQW Geocentric Coordinate system [68]. 38
Figure 4.1: ESSENCE CubeSat's Interconnect Diagram outlining placement of attitude sensors in BFF. 53
Figure 4.2: Nanomind A3200 3-axis gyroscope and magnetometer sensor frame [73]. 54
Figure 4.3: True angular velocity in the system, measured quantity, noise & bias present in the measured quantity for MEMS gyroscope. 57
Figure 4.4: Fibre Optic Gyroscope operational view [75]. 58
Figure 4.5: Sensor model simulated via Simulink using (4.2) for MEMS and Fibre Optic gyroscopes..... 60
Figure 4.6: True angular velocity in the system, measured quantity, noise & bias present in the measured quantity for the FOG..... 61
Figure 4.7: Measured angle described in BFF using for +X face of the ESSENCE CubeSat (4.3). 63
Figure 4.8: True Angle with sun vector and boresight vector, measured quantity, sensor visibility flag and CSS noise. 64
Figure 4.9: Sensor model simulated via Simulink using **(4.4)** and **(4.7)** for the CSS and FSS. ... 66
Figure 4.10: Shows the true sun position unit vector, measured sun position for +Z face, Visibility Flag of +Z face and noise associated with the FSS. 67
Figure 4.11: Sensor model simulated via Simulink using for MEMS magnetometer. 68
Figure 4.12: Shows the true magnetic field vector, measured magnetic field vector, and noise associated with the magnetometer. 68
Figure 4.13: OEM615 Receiver System operational diagram [85]. 70
Figure 4.14: OEM 615 GNSS Receiver sensor model using (4.9) and (4.10)..... 71
Figure 4.15: Shows the true CubeSat position, measured CubeSat position, noise associated with the measured quantity in ECEF for the GPS receiver..... 72
Figure 4.16: Shows the true CubeSat velocity, measured CubeSat velocity, noise associated with the measured quantity in ECEF for the GPS receiver..... 72
Figure 5.1: NORAD TLE of ISS Zarya from CelesTrak [87]. 76
Figure 5.2: Demonstrates relation between rotation R3(-Ω) R1(-i) R3(-ω) [89]..... 82
Figure 5.3: Position in TEME frame generated from the Simulink SGP model, STK orbit

propagator and the difference between the true and simulated quantity. 83
Figure 5.4: Velocity in TEME frame generated from the Simulink SGP model, STK orbit propagator and the difference between the true and simulated quantity. 83
Figure 5.5: Position in J2000 frame generated from the Simulink SGP model, STK orbit propagator and the difference between the true and simulated quantity. 88
Figure 5.6: Velocity in J2000 frame generated from the Simulink SGP model, STK orbit propagator and the difference between the true and simulated quantity. 88
Figure 5.7: Position in ECEF frame generated from the Simulink SGP model, STK orbit propagator and the difference between the true and simulated quantity. 92
Figure 5.8: Velocity in ECEF frame generated from the Simulink SGP model, STK orbit propagator and the difference between the true and simulated quantity. 92
Figure 5.9: GPS Receiver based Kalman Filter's Software Algorithm Schematic. 94
Figure 5.10: Simulink Model generated for the GPS Receiver based Kalman Filter. 99
Figure 5.11: Position in ECEF frame generated from the Simulink GPS-based Kalman Filter, STK orbit propagator and the difference between the true and simulated quantity. 99
Figure 5.12: Velocity in ECEF frame generated from the Simulink GPS-based Kalman Filter, STK orbit propagator and the difference between the true and simulated quantity. 100
Figure 5.13: Illustrations of the penumbra and umbra regions while the orbiting satellite enters an eclipse zone [67]. 102
Figure 5.14: Sun position using equations (5.56) and (5.57), True Sun position from STK, eclipse zones, and the error present in calculation. 103
Figure 5.15: SGP model built in Simulink to produce Magnetic Field and Sun position vectors. 107
Figure 5.16: GPS-based Kalman Filter built in Simulink to produce Magnetic Field and Sun position vectors. 108
Figure 5.17: Shows the Magnetic field generated by both orbit determination methods in the ECEF frame, True Magnetic Field, and error in the measured quantity. 109
Figure 5.18: Shows the Magnetic field generated by both orbit determination methods in the J2000 frame, True Magnetic Field, and error in the measured quantity. 110
Figure 6.1: Simulink Model of Optimized TRIAD algorithm, Sensor Measurement model and the LEO Satellite Simulator. 114
Figure 6.2: Predicted Euler angles produce via Optimized TRIAD algorithm, True Euler angles and Eclipse Flag. 115
Figure 6.3: Error plots shown in terms of Euler Angles, Error in Euler Angles focusing on Non-Eclipse Zone and the eclipse flag for the Optimized TRIAD algorithm. 115
Figure 6.4: Simulink Model of Davenport's Q-method, Sensor Measurement model and LEO Satellite Simulator. 120
Figure 6.5: Shows the predicted quaternion generated via Davenport's Q-method Algorithm, true quaternion, and the eclipse flag. 121
Figure 6.6: Error plots shown in terms of error quaternion, error in Euler angles focusing on the non-eclipse zones and the eclipse flag for the Davenport's Q-method Algorithm. 122
Figure 6.7: Simulink Model of QUEST Algorithm, Sensor Measurement model and the LEO Satellite Simulator. 126

Figure 6.8: Shows the predicted quaternion generated from the QUEST Algorithm, true quaternion, and the eclipse flag. 127
Figure 6.9: Error plots shown in terms of error quaternion, error in Euler angles focusing on the non-eclipse zones and the eclipse flag for the QUEST algorithm. 127
Figure 7.1: Simulink model developed as per the system model defined in **(7.5)**. 139
Figure 7.2: Plots displaying the quaternion and angular velocity states and the true quaternion and angular velocity quantities using the ZOH technique. 144
Figure 7.3: Plots displaying the error quaternion, Euler angles, angular velocity, and eclipse flags quantities using the ZOH technique. 144
Figure 7.4: Plots displaying the error quaternion, Euler angles, angular velocity, and eclipse flags using FBUA. 146
Figure 7.5: Magnetometer failure plots displaying the error quaternion, Euler angles, angular velocity, and visibility flags for the dynamics and kinematics based MEKF. 149
Figure 7.6: Fibre Optic gyroscope failure plots displaying the error in quaternion, Euler angles, angular velocity, and eclipse flag for the dynamics and kinematics based MEKF. 151
Figure 7.7: Multiple sensor failure plots displaying the error in quaternion, Euler angles, angular velocity, and eclipse flags for the dynamics and kinematics based MEKF. 153
Figure 7.8: Sensor misaligned plots displaying the error in quaternion, Euler angles, angular velocity, and eclipse flag. 156
Figure 7.9: Sensor misconfiguration plots displaying the error in quaternion, Euler angles, angular velocity, and eclipse flag. 158
Figure 7.10: Simulink model developed as per the system model defined in (7.34) for the kinematics based MEKF. 162
Figure 7.11: Kinematics based MEKF using FOG plots displaying the quaternion state, true quaternion, and eclipse flag. 163
Figure 7.12: Kinematics based MEKF using FOG plots displaying the error quaternion, error in term of Euler angles, and eclipse flag. 164
Figure 7.13: Kinematics based MEKF using MEMS Gyroscope plots displaying the quaternion state, true quaternion, and eclipse flag. 165
Figure 7.14: Kinematics based MEKF using MEMS Gyroscope plots displaying the error quaternion, error in term of Euler angles, and eclipse flag. 166
Figure 7.15: Shows plots displaying the error in quaternion, Euler angles and Visibility flag for magnetometer failure in the kinematics based MEKF. 168
Figure 7.16: Shows the error in quaternion, Euler angles and eclipse flag for sun sensor failure in the kinematics based MEKF. 169
Figure 7.17: Shows the error plots when FOG sensors are misaligned for the kinematics-based system model. 171
Figure 7.18: Shows the error plots when CSS sensors are misconfigured for the kinematics-based system model. 172
Figure 8.1: Hardware in the Loop experimental setup. 176
Figure 8.2: Displays the NED shown tangent to Earth and the topocentric view of the frame. . 179
Figure 8.3: Illustration of the local tangent frame used in the HIL experiment. 179
Figure 8.4: Illustration of the cell phone's body fixed frame used in the HIL experiment. 180

Figure 8.5: Shows the comparison between the IGRF-13 model and WMM, with the difference between the 2 models. 182
Figure 8.6: 3D plot displaying magnetometer calibration vs uncalibrated values. 185
Figure 8.7: The sensor extraction Simulink blocks for accelerometer, magnetometer, and gyroscope in the cell phone's body fixed frame. 186
Figure 8.8: Schematics used for the AHRS attitude estimator in the HIL experiment. 187
Figure 8.9: Simulink model of the Dynamics and Kinematics based MEKF used for the HIL experiment. 190
Figure 8.10: HIL experiment true vs predicted quaternion plots using Dynamics and Kinematics based system model. 191
Figure 8.11: HIL experiment plots consisting of error quaternion and error expressed in Euler angles for the Dynamics and Kinematics-based system model. 192
Figure 8.12: Simulink model of the Kinematics based MEKF used for the HIL experiment. ... 193
Figure 8.13: HIL experiment true vs predicted quaternion plots using kinematics-based system model. 195
Figure 8.14: HIL experiment plots consisting of error quaternion and error expressed in Euler angles for the kinematics-based system model. 195

LIST OF ABBREVIATIONS

ADCS	Attitude Determination and Control System
AEKF	Additive Extended Kalman Filter
AHRS	Attitude Heading Reference System
CAN	Controller Area Network
CCSDS	Consultative Committee for Space Data System
CDR	Critical Design Review
CIO	Celestial Intermediate Origin
COTS	Commercial Off the Shelf
CSS	Coarse Sun Sensor
DESCENT	Deorbiting Spacecraft using Electrodynamic Tethers
DCM	Direction Cosine Matrix
ECI	Earth Centered Inertial
ECEF	Earth Centered Earth Fixed
ECSS	European Cooperation for Space Standardization
EOP	Earth Orientation Parameters
ESA	European Space Agency
ESSENCE	Educational Space Science and Engineering CubeSat Experiment
FBUA	Frequency Based Update Algorithm
FSS	Fine Sun Sensor
FOG	Fibre Optic Gyroscope
FOV	Field of View
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HIL	Hardware in the Loop
IAGA	International Association of Geomagnetism and Aeronomy
IAU	International Astronomical Union
ICRF	International Celestial Reference Frame
IDD	Interface Definition Document

I2C	Inter-Integrated Circuit
IERS	International Earth Rotation and Reference System
IGRF	International Geomagnetic Reference Field
IMU	Inertial Measurement Unit
ISS	International Space Station
J2000	January 2000
LEO	Low Earth Orbit
MEMS	Micro-electromechanical systems
MEKF	Multiplicative Extended Kalman Filter
NASA	National Aeronautics and Space Administration
NED	North East Down
NRCSD	NanoRacks CubeSat Deployer
OBC	Onboard Computer
PCB	Printed Circuit Board
P-POD	Poly-Picosatellite Orbital Deployer
QUEST	Quaternion Estimator
RMS	Root Mean Square
ROS	Real Time Operating System
SDK	Software Development Kits
SDLC	Systems/Software Development Life Cycle
SGP	Simplified General Perturbation
SPI	Serial Peripheral Interface
STK	Systems Tool Kit
TAI	International Atomic Time
TEME	True Equator Mean Equator
TRL	Technology Readiness Level
TRIAD	TRIaxial Attitude Determination
TT	Terrestrial Time
UART	Universal Asynchronous Receiver and Transmitter
UAV	Unmanned Aerial Vehicles
UHF	Ultra-High Frequency

UTC	Coordinated Universal Time
UT1	Universal Time 1
WGS-84	World Geodetic System
ZOH	Zero Order Hold

CHAPTER 1 INTRODUCTION AND MOTIVATION

1.1 Background

The introduction of CubeSats by California Polytechnic State University and Stanford University in the late 1990s has been extremely important to the space industry [1]. CubeSats are classified under types of nanosatellites, as they contain a wet mass range from 1–10 kg. In addition to this, there are several other design constraints associated with CubeSat structures. These include cost, size, orbital parameters, mission objectives and mission duration. This type of a nanosatellite is described in terms of units (U). Each CubeSat unit has strict dimensions of 10cm x 10cm x 10cm in terms of length, width, and height. Throughout this thesis, York University's active and upcoming CubeSat missions are discussed and referenced. The currently active DEorbiting SpaCecraft using ElectrodyNamic Tethers (DESCENT) is 2U CubeSat mission. Hence, making its dimensions 10cm x 10cm x 20cm when each U is stacked. The second mission is Educational Space Science and Engineering CubeSat Experiment (ESSENCE) CubeSat is 3U. Thus, having dimensions measuring 10cm x 10cm x 30cm with each unit is stacked as shown in Figure 1.1. Although the DESCENT mission is mentioned, the focus is primarily on the ESSENCE mission because the attitude determination system is to be implemented on this mission.

1.2 ESSENCE Mission Overview and Objectives

CubeSats are used primarily for educational purposes, scientific research, technology demonstration and commercial use. Several start-ups along with companies like Canadensys take advantage of CubeSat missions to grow the Technology Readiness Level (TRL) of products, such

as optical payloads. This is the case for York University's current 3U ESSENCE CubeSat mission.



Figure 1.1: ESSENCE 3U-CubeSat bus dimensions via SolidWorks

This is an amateur CubeSat mission funded by the Canadian Space Agency and has successfully completed the critical design review (CDR). This mission features five objectives, which are:

1) Train high quality personnel through their involvement in the development of the 3U-CubeSat in multiple subsystems
2) Achieve a successful ESSENCE launch from the International Space Station (ISS)
3) Conduct various attitude control theory experiments
4) Monitor the thawing of permafrost in northern Canada and Arctic regions
5) Monitor space weather using theories defined to track gamma ray dosage

To meet the needs of objectives 4 and 5, two payloads are externally developed for the 3U CubeSat. The first payload is the Canadensys wide angle camera, and the second payload is a proton detector mounted on a printed circuit board (PCB) provided by the University of Sydney shown in Figure 1.2 A and B. Payloads like the Canadensys wide angle camera and the proton detector are common

due to their small size.

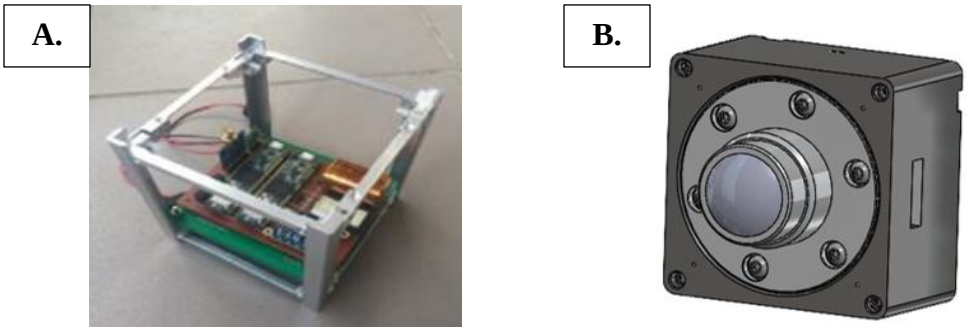


Figure 1.2: Proton detector Payload for the ESSENCE and DESCENT CubeSat missions. (A) Proton detector Payload. (B) Optical payload.

1.3 Discussions on Existing CubeSat Failures

The research motivation for this thesis stems from various causes of CubeSat failures. The failure types addressed in this section seek to analyze various issues in multiple satellite subsystems. These subsystems include electrical, command and data handling, communication and attitude determination and control. Upon the occurrence of the catastrophic issues in the mentioned domains, the satellite is either inactive, inaccessible, or uncontrollable. According to a reliability analysis conducted by Martin Lawrence, CubeSat success ranges from 48.49% to 65.49% once deployed in low earth orbit (LEO) [2]. In this analysis, the mentioned four subsystems contributed to these failure percentages. As of April 4th, 2021, there have been 1684 Nanosatellites and 1553 Cube-satellites have been launched [3] and are expected to be growing due to their cost-efficient nature. These missions are led by first time CubeSat developers which contributes to the failure rates.

1.3.1 CubeSat Failure Causes: Electrical Subsystem

The subsystem which is prone to failure is the electrical subsystem. One such cause of the electrical failure involves the onboard batteries, which can result in overvoltage and undervoltage safety

concerns. Overvoltage occurs when the voltage in a circuit is observed to be above the battery circuits design limits [4]. This results in a shortened lifespan of the component if overvoltage protection is not implemented for the battery circuit. Most suppliers provide this protection, but with additional charges. Thus, to conserve cost, in-house solutions are utilized but at the risk of increased failure. In addition, the exact opposite issue is defined as the undervoltage issue. Undervoltage leads to overheating and inefficient performance, which can also cause the components to have a short operational lifespan [4]. Satellite launch providers such as NanoRacks, have a strict requirement set for these types of components. Due to various hazardous concerns associated with space grade batteries, NanoRacks' CubeSat Deployer Interface Definition Document (IDD) [5] requires further testing if the energy density is greater than 80 Wh. In addition, solar cells are another critical electrical subsystem component. The cells can be easily damaged if precautional measures are not taken to protect the component. These measures include operating in a clean room or a controlled environment, where damage to sensitive equipment is at a low probability. However, this was one the potential causes of power fluctuations on the DESCENT CubeSat. This is echoed in the study conducted by Michael Swartwout which analyzed causes of satellite failures. The study states 17% of failures are due to the power and electrical subsystem, among all other subsystems [6]. Power generation issues caused by this mishap leads to a negative power budget. Hence, causing a potential satellite shutdown as it is not sustainable due a negative power margin. This issue can be solved by utilizing attitude actuators to reorient the solar panels to the most optimal beta angle. However, not all amateur missions have this capability, as it is quite demanding for high constraint applications like CubeSats.

1.3.2 CubeSat Failure Causes: Command and Data handling Subsystem

The second subsystem explored is the Command and Data handling subsystem. Within this area,

the cause of failure revolves around the utilization, allocation, and implementation of the data busses. The most common ones used on satellite missions include Inter-Integrated Circuit (I2C), serial peripheral interface (SPI), controller area network (CAN), RS-422, and universal asynchronous receiver and transmitter (UART). There are drawbacks associated with each of the data busses, specifically I2C bus. If multiple devices are present on this data bus, and any one of these devices fail, the whole bus is inaccessible. This can become extremely detrimental to the entire mission if critical mission components are connected on this data bus. The Dellingr CubeSat mission faced a unique issue while using this bus. Problems were observed when sending commands to the reaction wheels. Moreover, other components which shared the same data bus were inaccessible as well. Due to this setback, the operations team turned off the reaction wheels to mitigate this problem [7]. This action did not cause a mission failure, as they had redundant actuators. However, it shows that possible mission failure events would have occurred if redundancy was not considered. As this would entail that the satellite would have been underactuated with no pointing abilities, leading to further subsystem failures.

1.3.3 CubeSat Failure Causes: Communication Subsystem

The third potential cause of mission failure concerns with the communication subsystem. A common issue is the loss of communication due to the inability of the onboard transceiver to toggle between transmit and receive mode, single event upsets, etc. It has been surveyed that roughly 17% of mission failure is caused by this subsystem [6]. Transceivers typically come with built in switches which are controlled by software commands to be either in the receive or transmit state. Known issues like logic errors are commonly noticed in this technology. An example of this occurrence is an infinite loop when switching communication modes. For instance, a satellite can only transmit and no longer acknowledges the telecommands sent from the ground station to

reinitialize its state. Hence, leading to an unexpected increase in power consumption caused by this subsystem. If the power generation is not able to make up for negative power budget margin caused by this problem, it can result in a complete satellite shutdown. Furthermore, an incorrect selection of the antenna could cause communication issues as well. Although majority of missions use omnidirectional antenna, some consider using a directional antenna to focus on a specific location. However, these antennas require accurate pointing while conducting operations to reduce the losses endured in a space-ground linkage. ASTERIA a 6U CubeSat launched in 2017 consisted of such a component. The CubeSat was able to maximize coverage by successfully pointing their satellite with their robust attitude control techniques [8]. Although these are pinned to be possible issues, it is noticed that often that missions fail to conduct "first contact" with the CubeSat once it is deployed off the ISS. Approximately 45% of failures [6] are due to this cause, as they could be "dead on arrival" in which there is a failure to boot up or because of the mentioned issues in the communication subsystem.

1.3.4 CubeSat Failure Causes: Attitude Determination and Control Subsystem

The four causes of CubeSat failures mentioned have been mitigated in the recent years by incorporating commercial off the shelf (COTS) solutions. Since 2005, the success rate has grown from 48% to 61% [9]. For example, various companies employ multiple safety measures for known concerns regarding over- and under voltage issues. Thus, component suppliers use protection circuits for the battery to protect its performance. However, the cause of failures in power generation, accurate pointing for satellite communication and issues with data busses are often caused by poor implementation of the satellite's attitude determination and control subsystem (ADCS). The development of ADCS is a challenging task due to the complexity associated with this subsystem. The ADCS utilizes all major satellite subsystems, which increases

the chance for a CubeSat failure. Roughly 5% of CubeSat failure occured due to ADCS. However, issues such as satellite pointing, power consumption, lack of power generation, etc., contribute to the percentages of the previously mentioned subsystems [2] [6]. The presence of a poor attitude estimator is a common reason why failures in ADCS occur. Specifically for attitude estimation, the sensors and attitude determination algorithms work together to produce knowledge of the satellite's orientation. Thus, the two possible courses of failures in this system often involve the operational code and the sensors used. Software issues such as sensor interpretation, calibration, and input issues can lead to large inaccuracies in attitude. Moreover, hardware issues leading to high-power consumption, poor dissipation of heat, degradation of component can affect the CubeSat's functionality as well. These problems with the ADCS can result in downstream catastrophic events across multiple systems, including lack of power generation, loss in communication, incorrect pointing, and misuse of payload applications. Poor attitude determination leads to poor control, as the attitude estimates cannot be trusted to reorient the satellite. If actuation is performed in the presence of this inaccuracy, power is consumed meaninglessly. Incorrect satellite pointing also takes place, which leads to the misuse of payloads for earth observation missions and communication issues caused by a misdirected onboard antenna. The mentioned issues are critical for amateur mission such as ESSENCE 3U CubeSat. In addition, poor attitude determination algorithm software can also cause difficulties while conducting satellite operations. Known and unknown anomalies lead to the inability to produce onboard attitude state estimates of any human-made satellite. Some issues involve parameter tuning and incorrect factors to compensate sensor bias. Thus, these constraints lead to setbacks to achieve mission objectives such as key experiments related to control theories and earth observation endeavour using optical payloads.

In addition to onboard issues, lack of attitude information leads affects satellite operations. Some mission objectives often require initial verification to proceed with operations plans, such as the ones defined for the ESSENCE mission. These initial verifications include knowledge of orbit location and satellite orientation. If not verified, key objectives such as taking pictures of a target area is not attainable. An example of such events was seen in the active DESCENT CubeSat mission. The onboard ADCS software performed in an unexpected fashion, resulting in a loss of ADCS. In the DESCENT CubeSat, the only set of actuators were magnetorquers. When using this component, it is critical to ensure that the magnetometer is not in use while performing. This is because it forms local magnetic field which adds to the applied torque generated by the magnetorquers. Due to logic errors, the magnetometers could not be disabled, and attitude control was not possible. This led to the inability to use the optical payload shown in Figure 1.2 B, to observe a target area.

Due to the large spread of failures that are commonly caused by poor attitude determination implementation, it was chosen to be the focus of this thesis. This topic is found to be often reinvestigated to give amateur missions purpose. However, it is equally important to utilize such amateur mission to allow new satellite developers to gain valuable experience by simply focusing on a functional algorithm to produce accurate attitude estimates. In order fulfill this goal, there is a need to develop a standardization for attitude determination specifically for CubeSats.

1.4 Analysis of Common Practices and Standards used in Engineering Domain

To minimize the failures caused by poor attitude determination implementation in CubeSats, there is a growing need for a definite standard for this subsystem. To develop such standard for CubeSats, various engineering standards and common practices were explored. Five such practices and standard are studied and are outlined as European Cooperation for Space Standardization

(ECSS), PC104 Standard, CubeSat Deployers and Onboard Software Development Kits (SDK). . The standards discussed in this thesis provide procedures in an open ended fashion. Although there are benefits associated with them, they are frequently interpreted differently depending on the satellite developer. However, CubeSat developers are often students who lack experience in satellite development. Thus, a complex subsystem such as attitude determination requires a standard that is not trivial. This would lead to benefits such as lower percentage of CubeSat failures caused by ADCS and provide valuable knowledge for new satellite developers in the ADCS domain.

1.4.1 European Cooperation for Space Standardization (ECSS)

Standardization for satellites has been of interest since the introduction of the European Cooperation for Space Standardization (ECSS) by the European Space Agency (ESA). This unique standard consists of four branches used for the development of a successful mission. These branches are often modifiable based on the system at hand and are defined as [10]:

1) Space project management
2) Space product assurance
3) Space engineering
4) Space sustainability

These branches are defined with the collaboration of the Consultative Committee for Space Data System (CCSDS) standards [11]. These standards are implemented across all satellite-based missions due to their ability to have viable options to derive appropriate solutions to a given problem. The project management branch provides a thorough description and guidelines on how to manage a spacecraft mission. It is distributed in five disciplines which provide general guideline on key project management related activities. These include the planning and defining key

milestones required to be met at various design review stages, such as preliminary design review (PDR) and critical design review (CDR). In addition, key mission factors such as cost management techniques have been discussed. This is extremely crucial for CubeSat mission as one of the mission constraints that drive most decisions is cost. If proper mitigations to potential risks are not considered, they act as a catalyst for failure in any sort of space mission. The second branch pertains to the efforts made to meet goals associated with the system at hand in terms of product assurance. The seven disciplines that fall under this branch analyze potential failures, safety and appropriate regulations for various materials used for satellite development, and quality assurance. Any materials or components that are not COTS implemented in satellite design are required to pass regulatory tests. These tests include vibration, anechoic chamber, thermal vacuum chamber, etc. In addition to the satellite hardware, software testing is also important as defined in this ECSS branch. Testing strategies from the systems/software development life cycle (SDLC) are enhanced, to ensure proper functionality and security measures are taken depending on the mission's class. The space engineering branch one of the most important disciplines to ensure the mission is completed successfully. There are seven disciplines defined in this branch that incorporate all the major engineering domains. These include systems, electrical, mechanical, software, communication, control, and ground systems engineering. As of May 14^{th,} 2021, new handbooks are in production to help support the mentioned engineering disciplines. Concepts such as requirement formulation and verification, concerns revolving around space environment, attitude, and orbit control solutions, etc. are outlined in this branch. The final branch in this standardization is comply to the concerns related to space debris. The Kessler syndrome is defining collisions occur in space due to existing debris as the density of objects in the low earth orbit grows over time [12].

Similar to the ECSS, National Aeronautics and Space Administration's (NASA) systems engineering handbook also provides useful guidelines to develop satellite of all scales. However, both sources focus only on large missions rather than CubeSats. Such missions include ESA's Sentinel Missions, and Alphasat mission, who do not have strict cost and design constraints. However, these resources have helped coin and improve existing standards and services such as PC-104 stack and software development kits, which are used for CubeSats.

1.4.2 PC104 standardization

PC104 standard was first introduced in 1992 by the PC/104 Consortium and is used across several industries [13]. As the name suggests there are 104 total pins used among 2 connectors each and are found on PCBs that form "stacks". The main purpose behind this standard is its ability to fit in small systems. Unmanned Aerial Vehicles (UAV) often make use of this unique standard. Military forces across the world make the use of this technology due to its advantages revolving around power, size, and computation. The MQ-50 Hunter and Predator make the use of this standard, as they can handle the processing requirements associated with such systems [14]. Another industry which has taken advantage of this standard is its usage in robotics-based applications. A study conducted in 2000 demonstrated the use of this standard as it was small enough to handle the complex applications, such as interfacing among multiple onboard subsystems [15]. A recent use in the LEMUR 3 four-limbed robot makes the use of this standard as well. This robot is used in extreme terrain environments on earth to analyze sedimentary rocks. The PC104 stack was used to connect other hardware to the main board consisting of CPU module [16]. Applications such as autonomous planning behaviours using Real Time Operating System (ROS) was implemented using the PC104 stack as well [15]. The use of this standard is of high interest in other environments such as Mars missions.

Apart from the unique use of the PC104 standard on earth-based applications, it is commonly utilized in CubeSat applications. York University's DESCENT and ESSENCE CubeSat missions make this use of the PC104 stack to meet the electrical, mechanical, and cost constraints associated with these missions. This is where PCBs are enhanced as they can take the advantage of Micro-electromechanical systems (MEMS) based technology. These outputs of such MEMS components are easily delivered to appropriate inputs with the use of the PC104 stack. A noteworthy advantage of using the PC104 stack in the nanosatellite domain., is the reduction in need of wires. This is crucial as it allows the systems engineering team to generate robust solutions to their space-based applications. Figure 1.3 shows the use of the PC104 stack used for the burn wire mechanism for the deployable Ultra High Frequency (UHF) antenna. Thus, it is shown that using such a standard provides benefits so highly constraint systems such as UAVs and CubeSats.

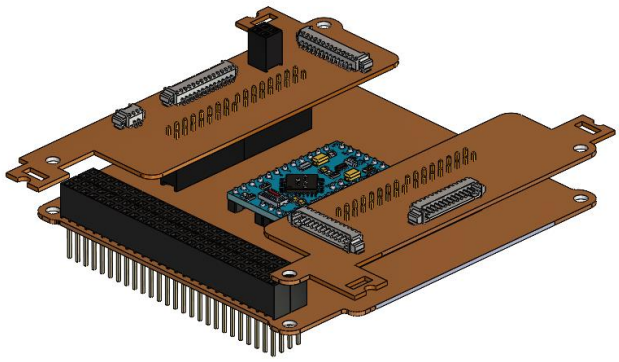


Figure 1.3: PC104 stack used for a custom board used for ESSENCE Burn Wire Mechanism via SolidWorks.

1.4.3 CubeSat Deployers

In addition to the PC104 standards, the Poly-Picosatellite Orbital Deployer (P-POD) was introduced by California Polytechnic State University [17]. The primary reason behind this deployer is to reduce potential catastrophic events detrimental to the launch vehicles. Apart from

the safety concerns, it is highly beneficial as the process of launching CubeSats is easily streamlined. For missions involving 1U-CubeSats, the initial P-Pod could hold up to 3 CubeSats, optimizing the space in the launch vehicle. Several CubeSat missions such as the MEROPE 1U and DESCENT CubeSats were launched using the P-POD. Since its first use in the early 2000s, companies such as NanoRacks have developed their own pod as a service. The NanoRacks CubeSat Deployer (NRCSD) shown in Figure 1.4 shall be used for the ESSENCE CubeSat mission. There are several requirements associated with the NRCSD, described in the NanoRack's IDD. This document allows the verification of all compatibility needs of the deployer along with the ISS flight safety requirements [5]. These measures efficiently avoid possible alarming events like battery leakage and short circuits. Therefore, the CubeSat deployer standard is well accepted in all CubeSat missions as it provides launch providers efficient means to launch small satellites.

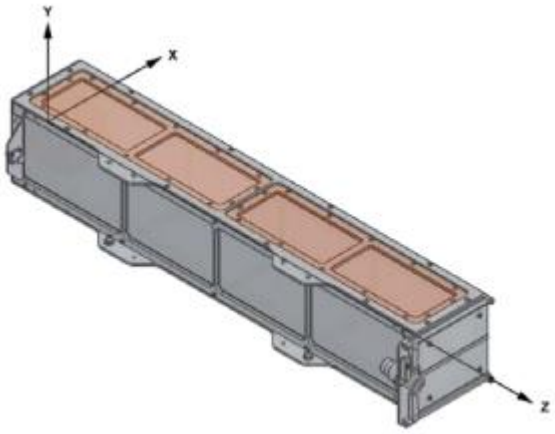


Figure 1.4: NanoRacks CubeSat Deployer used for DESCENT and ESSENCE missions [5].

1.4.4 Onboard Software Development kit

Onboard software of any satellite mission is one of the most complicated endeavours, as the objective of each mission varies. However, CubeSat missions are developed mainly for educational, technology development and scientific purposes. Thus, several missions are

developed by amateurs that lack software development experience. To help solve this issue, several companies such as GOMspace, Bright Ascension, EnduroSat provide end users SDKs to develop their satellite's onboard software. These kits are comprised of libraries, frameworks, and many other useful integration management techniques. In addition, SDK comes with compatibility guidelines which help systems engineers chose appropriate components that are suitable and compatible. However, most SDKs are very robust as they allow the use of various communication protocols, making them compatible with multiple components such as Global Positioning System (GPS) receiver, attitude sensors, etc. The DESCENT CubeSat mission used the Bright Ascension SDK [18] for onboard operations, and ESSENCE utilizes the GOMspace SDK [19]. Both solutions are proven to be acceptable SDKs for CubeSat missions as they have high TRLs values and have operational integrity. This is not defined as a formal standard used for CubeSat missions but serves CubeSat developers greatly, as it provides an easy medium to integrate numerous components.

1.5 Research Question

Upon exploring various relevant engineering standards and common practices, the importance of the development of attitude determination standardization for nanosatellites is evident. In section 1.2, various causes of CubeSat mission failures were discussed and analysed. The causes of failure showed a common point which pertained to the poor development of attitude determination and control system. Attitude control is highly dependent on good estimation of the current attitude/orientation of the CubeSat. Earth observations missions such as DESCENT displayed the effects on the system if attitude estimates are not produced. Hence, the research question this thesis seeks to answer is:

"Is there a path to standardize attitude determination for organizations developing amateur nanosatellite missions?"

1.6 Research Objectives

To help answer this research question, this thesis investigates the various attitude estimation methods. There are two categories of attitude determination methods, static and online. The methods explored are include TRIaxial Attitude Determination (TRIAD), Davenport's Q-method, Quaternion Estimator (QUEST) method, and Multiplicative Extended Kalman Filter (MEKF). Furthermore, a system analysis to gauge the performance of online attitude determination technique is also conducted. For the thesis the orbital regime is constrained to LEO and there is no fast or precise attitude control performed in the simulated testing environment.

Thus, the following objectives are defined to produce attitude estimates and answer the research question:

1. Develop and analyze orbit determination techniques using MATLAB/Simulink
    a. Generate orbit propagation using Simplified General Perturbation model to produce satellite position and velocity estimates
    b. Develop various coordinate transformation matrices for Earth centered refence frames
    c. Generate orbit propagation via Kalman Filter using GNSS receiver to produce satellite position and velocity estimates
    d. Compare outputs using orbit analysis software called Systems Tool Kit (STK)
    e. Generate models to determine sun and magnetic field vectors for LEO
2. Analyze accuracy levels of attitude estimates using static attitude determination algorithms using a satellite simulator using MATLAB/Simulink
    a. Develop optimized TRIAD algorithm to produce DCMs using dual vector observations

b. Develop Davenport's Q method algorithm to produce quaternions using dual vector observations
c. Develop QUEST algorithm to produce quaternions using dual vector observations

3. Analyze accuracy levels of attitude estimates using online attitude determination algorithms using a satellite simulator using MATLAB/Simulink
    a. Represent attitude profiles using quaternions
    b. Develop MEKF using rigid body dynamics and quaternion kinematics-based system model
    c. Develop MEKF using only quaternion kinematics-based system model
4. Conduct a system analysis under several operational scenarios
    a. Analyze results upon single sensor and multiple sensors failure
    b. Analyze results upon misalignment of optical gyroscope
    c. Analyze results upon misconfiguration of coarse sun sensors
    d. Analyze results using various operational frequencies
5. Develop and analyze asynchronous sensor fusion algorithm
    a. Implement Zero-order hold algorithm for sensor fusion
    b. Implement Frequency Update based sensor fusion
6. Validate online attitude determination algorithms using an inertial measurement unit
    a. Utilizes IMU sensor for both system models proposed for MEKF
    b. Compare results between both MEKF system models

1.7 Thesis Layout

This thesis consists of nine chapters that accumulate and elaborate on the work conducted to meet the objectives defined and answer the proposed research question. Chapter 1 outlines the research

domain in which this thesis will explore. Research was conducted where known issues of CubeSats was defined and how it relates to poor attitude determination practices. This leads to the formulation of the research question that is answered throughout this thesis. Chapter 2 consists of the literature review of various attitude estimation techniques that are common for all satellite missions. Moreover, sensor fusion algorithms are explored to deal with the asynchronous nature of attitude sensors. All these methods would be analyzed and discussed throughout the remaining parts of the thesis. This would lead to concluding what algorithm is suitable for the standardization of attitude determination. In chapter 3, a thorough description of a complete attitude determination system is explained, along with relevant equations and coordinate frames. Chapter 4 describes various sensors which are used on the ESSENCE CubeSat and their sensor models are illustrated and described. Moreover, chapter 5 dives into the orbit determination portion required to produce useful vector quantities used for attitude estimation. There will be 2 methods explored with uses the generic simplified general perturbation orbit model and a Kalman filter which incorporates the GNSS receiver for satellite position and velocity estimates. Chapter 6 introduces the static attitude determination algorithms and discuss the results. These algorithms include optimized TRIAD, Davenport's Q method and QUEST. Chapter 7 provides an analysis for the online attitude determination algorithms using MEKF, along with the sensor fusion algorithm. Chapter 8 performs a hardware in the loop experiment which analyzes both MEKF system models in a real-world scenario. Chapter 9 communicates the concluding remarks, contribution of the research and future work. Finally, the thesis concludes with all the references used throughout this thesis.

CHAPTER 2 LITERATURE REVIEW

The purpose of this chapter is to explore various attitude algorithms used in various missions. It was established that the need for attitude determination for nanosatellites is crucial for mission success. In the processes of developing an acceptable standard for attitude determination, six methods will be explored and analyzed. The development of conducting attitude estimation can be categorized in two methods, static attitude determination, and online attitude estimation. The following methods are reviewed in this chapter and further analyzed in this thesis:

1. Optimized TRIAD based attitude determination
2. Davenport's Q method-based attitude determination
3. QUEST based attitude determination
4. MEKF based attitude determination

2.1 Optimized TRIaxial Attitude Determination (TRIAD) Method

TRIaxial Attitude Determination (TRIAD) method was first introduced to the space industry by Harold D. Black in 1964 [20]. The algorithm was implemented on the US Navy's NAVSAT/Transit satellite system that provides positional knowledge of ballistic missile submarines and conducts hydrographic and geodetic surveying [21]. These series of satellites made use of a simplified and low-cost approach to dictate the attitude of the given rigid body. The unique concept of the TRIAD utilizes normalized vectors in two frames: the body and inertial reference frame. For satellite missions, the inertial frame is often referred to the Earth Centered Inertial (ECI) Frame at J2000 epoch. Using various orthogonal relationships between vectors defined in each of the respective frames, a Direction Cosine Matrix (DCM) is formulated to define the attitude profile

of a satellite. The vector pair used for this method are the Sun position and the Earth's magnetic field vectors. These vectors are chosen for static attitude determination because they are well defined in the Earth Centered Inertial (ECI) frame, making it a reliable reference point. To improve the accuracy of this method, a weight factor is incorporated to reduce noise and produce an optimal solution. This factor is the noise variances of the onboard sensors used to generate the mentioned vectors while in orbit. This method was demonstrated in modern missions such as Boeing's nanosatellite mission called CubeSat Test Bed 1 (CSTB1) [22] [23]. The mission consisted of low-powered components making the restriction on its computational abilities at its peak. This forced the satellite developers to implement a low-cost solution for attitude determination. Thus, the mentioned optimized TRIAD algorithm was used for this 1U CubeSat mission. Due to the chosen pair of vectors, sun sensors and magnetometers are incorporated to make use of this methodology. These sensors are one of the least expensive sensors as they are coupled with another large component. For example, magnetometers are often embedded in the onboard computer (OBC) and coarse sun sensors in the solar panels. Furthermore, this mission was launched in a specific type of orbit, called the sun-synchronous orbit with an altitude of 750 km [22]. Since this a static attitude determination method, the unavailability of sun sensors significantly affects the solution's accuracy levels. The paper detailed the sun's sensor unavailability in which error is seen to grow continuously while the satellite is in eclipse. This issue is held a variable importance level because it is mission dependent. CubeSat missions are often seen using an optimized TRIAD method for an educational objective, which do not require accurate attitude profiles at all times. However, larger missions which require high amounts of attitude maneuvers are less likely to use this solution. The error values observed in the CTSB1 mission was approximately ranged from 2 degrees to 8 degrees in the Roll, Pitch and Yaw axis.

In addition, the GT-1 CubeSat that is currently being developed by the Georgia Institute of Technology explored this option as an attitude estimation algorithm as well [24]. Similar to CSTB1, this also is a 1U CubeSat, but was placed in the same orbit as the ISS. Both missions utilized a 6-sun sensor suite, one per face for attitude determination. The key difference between the TRIAD method used was that it did not seek to optimize the sensors to reduce the noise inputted into the system. This resulted in a high attitude error output in the Yaw axis. It tends to draw in a sinusoidal fashion, in which the amplitude increases when the satellite is in eclipse. The eclipse period on average for LEO satellites is on average approximately 17 mins, making this attitude determination method far too sensor dependent. The optimized TRIAD approach aids with the error issue when the satellite is entering penumbra. Both papers demonstrated the importance of utilization of sensor suites specific to the mission.

Larger missions such as the Small Astronomy Satellite (SAS) and Seasat made use of the TRIAD method as their primary means to produce attitude estimates [25]. However, another application was introduced for this method where it gets used for the initialization of vectors for an online method. SERPENS, a Brazilian CubeSat mission incorporated the TRIAD algorithm with a Kalman filter [26]. It was used to initialize the filter as an attempt to improve accuracy of attitude estimates. However, this is unnecessarily computationally extensive. This is because the Kalman filter is designed to be capable of converging to the growth truth, even with bad initial estimates. Although the drawbacks of this mission are sensor availability and dual observational capability, it does provide adequate attitude estimates. Due to large amount of uses of this filter, it was deemed to be a suitable option to explore in this thesis. For amateur CubeSat missions such as ESSENCE, it is a noteworthy solution to explore as it builds knowledge for amateur satellite developers. Moreover, this method was utilized for an offline attitude estimation for the DESCENT mission

due to ADCS software failure. Therefore, it also proposed to be a backup solution in the case of attitude estimation failure for the ESSENCE mission.

2.2 Davenport's Q Method

In 1965, a well-regarded mathematician Grace Wahba developed a loss function which is used to further develop the original TRIAD algorithm [27]. This loss function was used as another avenue to optimize the TRIAD method and avoid potential singularities. As mentioned, the optimized TRIAD algorithm focuses on providing importance to sensors with a lower error variance. This is simply to ensure that the more reliant sensor is given preference. However, the loss function used by the Davenport's Q-method seeks to use a variation of the least square approach to provide a solution. The solved quantity avoids DCM and uses a 4-dimentional parameter called the quaternion. This method shall also utilize the sun and the magnetic field vectors to form quaternions. Paul Davenport's sole purpose was to provide an accurate estimate with no risk of a potential occurrence in singularity. The Wahba's problem is a general loss function that is to be minimized. When solving using this method, the loss function is interpreted to be an eigenvalue problem that leads to develop a gain function. Thus, the goal changes from minimization of function to maximization, as that would entail the loss function is minimized. The function is said to be maximized when the eigenvalue solved is the largest amount the solved eigenvalues [27] [28]. This innovative method was incorporated on the series of NASA's HEAO missions [25] [26] [29]. The attitude requirements for these missions were not very intense, due to the low pointing needs of these series of satellites. This method was able to provide sufficient results as the accuracy of attitude determination was at an arcsecond or 2.7×10^{-4} degrees [30]. The level of accuracy is high due to large amounts of accurate sensors were used for the ADCS of these series of satellites. In addition to large satellite missions, this technique is explored in CubeSat missions such as the

previously mentioned GT-1 CubeSat. It was implemented to demonstrate the functionality of static attitude determination specifically, quaternion-based estimators for nanosatellites. The simulated tests done for the GT-1 CubeSat provided concerns due to the periods the CubeSat is in eclipse, but this error is expected as the sun sensor suite is unavailable during that period [24]. However, this method's performance was shown to be slightly worse than the Optimized TRIAD method. The analysis conducted for GT-1 CubeSat showed an approximate 3-degree increase [24]. However, the error was improved by altering the variance values are it is simply the weight factor being altered in the loss function. This led to the same accurate levels as the TRIAD method discussed in this paper.

Although the accuracy levels are similar to the optimized TRIAD method, this algorithm is considered to be one of the most robust solutions. This is displayed with its use in application such as Unmanned Aerial Vehicles (UAVs), that have incorporated this algorithm. Cranfield University conducted a comparison of Gauss Newton's method and Davenport's Q method, to dictate which method will be used for attitude determination [31]. Although the system at hand considers vision sensors as that is an affordable approach from both financial and engineering perspective. However, similarities are drawn to CubeSats because of the size constraints associated with UAVs. The paper encouraged the use of quaternion-based attitude determination due to the low computational needs. Using DCM results in high computational needs as there are nine parameters to update, where a quaternion only updates four. Based on the simulated results, the paper conveyed that the Davenport's Q method is much more accurate [31]. A small drawback consists in this method, as the algorithm would not produce a unique solution if there were two largest eigenvalues as solution [27] [31].

As mentioned, the TRIAD algorithm requires a pair of vectors to be defined, but this is not an issue

for the Q-method. This is because the loss function updates through one vector at a time, where the TRIAD algorithm is dependent on the availability of two vectors. However, a drawback with the Davenport's Q method is it's computationally more intensive than the TRIAD method. The loss function is the reason for that, as it uses a recursive process to solve the eigenvalue problems. Due to the robustness of the Davenport's Q method, it is to be explore and compare with other methods to meet the thesis objectives and answer the research question.

2.3 Quaternion Estimator (QUEST) Method

Davenport's Q method was seen to be a robust solution and a singularity free solution, as it was explored across domains outside of space as well. The method found success in the HEAO missions as the operating frequency set for attitude determination was not very high. Due to the high computational needs of the algorithm, an efficient one was of need for modern missions. MAGSAT was one of such mission which had mitigate the efficiency issues. This initiated and displayed the first use of the Quaternion Estimator (QUEST) method in the MAGSAT mission in 1979 [27] [32].

The QUEST algorithm is intended to focus on an approximate solution rather than compute an exact solution of the attitude problem. This is where the solution deviates from the Davenport's Q method. This method grew popularity due to its efficiency in solving for the attitude quaternion. The introduction of a threshold value to produce an approximate solution led to low computational needs an efficient solution [27] [33]. Upon defining an appropriate threshold value which within the $\sim 10^{-7}$ range, is expected to reach similar accuracy levels as the method proposed by Paul Davenport. Malcom Shuster who implemented the QUEST on MAGSAT, confirmed that if the attitude information is accessible, then this approach will meet its purpose [26]. The threshold defined is utilized in the Newton-Raphson method that numerically solves for the eigenvalue. The

solution shows another deviation where a numerical approach is employed rather than an analytical one. An efficiency test was conducted where QUEST was concluded to be 1000 times faster than Davenport's Q method [26]. Although this solution is far more efficient, but it has its faults because numerical solution provides an approximated quaternion.

MAGSAT is a LEO satellite which was launched to survey and analyze the magnetic field around the earth [34]. This aided the early development of the magnetic field models such as the basic dipole model that treats Earth as a large bar magnet. Due to the nature of the orbit, sensors such magnetometers, sun sensor and gyroscope were present on the satellite's ADCS [34]. The paper further defined a unique approach taken to improve the sun vector approximation by combining the readings from star tracker and the sun sensor. The mission was able to meet the 20 arcsecond or 5.56×10^{-3}-degree error requirement aimed to achieve using the QUEST [34]. However, attitude error grew when sensor failure or unavailability occurred in this mission as the magnetic field estimates grew larger than 20 nT [35]. This is expected because sensors that are periodically unavailable cause an increase in error. Furthermore, it increases the computation time as fewer vectors are used for attitude correction.

QUEST method's success from the large missions, led to its usage in CubeSat missions as well. In which various use cases were introduced such as a combination of QUEST and Kalman Filter, specific pointing mode using QUEST, etc. CAT-2, a 6U CubeSat mission and a research project by Universitat Politècnica de Catalunya displayed a successful usage of QUEST [36]. However, CAT-2 only utilized this algorithm for a specific attitude determination mode called the Sun safe mode [36]. The use of QUEST provided benefits such as efficient solution as it was used for contingency. Unlike the nominal mode which utilizes several sensors, this method only incorporated the six coarse sun sensors and a magnetometer. This led to an expected increase in

inaccuracy as the nominal mode employed far more accurate sensors. Hence, forcing the CubeSat developers to utilize this method as a contingency method. Moreover, this algorithm is beneficial from the power perspective, as it is proven by its use in the CAT-2 mission [36]. Furthermore, QUEST was also implemented in the 2008 COMPASS-1 CubeSat mission by the FH Aachen University of Applied Sciences [37]. However, a drawback was experienced in which the ADCS of this CubeSat was under jeopardy. Thus, making the mission a prime example why static attitude estimators are not considered as primary options for determining attitude. The COMPASS-1 mission reached a point where attitude control was not attainable due to the lack in precision from attitude determination [37]. As discussed, attitude control is highly dependent on the onboard attitude determination algorithm. Although the mission was not terminated, but poor attitude estimates contributed to the possible issues that were experience such as loss in communication, power generation, etc. The currently active DESCENT CubeSat consisted of multiple software issues which led to the inability to produce attitude estimation. This problem was mitigated by using the QUEST method that led to quaternion values, describing the attitude state of this satellite. However, there are benefits still associated with this method as it provides the quaternion estimates at a rapid rate when compared to other static estimation methods.

In addition to CubeSats this method is used to produce an attitude matrix directly through the sensor. A star tracker project called CubeStar intended to be operational on nanosatellites, made use of the QUEST algorithm [38]. The algorithm produced attitude matrix, which relates the sensor's body frame to the inertial frame. Moreover, the paper solidified the notion of low computational needs of this algorithm as the execution time was <5ms when in operation [38]. Moreover, the paper discusses various tests conducted to communicate the effectiveness of this solution. Through these tests the quaternion solution accuracy is found to be at the arcminute or

0.017 degrees. Several other various of QUEST were developed such as the Estimator of the Optimal Quaternion (ESOQ) and Fast Optimal Attitude Matrix (FOAM). However, it was found there was no significant increase in accuracy when compared to QUEST [39]. Thus, QUEST would be investigated along with the other two static methods algorithms mentioned.

2.4 Multiplicative Extended Kalman Filter (MEKF)

State estimation is one of the most useful concepts used across several different domains. Applications include pose estimation for robots, orbit determination for satellites and attitude determination of any rigid body, etc. Several attitude estimation techniques have been used in the past such as complimentary filter, H-Infinity filter, particle filter, Unscented Kalman Filter (UKF), etc. One the main drawback of these methods is the need of high amount of computation required to produce attitude estimations, as discussed in [40]. For highly constrained mission such as CubeSats, a Multiplicative Extended Kalman Filters (MEKF) have been proven to be one of the most sought-after solutions. The filter consists of 3 major steps initialization, propagation, and measurement update/reset. This is further explored in this thesis through simulations and hardware-based experiments. A prime reason in using MEKF is its ability to easily conduct sensor fusion, minimal computation and modular.

Prior to MEKF, an additive approach called an Additive Extended Kalman filter (AEKF) was used for most applications which dealt with non-quaternion centric algorithms. The measurement update step is where the issue stemmed from while developing state estimation for attitude determination. As described, there are three factors which justify the lack of usage of this form of filter [27] [41]. The first problem described in this paper deals with the unit quaternion constraint. The technical analysis conducted described that due to the restriction of the probability distribution, it led to ignoring the describe constraint [41].This essentially implies the solution for

the attitude estimate is restricted [27] [41]. Hence applying this technique defeats its purpose to avoid the issues faced such as “Gimbal Lock” while using DCM. The second reason pertains to the rise of state errors at the cost of higher than required computational time. This occurs when updating the vector component of the state quaternion is negligible, leading to a covariance matrix consisting of zeros. This affects the continuous solution as it implied the filter trust the system model completely, hence neglecting the onboard sensors. The third issue addresses the use of error quaternion while using AEKF. The error quaternion represents the small angle approximations when rotations take place to update the quaternion state. However, the AEKF cannot use the error quaternion because this parameter cannot be used to form the covariance. This leads to a singularity in the covariance matrix, making the solution unattainable.

A more robust approach was introduced when updating the estimated quaternion, MEKF. This method uses the quaternion multiplication property to meet the quaternion constraint requirement, to avoid the issues found in AEKF. Moreover, MEKFs have an advantage over the static attitude determination techniques, as they can always produce information. Static attitude determination methods such as Optimized TRIAD, Davenport’s Q method and QUEST are not able to produce appropriate solutions when measurements are unavailable. This problem takes place when a satellite enters an eclipse or experiences potential sensor failure. Early amateur missions such as the AAU CubeSat mission implemented MEKF with a combination of the static estimation techniques [42]. In this thesis, two system models are compared when forming the MEKF.

2.4.1 MEKF- Dynamics and Kinematics based System Model

The system model implemented for the initial development of MEKF considers the dynamics and kinematics equations relevant to the rotational motion expressed in [43], [44] and [45]. These equations are the Euler’s equation for rigid bodies, and quaternion kinematics equation. This

system model is very commonly used for nanosatellite and satellite constellation missions as shown in [43] and [46]. Both sources focus on the use of the sensors utilized on the ESSENCE 3U CubeSat. Using this model, MEKF's effectiveness was shown by comparing its outputs to the AAUSAT3 quaternion estimates [43]. It was found that the quaternion estimates were approximately at the same accuracy levels as the AAUSAT3's ADCS algorithm. Moreover, a recent analysis of this system model was conducted on the RAX-1, O/OREOS and DESCENT missions [47] [48]. Although RAX-1 and O/OREOS contained sun sensors, the analysis utilized the current measurements from the solar panels to derive the sun position vector in the body frame [47]. This approach is common to meet budget constraints, but due to the noise associated with analog-to-digital convertors (ADCs) accuracy levels tend to vary. This was confirmed as the accuracy levels were found to be ~5 degrees across all axes when only using the sun vector [47]. This error can be controlled by using more sensors, but it was shown that the error is at a controllable amount when not in eclipse. As the previous source considers solely sun vector-based updates, analysis was also done with magnetic field vectors as well. Research done focusing on attitude estimation in the LEO environment, where magnetometer based MEKF was evaluated [49]. Simulation and hardware in the loop tests were done to test the filter's performance. It was found the error in the attitude estimates were found to be around 8.5 deg [49]. LEO satellites have the advantage of using magnetometers as opposed to higher orbits, as the strength of the Earth's magnetic field weakens. One of the crucial things to consider for magnetometers is the sampling time if the magnetorquer is to be used for actuation.

In addition, large satellite missions such as MSAT mission displayed the use of this model as well [40]. The sensors used for this mission are horizon sensor, sun sensor, magnetometer, gyroscope, and star tracker. The angular error was within 0.01 deg, showing the benefits of using highly

accurate sensors such as star trackers. However, these aren't common on CubeSat missions because of the geometric and the cost constraints associated with the star tracker.

Although this method is effective and robust for missions across all domains, there risks still present in the MEKF. Risks include sensor failure and large dependencies on the initialization of parameters such as moment of inertia matrix. The moment of inertia matrix tends to vary depending on the satellite's mass distribution. An example is the use of thrusters which consume fuel, causing a change in the satellite's mass. Another example of this issue is seen in the DESCENT mission, which consisted of a tether deployment experiment [48]. Upon separation, the moment of inertia matrix varies as the tether separating the Cube affects the off-diagonal terms.

2.4.2 MEKF-Kinematics based System Model

As research progressed, an interest in quaternion kinematics-based system model grew as it showed various benefits. The tradition approach has its advantages, but they can be improved by incorporating reliable sensors to reduce the risks associated with the dynamics and kinematics based MEKF. For this technique, gyroscopes are the required inputs for state propagation step. This is because the quaternion kinematics equation requires angular rates to produce a time varying quaternion quantity. The intention behind using the new model is to reduce the computational needs and decrease the state vector to improve efficiency and accuracy of the quaternion estimates. For the ESSENCE mission the sensors of interest are sun sensors, magnetometer, and gyroscopes. These sensors are very common for attitude determination as they are of low cost. The initial system model discussed in section 2.4.1 updates nine states to implement the MEKF. However, the kinematics-based system model contains six.

For CubeSat missions, this is an efficient way to produce attitude profiles. Using the kinematics only system model would allow CubeSats to perform other operations easily. Causes of CubeSat

failures such as high-power consumption, lack of testing etc. would be minimized by using this technique. Missions that are meant to perform various control experiments such as ESSENCE and DESCENT, this system model would allow the computation budget to be in their favour. Development of this algorithm is explored in the upcoming NJUST-2 CubeSat mission [50]. In which two variations of the MEKF that focused on multiple star trackers were analyzed. Both algorithms use the kinematics-based system model is used to determine attitude estimates. The solution in the paper that used two-star trackers, concluded the accuracy was approximately at 0.02 deg across all axes. The second model adds another star tracker and had attitude error of 0.007 deg, which is expected as star trackers are very accurate sensors. This was further generalized that using this the kinematics only system model is viable option for all small satellite missions [51]. The analysis used all the standard attitude sensors which showed confirmed the accuracy levels shown if only star trackers were used.

An important deviation from the first system model is seen in the state transition matrix, as this model relies more on the sensors. The traditional MEKF implemented in [52] uses the gyroscope sensor as input to the state transition matrix. In which the bias in the measured angular rate is removed in the state propagation step. Hence leading to an improvement in the state accuracy levels as time progresses. However, a more mathematically viable approach is taken in [53] in which the state propagation is done solely based on the quaternion kinematics equation. Neither one of these methods are incorrect, as results are present to display acceptable attitude estimates when compared to true quaternion. Therefore, the evaluation of this model conducted in [52] and [53], provides an assurance in the capabilities of this system model. In addition to NJUST-2 CubeSat, University of Victoria's ORCASat mission intends to utilize this method for attitude determination as well [54]. Using such a unique system model is not limited to nanosatellite

mission, as this can be used to describe the rotational dynamics in any system. The research shown in [55] explore attitude estimation using IMUs, where estimates and the covariance converged to produce lower inaccuracy error values.

2.4.3 Asynchronous Sensor Fusion

The development of an appropriate attitude algorithm is very important, but accounting for the asynchronous outputs of the sensor is needed as well. Many applications hold the assumption that the sensor sampling time is equal to the filter's timestep. For example, if the filter is expected to be operating at 10 Hz, then all sensors are to be available at that frequency rate. However, that is not the case in all scenarios. A common scheme to mitigate this issue was developed which made use of local filters. Local filters were made for each respective sensor that had its own respective system model [56] [57]. To accumulate the local filter outputs, a master filter was formed to operate at the fastest sensor's output rate. As per [56] and [57], the fastest sensor is the gyroscope, and the other sensors are updated using the local filters at their respective output rates. A key difference in [56] was the use of the fault tolerant logic which kept track of the integrity of all sensor outputs. Although this unique way was useful for missions with strict pointing requirements, the use of local filters make it redundant as the same solution is being computed in a multivariate fashion. An analysis conducted in [58] consisted of a star tracker and gyroscope fusion, where the star tracker provided an update faster than the gyroscope. The solution chosen was to simply update the algorithm with the gyroscope when the measurement is present or else the system model proposed in section 2.4.1 would propagate the state forward in time. The use of local filters causes high computation usage, it can easily be avoided by utilizing either a logic-based sensor fusion or zero order hold fusion technique.

The logic-based fusion is derived from the known output frequencies of the sensors and the filter's

operational frequency. A commonality is found between the set in which a flag is raised that communicates to the MEKF which sensor to update. The MEKF essentially skips unavailable sensors and uses the ones which are available at that time. This is one of two fusion techniques explored in this thesis, where the sensors available for the ESSENCE mission are examined. This logic is presented as an acceptable solution as the discussed fusion strategy for a ground vehicle and an UAV's attitude determination algorithm in [59] and [60]. A unique way of using the same technique is introduce as a "switch rule" was implemented in [61]. This was meant to deal with the known effects such additional measured linear acceleration values found in accelerometers. Moreover, MIST CubeSat mission analyzed this sensor fusion approach as well, where they simply updated the state based on the available sensor [62]. Therefore, it provided an avenue to implement this for this research and will be detailed further in the following chapters.

Another way to conduct sensor fusion algorithm is the zero-order hold technique. This is often found in control systems, embedded systems, etc. to handle sampling of data. It is a mathematical model with uses Taylor series expansion to hold the sensor output to match the MEKF's operating frequency. Zero-order hold was used for a digital pulse width modulator (DPWM), where a controller uses this sampling model [63]. It was found this technique was accurate for the proposed system, but the operating frequency should be kept at a low to reduce deviations. Various options are also suitable such as a first order hold as explored in [64] and [65]. Systems which require high sampling rates, higher orders sample and hold techniques are suitable. However, zero-order hold is acceptable for the ESSENCE mission as the attitude determination filers is required to be running at 10 Hz. A higher sampling rate could cause an increase in the failure percentage CubeSats. This is because ADCS tends to draw a lot of power and require high computation for amateur missions as experiments tend to more attitude based. Thus, choosing a relatively low

frequency when compared to non-space-based system such the PWM is acceptable. The ROBUSTA 3-A CubeSat mission integrates this model for sampling current and speed sensors [66]. Thus, it can be concluded that this as a viable option to be further analyze for this research.

2.5 Conclusion

The attitude determination approaches discussed in this section communicated their benefits and drawbacks. The use cases presented in this chapter, exhibit the vulnerability of all three static attitude determination techniques. The concerning factors related to static estimation methods include the risk of singularity, high computation requirements, inability to enhance the asynchronous nature of attitude sensors and restricted outputs of attitude estimation. Attitude determination conducted through static methods in various satellite missions and other systems discussed, indicate the risks are high if they are selected as the primary attitude determination algorithm. However, the online attitude determination techniques analyzed in this chapter seek to make all the drawbacks mentioned regarding the static estimators their benefits. Upon research, the MEKF algorithm was found to be a robust option as the use cases of this orientation algorithm is limitless. Satellite missions discussed in this chapter show it can handle the asynchronous nature of the sensors, and produce sufficient attitude estimates in vulnerable orbital phases, such as eclipse. Hence, hinting its superiority over the static estimation techniques. Moreover, the MEKF has the capability to handle two system models. This demonstrates the adaptability of the MEKF as it desires to produce acceptable orientation estimation of any system. This isn't obtainable if the Optimized TRIAD, Davenport's Q-method, and the QUEST algorithms are used. In conclusion, based on the finding of this literature review, the chosen approach as the primary attitude determination standard is the kinematics based MEKF.

Chapter 3 Overview of Attitude Determination and Control and Reference Frames

The purpose of this chapter is to provide a brief introduction to the ADCS for satellites. Components which form this subsystem are expressed and described. However, the focus is drawn towards the attitude determination as it relevant to answer the research question. In addition, coordinate frame used for attitude and orbit determination are elaborated. Moreover, the equations of motions are described where the dynamics and kinematics equations are outlined.

3.1 Overview of Attitude Determination and Control Subsystem (ADCS)

The subsystem that handles maneuvers such as target pointing, nadir pointing, etc is the Attitude Determination and Control Subsystem (ADCS). It is also referred to as attitude and orbit control system (AOCS). The system is categorized in terms of 4 subsidies: “Measurement Management”, “Navigation”, “ADCS” and “Actuator Management”, as shown in Figure 3.1. The measurement management is intended to manage attitude sensors utilized for attitude estimations. Some common sensors are gyroscopes, sun sensors, star trackers and magnetometer. The outputs from these sensors are manipulated and transformed into useful parameters. The navigation filter uses these parameters to generate a satellite’s orientation and provide inputs for actuation. The actuation uses reaction wheels, thrusters or magnetorquers to manipulate and change the orientation of the satellite. This is where the mentioned maneuvers are conducted for either experimental or payload operations purposes. To utilize the actuators and meet a satellite’s pointing needs, control laws are utilized. The ADCS uses the input from the measurement and navigation and feeds it to the actuation management to conduct attitude control.

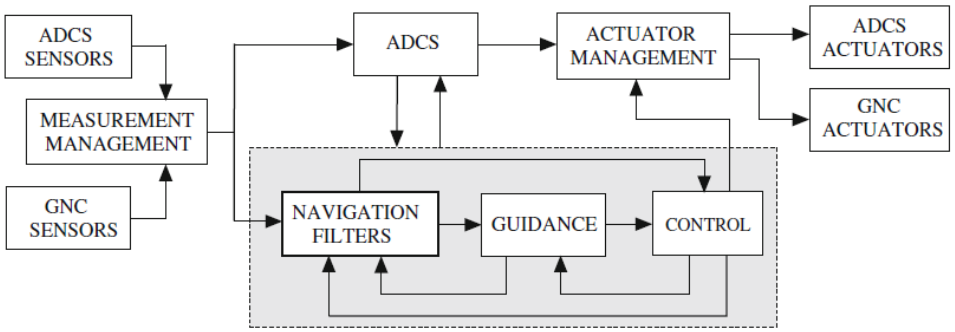


Figure 3.1: Summation of ADCS for a typical satellite [66].

As per the research motivation and literature review, it was communicated that there is a need for an attitude determination standard for CubeSats. Thus, the focus is more towards the attitude determination portion of AOCS. Although this is a rotational motion problem, but it is dependents on the knowledge of the satellite's orbit. Earth-based orbits have several environmental effects which alter the orbit over time. However, as the orbit gets farther away from the center body, these effects diminish. Some of the major effects that affect the nature of an orbit are prominent for satellites in low earth orbit (LEO). These effects result from Earth's shape and mass distribution and causes disturbance torques and other perturbating effects that causes variations in orbits. The knowledge of orbit is critical for magnetometers and sun sensors, because the used to model the vectors in an inertial frame require the satellite's position as an input. This information is then utilized for attitude determination after the derivation of the magnetic field vector. In addition to the mentioned drawbacks of a LEO satellite, there are benefits present as well. The Earth's magnetic field is detectable easily due to its strength. However, the further the orbit is from earth it is difficult to measure the magnetic field as the magnetic field strength diminishes. Moreover, it is one the most explored attribute of Earth as the use cases for the magnetic field is limitless. If the magnetometer is still used in a higher orbit such as a highly elliptical orbit (HEO), it is exposed to various internal noise sources. This causes a domino effect, where the result of attitude

determination are incorrect attitude estimates.

As mentioned, the need for orbit determination is equally required for attitude determination. Furthermore, the satellite's position is required to be altered through multiple reference frames at a certain epoch. Two possible solutions for orbit determination are introduced and explored in this thesis. These approaches are exploring a common orbit propagator called Simplified General Perturbation (SGP) model and GPS/GNSS based Kalman filter. Evidence will be provided throughout this thesis which confirms an accurate orbit determination technique between the two. Research has been conducted for GPS based attitude determination, but it is not commonly used due to high power consumption and complex calculations associated with the approach.

For both orbit and attitude determination, knowledge of various reference frames is necessary. Ideally, quantities such as sun position and magnetic field vectors are expressed in an inertial frame. To do various transformations are required from moving frames. However, Earth's not a simple rigid body as several effects are easily expressed in moving frame due their physical characteristics. A prime example of this is the Earth's magnetic field.

3.2 Coordinate Frames

Coordinate frames provide an insight to understand coordinate systems that compliments and helps derive concept of operations for the ADCS. The operations would include target pointing modes, performing experiments, etc. This section will dive into the reference frames used in this thesis and the rationale behind it. The mentioned frames are either geocentric coordinate systems or satellite-centric coordinate systems.

3.2.1 Body Fixed Frame (BFF)

Body Fixed Frame is a moving frame fixed to the satellite's body. This coordinate system

references the ECI frame when the satellite's orientation is described. The origin of this frame stems from the center of mass of the satellite. Because the thesis focuses on a CubeSat, the frame definition makes use of the principal moment of inertia axis, as the center of mass is used for the quantity. Figure 3.2 shows the frame definition of the body fixed frame and is commonly referred to as BFF throughout this thesis and is used as B in equations. Furthermore, the sensors inherently have their own frames which are required to be converted into BFF to conduct attitude determination.

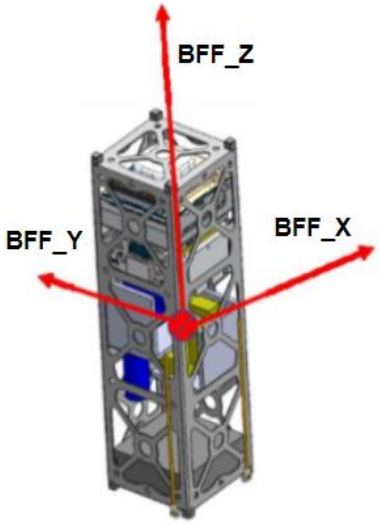


Figure 3.2: Body Fixed Frame for the ESSENCE 3U-CubeSat.

Satellite-centric Coordinate System:

- X-axis : CubeSat's principal moment of inertia X-axis.
- Y-axis : CubeSat's principal moment of inertia Y-axis.
- Z-axis : CubeSat's principal moment of inertia Z-axis.

3.2.2 Perifocal Reference Frame (PQW)

The perifocal coordinate system is a geocentric coordinate system and is used to describe a

satellite's motion. This system is an intermediate frame and is used due to its ability to process observations that describe the satellite's dynamics in an efficient manner. This coordinate system incorporates the concepts such as apogee/apoapsis and perigee/periapsis [67]. This is not an inertial frame but is dependent on orbital knowledge as the apoapsis and periapsis vary over time. This would further require transformation for orbit determination using SGP, to describe position and velocity vectors in an inertial frame for attitude estimation. This is commonly referred as PQW in this thesis and is described in Figure 3.3. The vectors described in this frame makes use of several orbital elements such as eccentricity, inclination, etc.

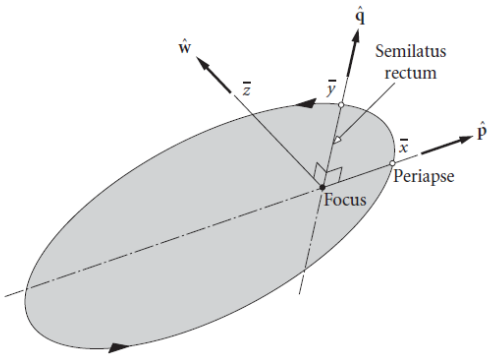


Figure 3.3: PQW Geocentric Coordinate system [68].

Geocentric Coordinate System:

- X-axis (P) : Coincident with the orbit perigee.
- Y-axis (Q) : 90^{o} from the X-axis in the direction of the satellite's motion.
- Z-axis (W) : Normal to the orbital plane.

3.2.3 True Equator Mean Equator (TEME) Frame

True Equator Mean Equator (TEME) is an earth-based coordinate system. This coordinate system used to generate two-line elements (TLEs) of an orbiting satellite. TLEs describe an orbit's trajectory and is used for many orbit propagators such as SGP, SGP4, etc. Often times this

coordinate system is assumed be the inertial frame, but its axis definition do not fully make it inertial. When mapped on earth, the vector from frame's mean equator does not align with the first point of Aries. This point of intersection between the Earth's center of mass and mean equator location varies over time. This entails that the TEME frame rotates along the mean of date equinox [69]. Hence, it is often assumed to be nearly inertial as the changes are small. A drawback associated with TEME is that it does not account for the earth's precession and nutation effects caused by the sun and moon. Although this does not affect the orbit significantly but affects are noticeable for location of the equinoxes [70]. Thus, making the frame not fully inertial, as in inertial frames are referenced to the distance star called Aries. Thus, leading to affect the vernal equinox also known as first point of Aries. These changes cause minimal error at one instance but its does grow over time. The X-Axis deviates slightly from the January 2000 (J2000) epoch frame, which accounts for the mentioned effects as described in terms of Celestial Intermediate Origin (CIO) [67]. Therefore, making this an intermediate frame that is further transformed to the true ECI frame, J2000. This transformation will be addressed in the following Chapter 5.

Geocentric Coordinate System:

- X-axis : Deviated vector from the vernal equinox of January 2000 epoch with respect to Earth's nutation and precession effects.
- Y-axis : Completes the triad.
- Z-axis : Along the Earth's rotational axis.

3.2.4 January 2000 (J2000) frame (ECI)

The J2000 reference frame is referred as the International Celestial Reference frame (ICRF). Thus, by default when operations deal with satellite-based missions, the chosen ECI frame refers to this axis. This is also a geocentric coordinate system and was formed at noon January 1st, 2000, epoch.

This frame accounts for effects such as precession and nutation. Furthermore, equation of equinox is accounted for to align the TEME X axis to be coincident with earth's equatorial plane and the ecliptic plane. Another reason why J2000 frame is considered to be the ECI frame is because, the sun position model produces the location of the sun in the J2000 frame as it uses the Julian Ephemeris Date as a time reference [71]. This is where the Julian days and century is also realized as the J2000 epoch or the "zero" epoch. Therefore, making this the frame of choice to be the inertial frame. Throughout this thesis when this is frame either referred to as the J2000 ECI frame and is used as I in equations.

Geocentric Coordinate System:

- X-axis : Towards the vernal equinox direction at noon January 2000 epoch.
- Y-axis : Completes the triad.
- Z-axis : Along the Earth's rotational axis.

3.2.5 Earth Centered Earth Fixed (ECEF) Frame

The Earth Cantered Earth Fixed (ECEF) frame is also known as the International Terrestrial Reference Frame (ITRF). This is a geocentric frame is fixed to the celestial body. Unlike the two inertial frames that were previously defined, this frame rotates with the body. It can be perceived as a body fixed frame of the Earth. Several environmental models are derived from this frame as the effects are commonly observed and modelled. This is an intermediate step for a conversion to a topocentric frame but is not required for this thesis. The realization of this frame is crucial as the magnetic field model used for attitude estimation begins in this frame and is later transformed to an inertial frame. Moreover, earth orientation parameters such as length of day and polar motion, and Greenwich Apparent Sidereal Time (GAST) is utilized to form transformation matrices and express vectors in this frame.

Geocentric Coordinate System:

- X-axis : Vector going through the intersection of the prime meridian and the equatorial plane at 0 latitude and longitude.
- Y-axis : Completes the triad.
- Z-axis : Along the Earth's True North axis.

3.2.6 North East Down (NED) Frame

The North, East and Down (NED) is a non inertial reference frame, and is often used for airplanes, spacecrafts, etc. However, to describe attitude of objects which are relatively inertial with respect to Earth it is assumed to be locally inertial frame. Thus, it is a topocentric frame as it is specific to a geographic location [47]. The hardware experiment described in Chapter 8, conducted in this thesis makes use of this logic. Thus, it is commonly referred to as an inertial frame referenced for many systems such as cellphones and microcontrollers. Moreover, if the assumption is made that this is an "inertial" frame it would entail the parameters used to describe the frame don't vary. The parameters include latitude, longitude, and mean sea level.

Local tangent fixed object-centric Coordinate System:

- X-axis : Vector pointing north
- Y-axis : Vector pointing east
- Z-axis : Completes the TRIAD and is in the down direction

3.2.7 Spherical Coordinates with respect to ECEF

This frame is used to aid the onboard GPS receiver due to the manner of its measured quantity. This is a geocentric system as well since it uses the ECEF frame and converts it into polar coordinates. This means the Cartesian X, Y, Z definition is described in section 3.2.5 is used to

derive this coordinate system. Although cartesian coordinates are used throughout this thesis but the use of radial distance, azimuth and polar angles are used to describe the location of a revolving entity. This is analogous to the use of height, longitude, and latitude by the GPS/GNSS receivers. Moreover, the polar angle is equivalent to the co-latitude as it is measured from the true north. The quantities are expressed as follows [72]:

- r : Distance from the center of mass to the orbiting entity
- colatitude (φ) : Vector pointing east and is measured
- azimuth/longitude (θ) : Completes the TRIAD, in the down direction

3.3 Equations of Motion

To describe any rigid body system, it is described in terms of the dynamics of the system and kinematics. For static attitude estimation technique, these equations are not utilized as the problem is describe which begs a direct mathematical solution. However, for online attitude estimation these equations cater towards the formulation of the system of equations for state estimation.

3.3.1 Attitude Dynamics

The dynamics of the system relevant to this research only concerns with rotational motion. Leonhard Euler had derived a formula to describe rotational motion of any rigid body. As hinted in the previous section for LEO orbits various types of perturbations and disturbances experienced. Torque or moment that affects the rotational motion are characterized by this first order differential equation. Equation (3.1) expresses the form in which it is described as a change in angular rate as a function of time [27].

$$\dot{\omega}_B^{BI} = J_B^{-1}[\tau_B - \omega_B^{BI} \times (J_B \omega_B^{BI})] \tag{3.1}$$

where J is the moment of inertia of the CubeSat, or any rigid body expressed in BFF. ω is the angular rate of the rigid body with respect to ECI and expressed in BFF. Lastly, τ is the torque applied to the system and is expressed as B in notation form. The torque and angular velocity vectors can be expanded as they describe the motion is all 3 axes. Therefore, equation (3.1) can be expanded as follows:

$$\dot{\omega}_x = \frac{\tau_x + J_{yy}\omega_y\omega_z - J_{zz}\omega_y\omega_z}{J_{xx}} \tag{3.2}$$

$$\dot{\omega}_y = \frac{\tau_y + J_{xx}\omega_x\omega_z - J_{zz}\omega_x\omega_z}{J_{yy}} \tag{3.3}$$

$$\dot{\omega}_z = \frac{\tau_z + J_{xx}\omega_x\omega_y - J_{yy}\omega_x\omega_y}{J_{zz}} \tag{3.4}$$

Since the motion of interest is rotational, it is assumed that the rigid body's frame is parallel to the principal axes of inertia. This is ideal as it simplifies the moment of inertia matrix as the off-diagonal terms are approximated to 0. Moreover, this notion has validity because a CubeSat is symmetric leading to a resultant 3x3 diagonal inertia tensor and negligible off diagonal terms. However, for larger satellites which are variants of a polygon it is required to calculate all tensor terms. The ESSENCE CubeSat dimensional and mass characteristics are described in Table 3.1.

Table 3.1: ESSENCE CubeSat dimensions and COM breakdown.

CubeSat Dimensions			Center of Mass			CubeSat mass
X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]	Mass [kg]
0.1	0.1	0.3	0.05028	0.05029	0.1471	3.2745

As it is evident from the table, the center of mass is approximately at the center of the CubeSat. Thus, it supports the assumption made when forming the moment of inertia matrix. The system at hand is a CubeSat, the mass does not vary as there is no fuel or any components that dissipate in the vacuum. Thus, it is assumed the mass is constant and the inertia tensor is not a function of time, as shown in (3.5).

$$J = \begin{pmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{pmatrix}$$

$$J = \begin{pmatrix} 2.73\times10^{-2} & 0 & 0 \\ 0 & 2.73\times10^{-2} & 0 \\ 0 & 0 & 5.5\times10^{-3} \end{pmatrix} kg \cdot m^2 \tag{3.5}$$

where J_{xy}, J_{xz}, J_{yx}, J_{yz}, J_{zx}, J_{zy} are the off-diagonal term and are approximated to 0 kgm^2.

3.3.2 Disturbance Torques

It is seen in equation (3.1), torque is considered when describing the dynamics of a system. When operating in a LEO satellite, the most common environmental factors are disturbance torques. For ADCS, it is crucial to consider them as they to affect the controllability of the rotation motion. For attitude determination, these toques are inherited and considered by the sensors that are measuring angular rates. These sensors are easily incorporated in an online attitude determination algorithm rather than static. Gravity Gradient, Aerodynamic, Magnetic and Solar radiation pressure disturbance torques are found in LEO.

Gravity Gradient Torque

This form of disturbance torque is cause from Earth's gravitational field. Moreover, it affects satellites that have an odd distribution of mass leading to a non uniform attraction force. This is

leads to a formulation of torque at orbits because the position vector of the LEO satellite is perpendicular to the force of attraction vector. However, there are advantages associated with the gravity gradient torque. As it is commonly used for passive actuation called, gravity gradient stabilization.

$$\tau_{gg} = \frac{3\mu}{\|r_B\|^3}\left[\hat{r}_B \times J_b \hat{r}_B\right] \tag{3.6}$$

where $\hat{r}_B$ is the distance from the center of the primary body to an orbiting mass as a unit vector in the satellite's body fixed frame, μ is the gravitational parameter of earth, and J is the moment of inertia matrix.

Magnetic Torque

Magnetic Torque is an internal torque generated due to the present of a magnetic dipole. Incorporation of current loops for magnetorquers result in a local dipole. These are often used for Bdot control which can produce torques up to 10^{-5} Nm. However, the disturbances caused by the torque generate from this dipole are several orders less than 10^{-5} Nm [27].

$$\tau_m = m_B \times B_B \tag{3.7}$$

where m_B is the magnetic dipole moment and B_B is the magnetic field, both expressed in the body fixed frame.

Aerodynamic Torque

For LEO satellites, environmental effects such as atmospheric drag results from the Earth's atmospheric effects. Common result of such effect leads to a shortened orbital lifetime, and aerodynamic torque on the CubeSat. It is necessary to model the earth's atmosphere using the

exponential model shown in (3.8) [27]. This function is an exponential decay, so these effects do not affect orbiting satellites that are higher than 600 km in altitude.

$$\rho = \rho_o \exp\left(-\frac{h-h_o}{H}\right) \tag{3.8}$$

where ρ_o is the reference density, h_o is the reference height, h is the altitude from the earth's surface, and H is the scale height. The ESSENCE CubeSat it is assumed to be operating in the same orbit as the ISS. Thus, for the 400-450 km altitude range (h):

- ρ_o = 3.725 x 10^{-12} km/m^3
- h_o = 400 km
- H = 59.4 km

The disturbance torque is not equal on all sides of the CubeSat as it is dependent on the area exposed to the flow. Therefore, the force of drag is dependent on the incident angle communicating if the satellite's surface is exposed or not, shown in (3.9) [27].

$$F_d = -\frac{1}{2}\rho C_d S_a v_{rel,B}\cos(\theta) \tag{3.9}$$

$$\theta = \cos^{-1}\left(\frac{n_B \cdot v_{rel,B}}{|v_{rel,B}|}\right) \tag{3.10}$$

where n_B is the boresight vector, ρ is the atmospheric density, C_d is the drag coefficient, S_a is the surface area and $v_{rel,B}$ is the relative velocity of the satellite in the body fixed frame. The drag coefficient is critical in this domain as it drives the formulation of the drag force. This coefficient is characterized by analyzing how a fluid flows through a given surface area of an object. For a

CubeSat this entails different drag force values for each side. This is because the surface area is not the same for two of the six sides. However, as mentioned for LEO CubeSat missions the 2-2.3 range is sufficient to form this force.

Lastly, torque is calculated by using these quantities described in (3.11). This is expressed as a summation, due to the altering values of force of drag acting on a satellite [27].

$$\tau_d = \sum_{i=1}^{n} r_i \times F_{d,i} \tag{3.11}$$

where r_i is the center of pressure vector, and $F_{d,i}$ is the force of drag acting on the on the i^{th} face all expressed in BFF.

Solar Radiation Pressure Torque

The final disturbance torque which affects the system is caused by the intense radiation produced by the sun. Solar radiation pressure causes phenomena such as single event upsets, components degradation, outgassing, etc. Like other torques, a force is also generated which leads to this torque. The concept is similar to aerodynamic torque, but the effects are minimized when entering an eclipse. Furthermore, the force caused by solar radiation is dependent on the incident angle is measured from the boresight vector of the i^{th} face to incoming flow of solar radiation, (3.12) [27].

$$\cos\left(\theta_{SRP}\right) = n_B \cdot s_B \tag{3.12}$$

where n_B is the boresight vector and the s_B is the satellite to sun position vector in BFF. Thus, as long as the solar vector is not perpendicular to the boresight, the force of solar radiation pressure is calculated. This is illustrated in (3.13) [27].

$$F_{SRP} = -P_{\odot} S_a [2\left(\frac{R_{diff}}{3} + R_{spec} \cos(\theta_{SRP}) n_B + (1 - R_{spec}) s_B \right) \cos(\theta_{SRP})] \tag{3.13}$$

where $P_{\odot}$ is the solar radiation pressure, R_{diff} is the reflection coefficient and R_{spec} is the specular reflection coefficient.

Lastly, the solar radiation pressure torque is shown in (3.14) using the variables defined in (3.12) and (3.13).

$$\tau_{SRP} = \sum_{i=1}^{n} r_i \times F_{SRP,i} \tag{3.14}$$

The mentioned four disturbance torques do not play a large role in attitude determination as they are simply treated as process noise that are not required to be modelled for attitude estimation algorithms. However, when enabling control to system these disturbances are required to be modelled and added to the control torque produced by onboard actuators. For a LEO orbit solar radiation pressure torque isn't very prominent as forces such as drag, and gravity are more dominant. This is simply because the Earth is the closest celestial body, so its effects more noticeable than the Sun.

3.3.3 Attitude Kinematics – Quaternions

As prefaced in the literature review conducted in Chapter 2, it was found the use of Euler angles lead to a "gimbal lock" problem. Thus, causing its use limited to small attitude maneuver due to the presence of singularities. This is easily avoided using the quaternion for attitude representation formulated by Hamilton in 1844.

A quaternion is a four-dimensional quantity that is formed by four parameters. These parameters are expressed in real and complex numeric quantities. There is a scalar part to the quaternion and

a vector part consisting of 3 components as shown in (3.15). Moreover, an equivalent reduced form is also represented in (3.15), in which the scalar term is listed first. Quaternions are not often the preferred quantity to express orientation. The difficulties associated with quaternions is its non-intuitive nature and irregular mathematical relationship.

$$q = (q_0, \vec{q}) = [q_0, q_1, q_2, q_3]^T \tag{3.15}$$

where q_0 is the scalar and $\vec{q}$ are the vector components of the quaternion. In order to make use of this representation it is the required to meet the quaternion norm constraint, shown in (3.16). The primary reason for this is that the quaternion will grow to infinity leading to more complex solution while its being update in the online attitude techniques.

$$|q| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \tag{3.16}$$

A transformation using a quaternion from frame I to B is described as q_{IB} and the inverse is described in (3.17).

$$q_{IB} = q_{BI}{}^{-1} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix}^T \tag{3.17}$$

For state estimation, it is necessary to incorporate the change in quaternion as a function of time. Thus, for the rotational motion, (3.18) is defined to model the concept and will be used for the system model definition. This can be also described in another manner as shown in (3.19) as it a more intuitive form which easily utilizes the time variation in both scalar and vector components of a quaternion.

$$\dot{q}_{BI} = \begin{bmatrix} \dot{q}_0 \\ \dot{\vec{q}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\vec{q}\omega_B^{BI} \\ \frac{1}{2}(\vec{q}^{\times} + q_0 I_3)\omega_B^{BI} \end{bmatrix} \tag{3.18}$$

$$\dot{q}_{BI} = \frac{1}{2}\Xi\left(q_{BI}\right)\omega_B^{BI} = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{3.19}$$

where the quaternion attitude representation is expressed as BFF relative to ECI.

CHAPTER 4 OVERVIEW OF ATTITUDE AND ORBIT DETERMINATION SENSORS

In this chapter, the contents explore all the sensors that are of low cost and suitable for a nanosatellite mission. This chapter seeks to aid amateur satellite developers with their attitude determination algorithms by utilizing appropriate sensors for their system. The contents communicate the measurement models and interpretation of all basic attitude determination sensors.

Throughout this thesis, the sensors explored are relevant to the ESSENCE CubeSat mission. The sensors are chosen for this mission are common and low cost, as it allows for a cost-effective solution to help standardize a CubeSat's attitude determination system. Constraints such as cost forbid the use of more accurate sensors such as star trackers. Therefore, the following sensors are used:

- 3-Axis Micro MEMS Gyroscope
- Fibre Optic Gyroscope (FOG)
- Coarse Sun Sensor (CSS)
- Fine Sun Sensor (FSS)
- 3-Axis Magnetometer

This incorporation of the mentioned five sensor typed are outlined in the ESSENCE mission's interconnect diagram in Figure 4.1. The nine boards in this figure form the stack of the 3U CubeSat. These sensors are ideal for standard amateur missions, due to their compact characteristics and

highly intuitive nature to form relevant vectors for ADCS. To test attitude determination for the current research, a simulator was built by Ryerson University's PhD candidate Mike Alger, to mimic all the environmental effects and orbital anomalies for a typical LEO satellite. Hence, it was used to mimic the sensor outputs by using useful quantities such as angular rates, quaternion, sun position and magnetic field vectors. Upon manipulation of these simulated values with the information provided by every sensor's datasheet, sensor outputs are produced using measurement models in a simulated setting. The simulated environment in formulated in MATLAB/Simulink.

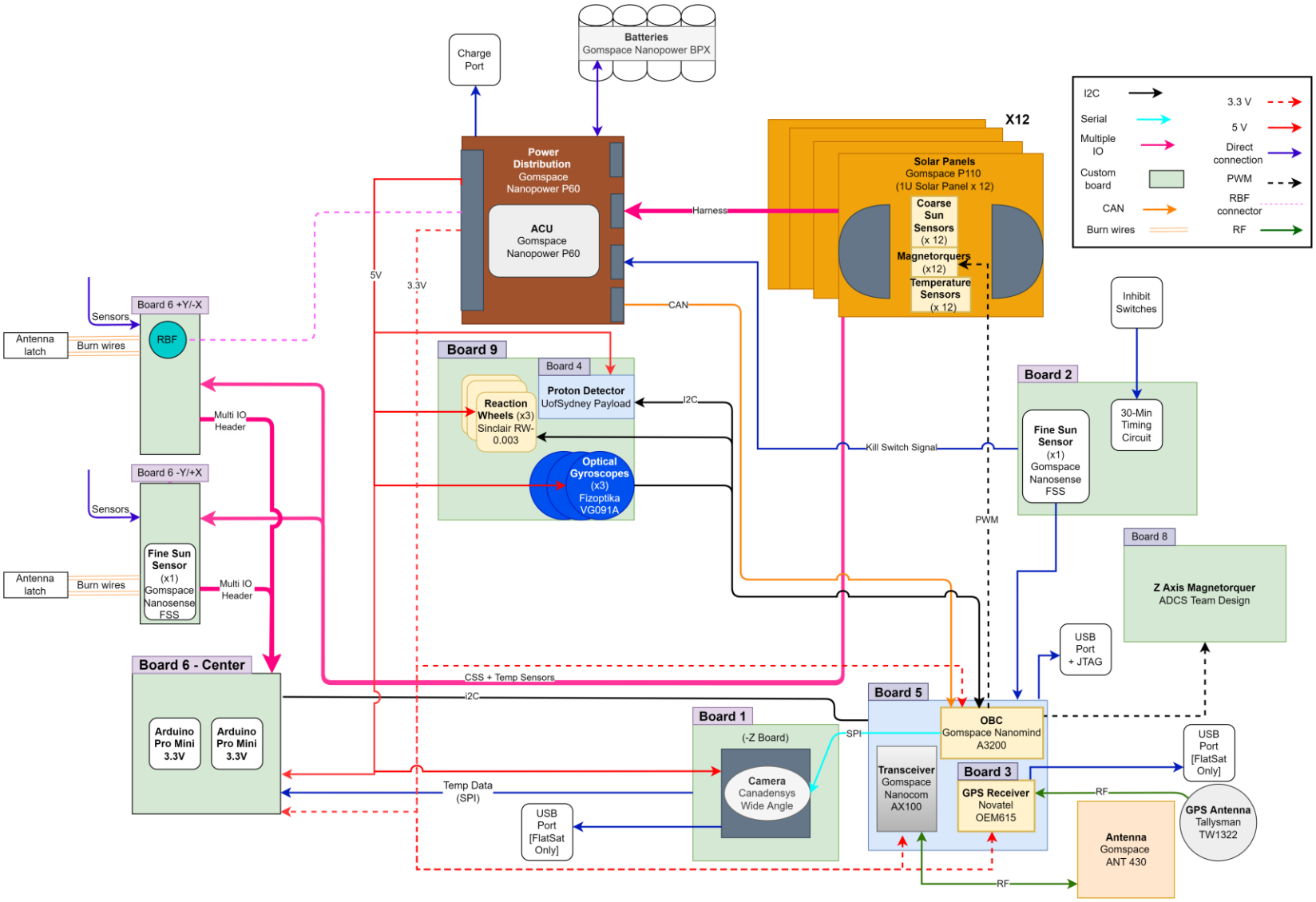


Figure 4.1: ESSENCE CubeSat's Interconnect Diagram outlining placement of attitude sensors in BFF.

4.1 Three-axis MEMS Gyroscope

The GOMspace Nanomind A3200 is the onboard computer (OBC) for the ESSENCE mission and consists of a three axis MEMS gyroscope and magnetometer [73]. The InvenSense MPU-3300 is the MEMS gyroscopes embedded int the OBC [74]. The following sensors provides the angular velocity in terms of the sensor frame. It is crucial to convert this into the satellite's body fixed frame as attitude estimation is conducted in the global body reference frame. Figure 4.2 shows the local sensor frame of reference, and the conversion DCM is described in (4.1).



Figure 4.2: Nanomind A3200 3-axis gyroscope and magnetometer sensor frame [73].

$$\omega_B = TM_{sen}^{B}\omega_{sen}$$

$$\omega_B = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}\omega_{sen} \tag{4.1}$$

where TM_{sen}^{B} is the transformation matrix which converts the measured value from the sensor frame to the BFF, and ω_{sen} is the angular velocity measured in the sensor frame.

This sensor is used for online attitude determination algorithm as the static method makes use of modelled vector in both inertial and body frame. Although, it is simple to get the vector in BFF, it isn't as intuitive to describe it in the J2000 ECI frame. A 16-bit analog to digital converter (ADC)

is associated with this MEMS gyroscope and relays the information internally to the OBC. This is information is required to be known as it allows the developer to form error variance values for the sensor. The gyroscope measurement function is expressed as (4.2) [27]. Table 4.1 shows the characteristics required to define the sensor noise covariance matrix for an online attitude estimation method. The parameters are substituted as defined in (4.2).

$$\begin{aligned} \omega_B^{meas} &= \omega_B^{true} + \beta + \eta_v \\ \dot{\beta} &= \eta_u \end{aligned} \tag{4.2}$$

where ω_B^{meas} is the measured angular velocity, ω_B^{true} is the true angular velocity β is the bias/drift associated with the angular rate, η_v and η_u are zero-mean Gaussian white-noise values.

The bias is incorporated into the sensor measurement model after $\dot{\beta}$ is integrated. For online estimation, it is crucial to include this in the state variables to stabilize the estimated values. Figure 4.3 shows three plots that display the measured angular rate, noise with bias and true bias without noise, once transformed from the sensor frame to the CubeSat's BFF using (4.1). As observed, a large deviation from the true vector is seen in the measured quantity. This is due to the presence of a bias associated with the sensor. The measurement plot shows the order of the values are the same as the simulated bias plot, hinting the main cause of error in the measured output.

Table 4.1: Parameters of the MEMS gyroscope integrated in the Nanomind A3200 [73] [74].

Parameter	Value
Full Scale Range	450 deg/sec
Bias Instability	15 deg/hr
ADC Word Length	16 bits
Nonlinearity	0.2%
Cross-Axis Sensitivity	±2 %
Frequency Range	Up to 1kHz

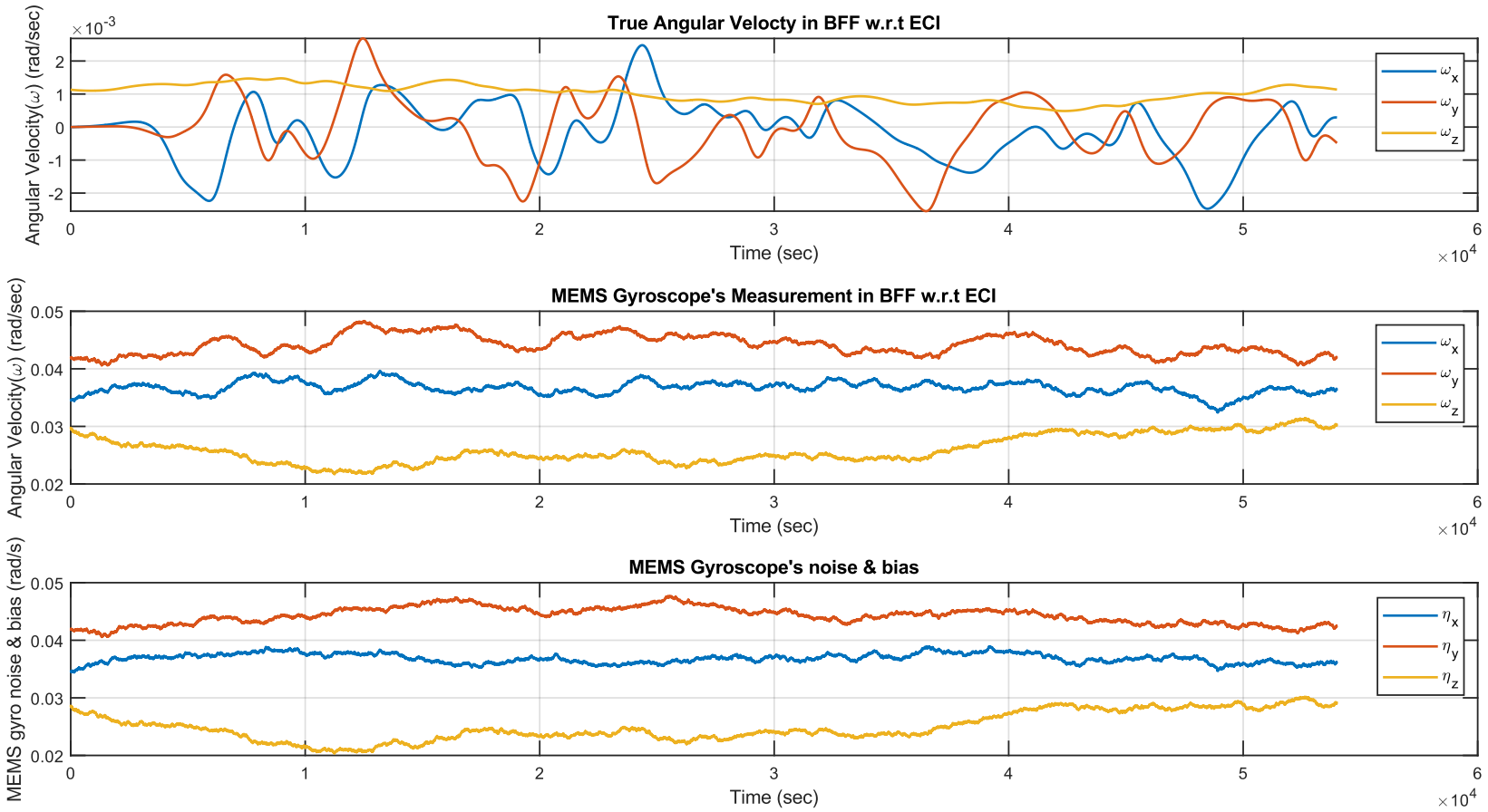


Figure 4.3: True angular velocity in the system, measured quantity, noise & bias present in the measured quantity for MEMS gyroscope.

4.2 Fibre Optic Gyroscope (FOG)

An optical gyroscope is one of the most accurate means to measure angular velocity of a rigid body. Unlike the MEMS gyroscope, it utilizes the Sagnac effect to produce its measurements [75]. The output is the same as a MEMS gyroscope, but at a greater accuracy. This process is illustrated in Figure 4.4 which shows the passing of light in two directions and reporting measurements by keeping track of travel time when a rotation occurs. These sensors avoid the risks associated with MEMS gyroscopes as they are not prone to faulty measurements and high error values. These MEMS based issues are caused by vibration, shock, overheating, etc.

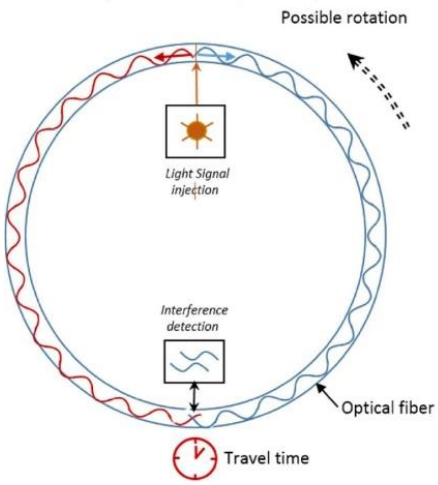


Figure 4.4: Fibre Optic Gyroscope operational view [75].

The Fizoptika VG091A is the fibre optic gyroscope chosen for the ESSENCE mission, but unlike the MEMS gyroscope this is not a 3-axis device. Therefore, the total quantity required of this sensor is three, and is placed orthogonally to each other in terms of BFF. The measurement model is the same as the one outlined in (4.2) but at a greater accuracy. Table 4.2 outlines the parameters associated with this component. Figure 4.5 shows the simulated model of the MEMS and Fibre Optic gyroscopes models using (4.2). It can be seen both models are the same but differ at the noise blocks seen in the Figure 4.5. The noise variance is where the difference is found between

the MEMS and Fibre Optic gyroscopes.

The simulated measured output plots are presented for FOG in Figure 4.6. The figure displays true angular velocity, the measured angular rate and noise with bias used for online determination filter. Similar outcomes as the MEMS gyroscope is noticed but with at much better accuracy when compared to the truth angular velocity vector. Moreover, the deviation at the each of the vector components from the truth is far less drastic than the ones seen for the MEMS gyroscope.

Table 4.2: Parameters of Fizoptika VGA091A in conjunction with ADS1115 ADC [76] [77].

Parameters	**Values**
Input Range	300 deg /sec
Scale Factor	7 mV/ deg /sec
Full Scale Range (ADC)	±2.048 V
Angular Random Walk	0.04 °/√hr
Bias	3 °/hr
Frequency Range	Up to 1kHz

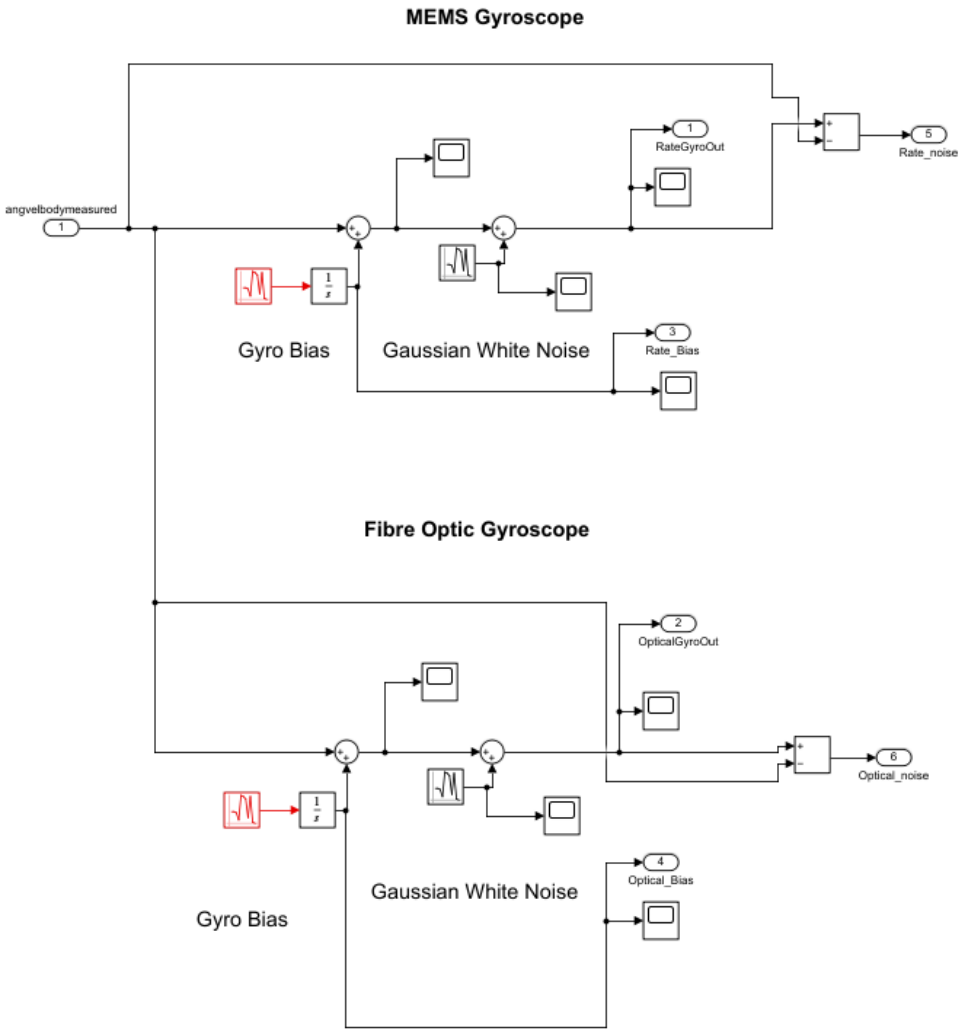


Figure 4.5: Sensor model simulated via Simulink using (4.2) for MEMS and Fibre Optic gyroscopes.



Figure 4.6: True angular velocity in the system, measured quantity, noise & bias present in the measured quantity for the FOG.

4.3 Coarse Sun Sensor (CSS)

Coarse sun sensors are one of the simplest of all the attitude sensors in terms of system incorporation. However, there are several steps to manipulate its output and make it useful to conduct attitude determination. For the ESSENCE mission, a suite of twelve coarse sun sensors embedded in the NanoPower P110 solar panel, are employed and placed on the ±X and ±Y axis defined in the satellite's BFF [78]. This is shown in the interconnect diagram in Figure 4.1. The figure exhibits all connections from the solar panels to Board 6's ±X and ±Y faces.

There are three factors that are necessary to consider in order to use the sensor's output. First, the CSS is integrated in the solar cell and outputs as current measurement (*I*). Using (4.3) it is converted into an incident angle (θ), measuring the angle form the sensor's boresight to the sun ray. Second, the usage of this sensor is limited by the satellite's eclipse periods and the sensor's

field of view (FOV). This is where the sensor is not considered when incorporating in an MEKF or any of the three static attitude estimators. Third, is the formation of the sun vector. Using the angle measurement and basic mathematical relationships, the sun vector is formed in BFF. The GOMspace NanoPower P110 is the chosen solar cells for the ESSENCE mission as it comes with CSS. Table 4.3 shows the all the relevant parameters for this sensor [44].

$$\theta = \cos^{-1}\left(\frac{I_i}{I_{\max}}\right) \tag{4.3}$$

Table 4.3: NanoPower P110 parameters relevant via datasheet [78].

Parameter	Values
Cell Efficiency	30%
Cosine Error	1.8 deg
Current (Short Circuit Current @ 1367 W/m^2)	930µA
Frequency Range	Up to 50 Hz

Figure 4.7 shows how the angle derived from the measured quantities of the CSS is interpreted. It is evident a single panel is not sufficient to determine relative solar position in the satellite's body frame. Therefore, multiple sensors are required on the faces of the CubeSat to ensure the best representation of the sun vector.

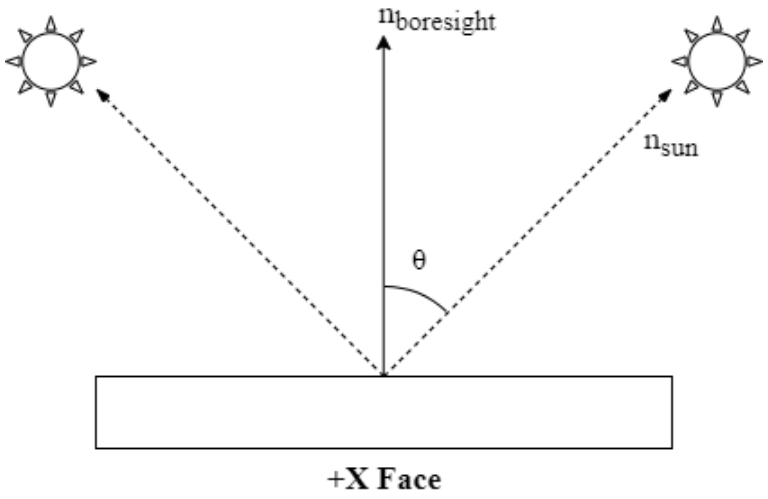


Figure 4.7: Measured angle described in BFF using for +X face of the ESSENCE CubeSat (4.3).

The sensor model is further expressed to incorporate the error experienced by forming the incident angle from the CSS's datasheet, shown in (4.4).

$$\theta^i_{meas} = \theta^i_{true} + \eta_v \tag{4.4}$$

where θ^i_{true} is the angle between the boresight vector of the ith face and the observed sun vector, η_v is the Gaussian white noise and θ^i_{meas} is the measured angle.

Once modelled, this is described in Figure 4.8 which shows the true angle, measured angle, noise, and the visibility flag. The validity flag communicates when the satellite is in eclipse and when the sun is outside the sensor's FOV (75 deg). Whenever any of these conditions are valid, a drop in measurement is noticed that only consists of sensor noise. In addition, the noise grows over time as there is no measurement available. Thus, by accumulating all twelve incident angles, (4.5) is used to form sun vector is formed [44].

$$\hat{n}_{sun} = \sum_{i=1}^{12} \hat{n}_{boresight,i} \cdot \cos\left(\theta_i\right) \tag{4.5}$$

where $\hat{n}_{boresight,i}$ is the boresight unit vector normal to the corresponding CubeSat face and $\hat{n}_{sun}$ is the sun position unit vector. Calculating the unit vector is the end goal for sun sensors, as the direction vector of the sun position is a critical quantity for attitude estimation.

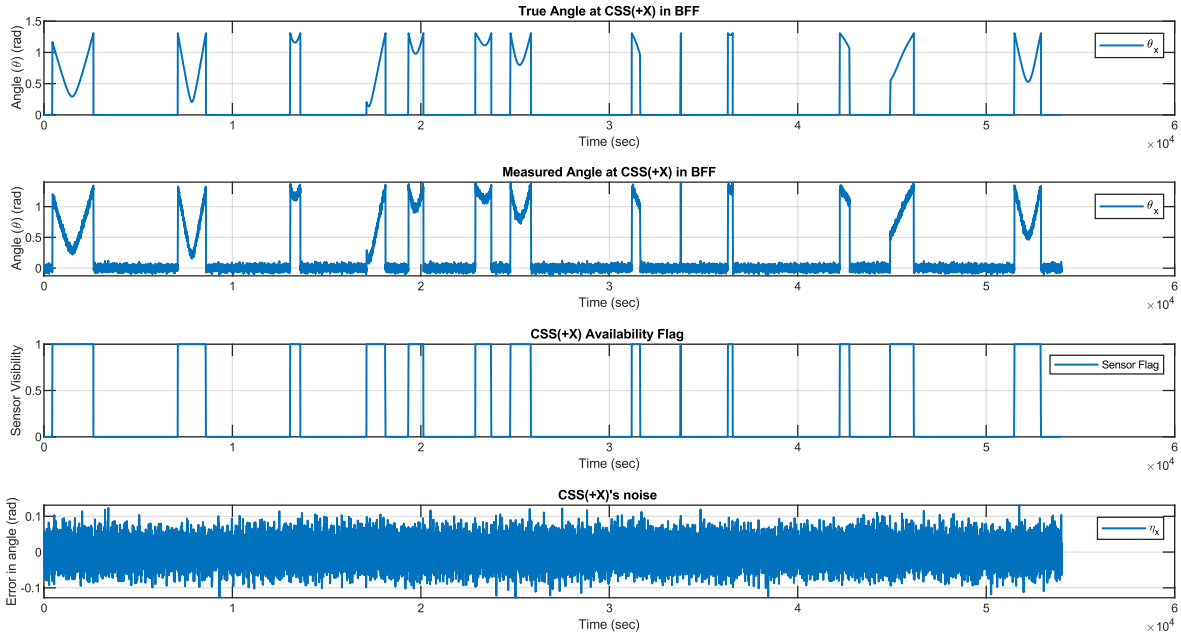


Figure 4.8: True Angle with sun vector and boresight vector, measured quantity, sensor visibility flag and CSS noise.

4.4 Fine Sun Sensor (FSS)

A Fine Sun Sensors are another form of a sun sensors and are more accurate than the CSS. The FSS uses a more constraint-based output which reports the measured quantity in a more accurate fashion. GOMspace Nanosense FSS was chosen, but only 2 were incorporated on the +Z and +Y axis in the BFF [79]. The decisions towards the quantity and the location were chosen due to budget, geometric, and size constraints of the ESSENCE mission.

Similar to the CSS, FSS has a photodiode present which detects the illumination caused by the sun. However, FSS reports the measurements in terms of spherical coordinates in the sensor frame.

A quadrant is formed that is coincident to the centre of photodiode which provides the location where the light hits the sensor. Once the location is determined, the FSS would provide the measurement in terms of azimuth and elevation. The altitude is not required as the sun position unit vector is formed directly from the mentioned two angles. The product manual of the FSS contains a medium to interpret the raw measurements to the true sensor outputs [80]. It is assumed the sensor frame coincides with the body fixed frame. Thus, using (4.6) as per FSS's product manual, the sun position unit vector is formed.

$$\hat{n}_{sun} = \begin{bmatrix} \cos(\theta_{el}) \\ \sin(\theta_{el})\cos(\phi_{az}) \\ \sin(\theta_{el})\sin(\phi_{az}) \end{bmatrix} \tag{4.6}$$

where $\hat{n}_{sun}$ is the sun's position unit vector, θ_{el} is the elevation angle and ϕ_{az} is azimuth. The FOV factor is still relevant for the FSS. Thus, the visibility flag is also checked for both eclipse and FOV limitations of the sensor.

The parameters relevant to develop the model in a simulated setting is outlined in Table 4.4 and is used for both attitude determination methods. The sensor model demonstrated in (4.7), incorporates the (4.6), Gaussian white noise η_v, in conjunction with Table 4.4.

$$\hat{n}_{sun,meas} = \hat{n}_{sun} + \eta_v \tag{4.7}$$

Table 4.4: Nanosense Fine Sun Sensor parameters relevant via datasheet [79] [80].

Parameters	Values
Accuracy (<60 deg)	±2 deg
FOV Half angle	60 deg
Frequency Range	Up to 50 Hz

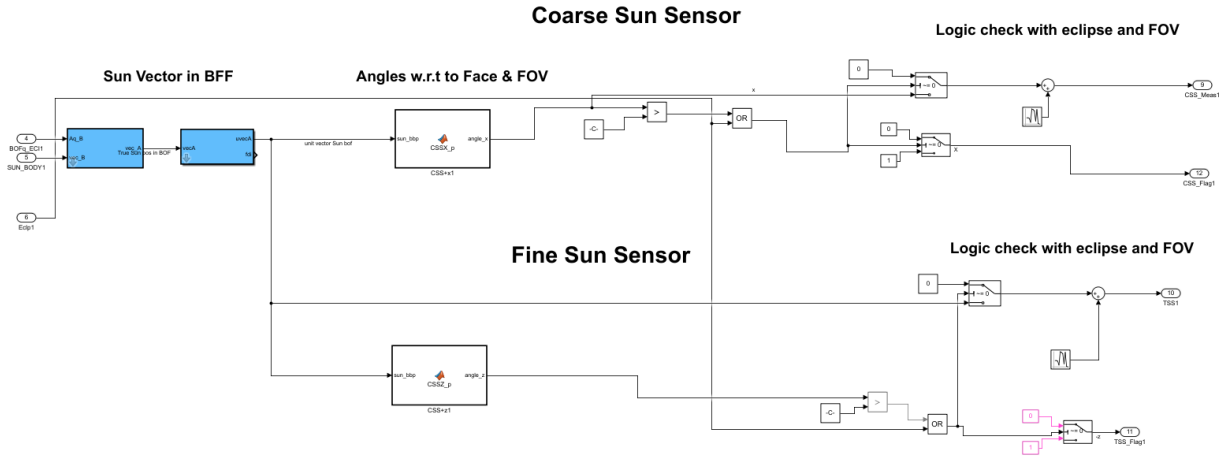


Figure 4.9: Sensor model simulated via Simulink using (4.4) and (4.7) for the CSS and FSS.

Figure 4.9 shows the incorporation of the sun vector in the BFF generated by the LEO simulator which is manipulated to mimic the sensor measurement models for CSS and FSS. Figure 4.10 demonstrates plots for the true sun vector, +Z FSS in BFF, noise, and +Z FSS availability flag. It is seen when the sensor is out of FOV or in eclipse the measured quantity is just the noise in the system. This is similar to the CSS mentioned in the previous section, hence making these types of sensors limited in its usage. However, they are sufficient for most satellite missions as a suite of multiple CSS or FSS are present on the satellite. The -Z side of the CubeSat does not contain any attitude sensors, but there are multiple 3-axis sensors such as magnetometer and gyroscopes that fill this void.

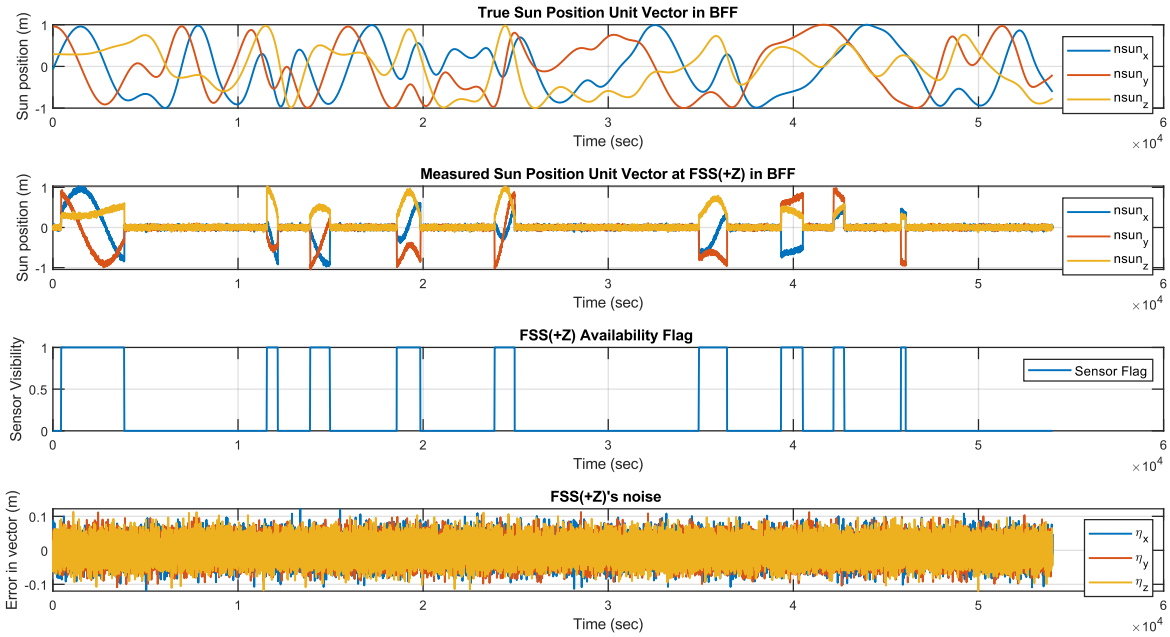


Figure 4.10: Shows the true sun position unit vector, measured sun position for +Z face, Visibility Flag of +Z face and noise associated with the FSS.

4.5 Magnetometer

Similar to the MEMS gyroscope mentioned, this sensor is also integrated in the Nanomind A3200 OBC [73]. As seen in Figure 4.2, the sensor outputs in sensor frame, and requires a transformation to the body frame using the TM_{sen}^{B} transformation matrix shown in (4.1). The specific three axis magnetometer used in the OBC is the Honeywell HMC5843. This MEMS sensor provides the magnetic field measurements in terms of Gauss. It is necessary to convert this to Tesla because it is a standard unit used to describe magnetic field strength. The characteristic of the magnetometer is outline in Table 4.5. Moreover, the magnetometer's measurement model is shown in (4.8) [48]. It is used to simulate the sensor's measured output in BFF, and the Simulink model can be found in Figure 4.11. In addition, Figure 4.12 shows the true value of the magnetic field, measured magnetometer reading, and the noise associated in the simulated sensor.

$$B_{meas} = B_{true} + \eta_v \tag{4.8}$$

Table 4.5: Honeywell HMC5843 parameters relevant via datasheet [73] [81].

Parameters	Values
Field Range	±4 G
Resolution	7 mG
Frequency Range	Up to 116 Hz

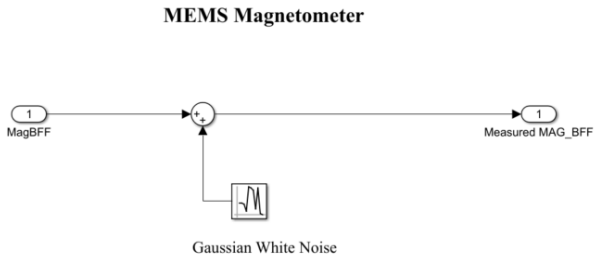


Figure 4.11: Sensor model simulated via Simulink using for MEMS magnetometer.

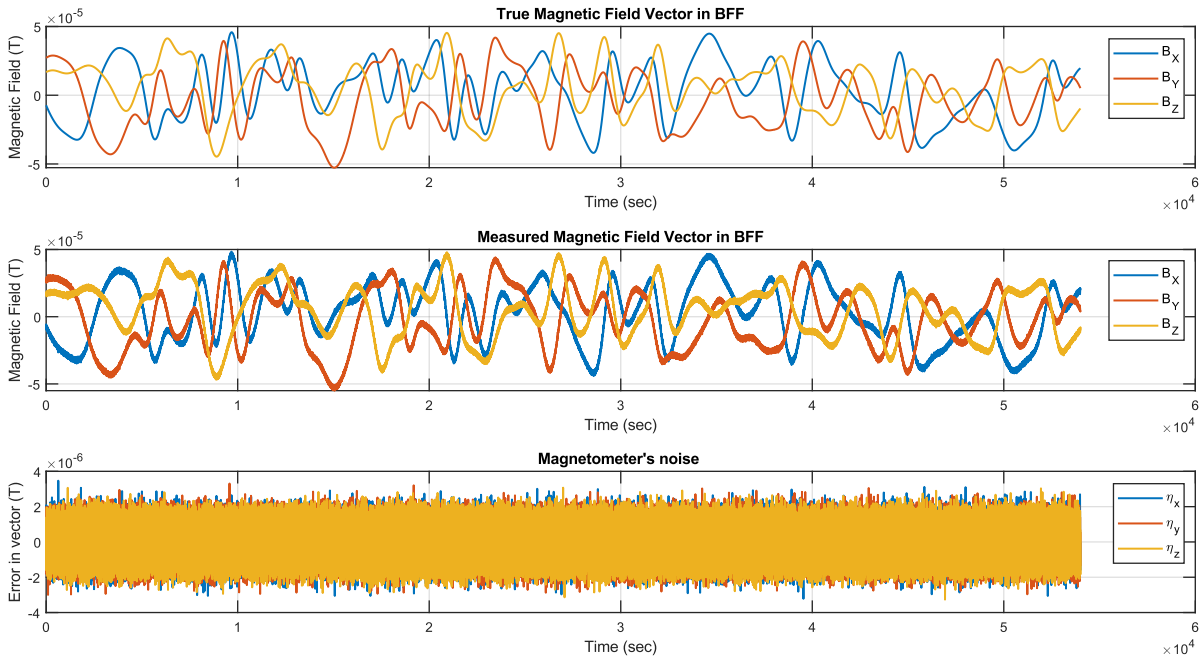


Figure 4.12: Shows the true magnetic field vector, measured magnetic field vector, and noise associated with the magnetometer.

4.6 Global Navigation Satellite System (GNSS) Receiver

The Global Navigation Satellite System (GNSS) or GPS receiver is incorporated in the ESSENCE mission for orbit determination. This aids with the implementation of environmental models used for both types of attitude estimation techniques. This receiver works in conjunction with a patch antenna. This antenna and receiver linkage works with the GPS constellation to measure the satellite's position and velocity data. GPS operates in the World Geodetic System 1984 (WGS-84) is referenced by the onboard receiver to form the position and velocity [82]. Table 4.6 shows the parameters used in the WGS-84 is used to interpret the measured quantities in the ECEF frame. An assumption is made where the receiver outputs measurements in ECEF which describes location of the satellite or any entity that used the GPS receiver [82] [83].

For the ESSENCE mission, the NovAtel OEM615 GNSS receiver is used due to its ability to provide outputs that are highly precise [84]. A GNSS receiver uses the signals received in the L1 and L2 frequency bands from a series of orbiting GPS satellites to determine several pseudoranges. The receiver uses these quantities and further analyzes it by employing numerical methods to solve for the measured navigation data.

Table 4.6: Parameters of WGS-84 ellipsoid [82].

Parameter	Variable Name	Value
Semi-major axis of ellipsoid	a	6378.137×10^{-3} m
Flattening factor	f	1/298.257223563
Earth's angular rate	ω_e	7292115.0×10^{11} rad/sec
Gravitational Constant	μ	6.674×10^{11} $m^3/kg/sec^2$
Speed of light	c	2.998×10^8 m/sec

Figure 4.13 shows how the OEM615 receiver board is distributed in terms of its functionality. The diagram is split into 2 sections radio frequency and digital sections [85]. A key benefit of this receiver is its ability to process the GPS signal data and provide a solution consisting of position and velocity in the ECEF frame.

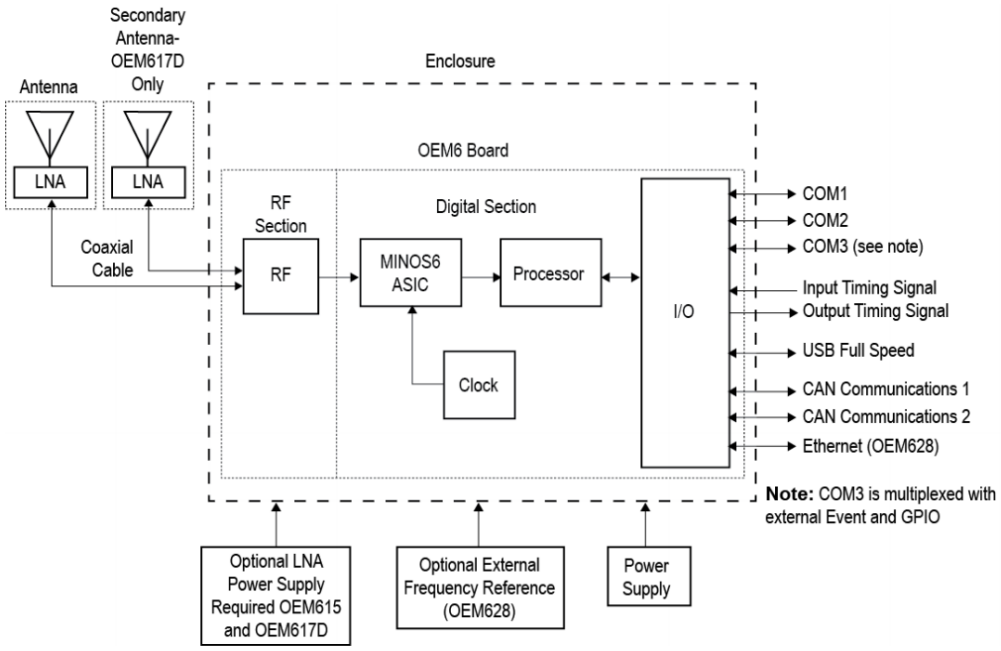


Figure 4.13: OEM615 Receiver System operational diagram [85].

Table 4.7 demonstrates the accuracy levels of the GNSS receiver provides. The RMS values placed here are listed as the measurement position and velocity error values for a GPS functioning on single point L1/L2. GPS satellite's transmit values in two frequency L1 and L2. This is why the values shown in Table 4.7 these were selected. The chosen receiver also had 1 ppm level position accuracy but the worst case for selected for all GPS receiver based simulated tests. Thus, using these quantities, the simulated GPS output is modelled. The position and velocity values outputted form the receiver are modelled using (4.9) and (4.10). Figures 4.14, shows the GPS receiver model

formed in Simulink. Moreover, Figures 4.15 and 4.16 consists of plots showing the true and measured position and velocity vectors along with noise associated in each of the measured quantities. The difference between true and measure is negligible because the error associated with this board is very low as the accuracy levels is high for the OEM615 receiver.

Table 4.7: Shows the RMS error in position and velocity for the GNSS Receiver [84].

Parameters	Value
Single Point L1/L2 (RMS)	1.2 m
Velocity (RMS)	0.03 m/sec

$$p_{meas} = p_{true} + \eta_v \tag{4.9}$$

$$v_{meas} = v_{true} + \eta_v \tag{4.10}$$

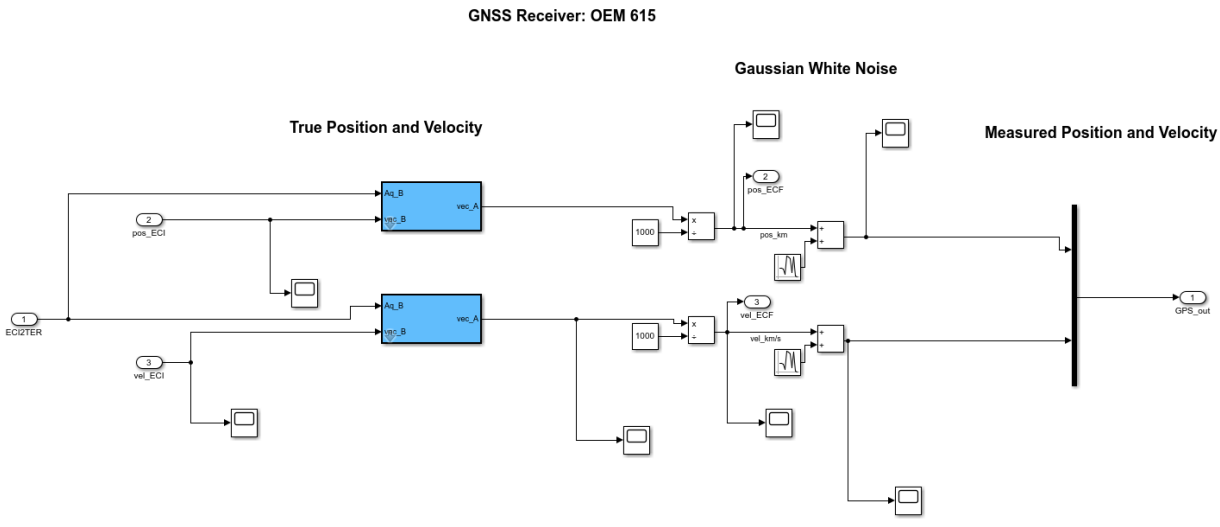


Figure 4.14: OEM 615 GNSS Receiver sensor model using (4.9) and (4.10).

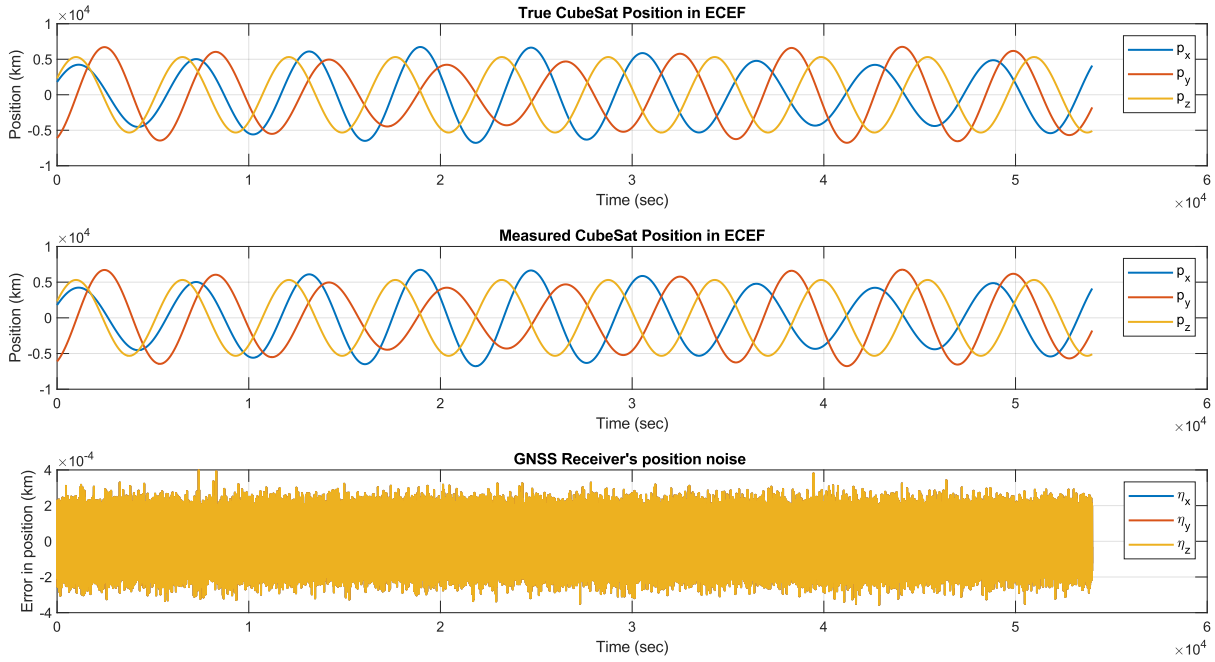


Figure 4.15: Shows the true CubeSat position, measured CubeSat position, noise associated with the measured quantity in ECEF for the GPS receiver.

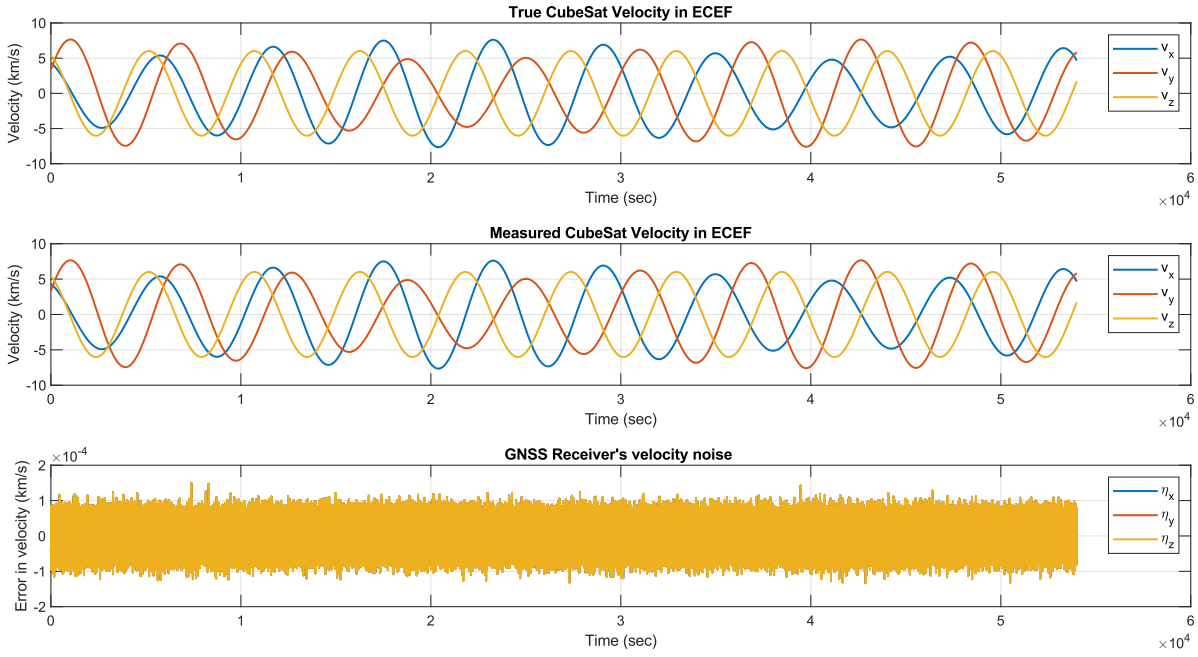


Figure 4.16: Shows the true CubeSat velocity, measured CubeSat velocity, noise associated with the measured quantity in ECEF for the GPS receiver.

CHAPTER 5 ORBIT DETERMINATION/PROPAGATION

This chapter initiates the first step to conduct attitude determination of an orbiting entity. The concept of time is first introduced, as it is required to be considered to determine orbit. In addition, two ways to provide position and velocity vectors of a satellite are discussed and analyzed. The two methods explored, are orbit propagation via TLEs and SGP model, and real-time orbit determination via GPS receiver based Kalman Filter. Lasty, environmental models are formed and compared through both orbit determination methods.

5.1 Overview of Time Domain

To determine the orbital characteristics of an earth orbiting satellite, it is important to incorporate time in a correct manner. The concept of time is interpreted as a simple measured quantity that is relative to represent a known phenomenon. For example, the derivation of hour is made because earth rotates at an angular rate of 15 deg/hr. Moreover, various effect resulting from earth's motion are represented by several time standards. The time standards that are and relevant for orbit estimation are Coordinated Universal Time (UTC), Universal Time (UT1), International Atomic Time (TAI) and Terrestrial Time (TT). The International Earth Rotation and Reference System (IERS) service keeps track of Earth Orientation Parameters (EOP) in bulletins A to C, as they are required to define some of the time standards such as UT1, TAI and TT [86]. Moreover, the bulletins also provide parameters for frame transformation, which is utilized for orbit determination. These include polar motion angles, time difference values, length of day (LOD) and nutation correction parameters.

Reference frame transformation require calculations to be conducted in terms of Julian date (JD).

Thus, it is necessary to convert time in terms of UT1 and TT for the calculation of Julian date and century respectively. (5.1) shows the relation among various time standards to achieve this goal [67].

$$\begin{aligned} UTC &= UT1 - \Delta UT1 \\ TAI &= UTC + \Delta AT \\ TAI &= GPS + 19s \\ TT &= TAI + 32.184s \end{aligned} \tag{5.1}$$

where

- $\Delta UT1$: Represents the difference between UT1 and UTC time standards
- ΔAT : Represents the difference in TAI and UTC time standards

These time differential parameters are updated periodically in the Bulletin A on the IERS website [86]. Throughout this thesis, any analysis that makes use the time difference parameters are shown in Table 5.1. Moreover, the GPS to TAI relation is drawn from the epoch of GPS, January 7th, 1980, 0:00:00 UTC [67]. This is needed to synchronize time when utilizing the GPS receiver for orbit determination. Furthermore, the difference between UT1 and UTC is not large. Thus, it often to equate the quantities for any applications using these standards. A primary benefit in this assumption is that these time calculations do not contribute to error factors seen in both orbit estimators. However, throughout this research this difference is considered and incorporated to demonstrate the reference frame transformations that are reliant on time correctly.

Table 5.1: Shows the Time difference parameters from Bulletin A on the IERS website [86].

EOP Time Difference Parameters	Values
$\Delta UT1$	-0.2 seconds
ΔAT	37 seconds

As hinted, Julian date and century is calculated in UT1 time representation, shown in equation (5.2) [67]. As per Chapter 3 the ECI frame is realized at the J2000 epoch as it is the ICRS standard. Thus, when calculating the Julian century this is used as a reference point. The J2000 frame's epoch is on January 1st, 2000, 12:00:00 PM UTC, which has a Julian date of 2451545.0. This is seen as the difference is taken between the desired date and the J2000 Julian date to calculate the quantity in terms of Julian century.

$$JD_{xx} = 367(yr) - INT\left\{\frac{7\left\{yr + INT\left(\frac{mo+9}{12}\right)\right\}}{4}\right\} + INT\left(\frac{275mo}{9}\right) + d + 1721013.5 + \frac{\frac{\left(\frac{\sec}{60} + \min\right)}{60} + hr}{24}$$

$$T_{xx} = \frac{JD_{xx} - 2451545.0}{36525} \tag{5.2}$$

where

- xx: Represents an arbitrary time standard discussed in section 5.1
- INT: Equivalent to the floor functionality which produces an integer value,
- JD_{xx}: Represents the Julian date quantity
- T_{xx}: Represent the Julian century quantity

5.2 Simplified General Perturbation (SGP) Model

The SGP model is orbit propagation, which hold the assumption the orbit has been approximated via TLEs. These TLEs are required to be updated once a day for the acceptable position and velocity values of a satellite. The common applications of orbit propagators include satellite tracking, monitoring changes in orbit, etc. However, this concept can also be incorporated to find the position and velocity of the satellite. An assumption is made to utilize this method. This assumption states, at initial satellite contact the orbital elements shall be updated via uplink command to update onboard parameters. The orbital parameters are reported as NORAD TLEs found on the CelesTrak's website [87] and are shown in Figure 5.1. Upon extraction, Table 5.2 shows the elements found in the TLE shown in Figure 5.1.

```
ISS (ZARYA)
1 25544U 98067A   16195.24184387  .00006167  00000-0  97898-4 0  9994
2 25544  51.6453 288.0416 0001507  55.3972 330.6245 15.54810785  9049
```

Figure 5.1: NORAD TLE of ISS Zarya from CelesTrak [87].

Table 5.2: Orbital elements and other parameters parsed and derived from the NORAD TLEs.

TLE Parameters	Values
Name of Satellite	ISS (Zarya)
Satellite Number	25544U/25544
International Designator	98067A
Epoch Year & Day Fraction	16, 195.24184387 (July 13th, 2016, 05:48:15.310 UTC)
1st derivative of Mean Motion/Ballistic coefficient (ndot) update this	0.00006167

2nd derivative of Mean Motion (nddot)	0
Argument of Perigee (ω)	55.3972 deg
Right Ascension of the Ascending Node (Ω) or (RAAN)	288.0416 deg
Mean Anomaly (M)	330.6245 deg
Mean Motion (n)	15.5481 rev/day
Inclination (i)	51.6453 deg
Eccentricity (e)	0.0001507

The orbital elements mentioned in the TLE are updated daily and are taken at the mentioned epoch which is described in terms of the last 2 digits of the year, and day fraction. Upon extraction, SGP process is initiated. The TLE values are described at time instance of t = 0 , which represents the epoch time mentioned in Table 5.2. The SGP process consists of multiple frame transformations that leads to describe the position and velocity in J2000 and ECEF frames.

Perifocal to TEME Transformation

The perifocal reference frame which is often referred as PQW coordinate frame, and it describes the satellite's motion with Earth being its center. Values such as true anomaly, radius of perigee, semi-major axis, argument of perigee, RAAN and eccentricity are incorporated to conduct the transformation to the TEME frame. As mentioned, both frames are intermediate frames which are further manipulated to define a true inertial frame for attitude estimation purposes. CelesTrak provides Space Track Report No 3 which outlines a set of equations that produces position and velocity outputs expressed in TEME reference frame [88]. Equations (5.3) to (5.10) demonstrated

the initializations of the following quantities:

- a_0 : Semi-major axis
- p_0 : Semi-latus rectum
- L_0 : Mean longitude
- q_0 : Radius of perigee
- $\frac{d\Omega}{dt}$: Rate of change in RAAN
- $\frac{d\omega}{dt}$: Rate of change in argument of perigee

$$a_1 = \left(\frac{\sqrt{Gm_e}}{n_0} \right)^{\frac{2}{3}} \tag{5.3}$$

$$\delta_1 = \frac{3}{4} J_2 \frac{r_e^{\,2}}{a_1^{\,2}} \frac{\left(3\cos^2 i_o - 1\right)}{\left(1 - e_o^{\,2}\right)^{\frac{3}{2}}} \tag{5.4}$$

$$a_0 = a_1 \left[1 - \frac{1}{3}\delta_1 - \delta_1^{\,2} - \frac{134}{81}\delta_1^{\,3} \right] \tag{5.5}$$

$$p_0 = a_0 \left(1 - e_0^{\,2}\right) \tag{5.6}$$

$$q_0 = a_0 \left(1 - e_0\right) \tag{5.7}$$

$$L_0 = M_0 + \omega_0 + \Omega_0 \tag{5.8}$$

$$\frac{d\Omega}{dt} = -\frac{3}{2} J_2 \frac{r_e^{\,2}}{p_0^{\,2}} n_0 \cos\left(i_o\right) \tag{5.9}$$

$$\frac{d\omega}{dt} = \frac{3}{4} J_2 \frac{r_e^2}{p_0^{\,2}} n_0 \left(5\cos^2 i_0 - 1\right) \tag{5.10}$$

The next step is to incorporate the secular effects caused by the perturbation forces generated by Earth's gravitational field and the atmospheric drag force. Moreover, the propagation where the orbital elements are extrapolated forward in time. Due to the uncertainty in the time of first contact with the satellite, factors such as long-term propagation, partially oscillating parameters, and short perturbation quantities are considered. These affect terms such as the semimajor axis vectors (a_x, a_y) and the mean longitude (L). The sum of the eccentric anomaly and the argument of perigee (ω) is solved using the Newton-Raphson technique. In which, a 95% confidence interval was chosen to hold the error within 5% of the true value to maintain the integrity of the solved quantity. Therefore, the tolerance is defined at $1x10^{-12}$.

$$a = a_0 \left[\frac{n_0}{n_0 + 2t\left(\frac{\dot{n}_0}{2}\right) + 3t^2\left(\frac{\ddot{n}_0}{6}\right)} \right]^{\frac{2}{3}} \tag{5.11}$$

$$p = a(1-e^2) \tag{5.12}$$

$$\Omega_{s0} = \Omega_0 + \frac{d\Omega}{dt} t \tag{5.13}$$

$$\omega_{s0} = \omega_0 + \frac{d\omega}{dt} t \tag{5.14}$$

$$L_s = L_0 + \left(n_0 + \frac{d\Omega}{dt} + \frac{d\omega}{dt} \right) t + \frac{\dot{n}_0}{2} t^2 + \frac{\ddot{n}_0}{6} t^3 \tag{5.15}$$

$$a_x = e\cos(\omega_{so})$$
$$a_y = e\sin(\omega_{so}) - \frac{J_3 r_e}{2J_2 p}\sin(i_0) \tag{5.16}$$

$$L = L_s - \frac{J_3 r_e}{J_2 p} a_x \sin(i_0)\left[\frac{3+5\cos(i_0)}{1+\cos(i_0)}\right] \tag{5.17}$$

$$U_{i-1} = L + \Omega_{so}$$
$$\Delta U = \frac{U - a_y\cos(U)_{i-1} + a_x\sin(U)_{i-1} - U_{i-1}}{1 - a_y\sin(U)_{i-1} - a_x\cos(U)_{i-1}} \tag{5.18}$$

$$U_i = U_{i-1} + \Delta U \text{ ; where } \|\Delta U\| \leq 1\times10^{-12} \tag{5.19}$$

$$e\cos(E) = a_x\cos(U) + a_y\sin(U)$$
$$e\sin(E) = a_x\cos(U) - a_y\sin(U) \tag{5.20}$$

$$e_l = \sqrt{a_x^{\,2} + a_y^{\,2}} \tag{5.21}$$

$$p_L = a\left(1 - e_L^{\,2}\right) \tag{5.22}$$

$$r = a\left(1 - e\cos(E)\right) \tag{5.23}$$

$$\dot{r} = \frac{\sqrt{aGm_e}}{r} e\sin(E) \tag{5.24}$$

$$r\dot{v} = \frac{\sqrt{p_L Gm_e}}{r} \tag{5.25}$$

$$\begin{aligned}
\sin(u) &= \frac{a}{r}\left[\sin(U) - a_y - a_x \frac{e\sin(E)}{1+\sqrt{1-e_L^{\,2}}}\right] \\
\cos(u) &= \frac{a}{r}\left[\cos(U) - a_x + a_y \frac{e\sin(E)}{1+\sqrt{1-e_L^{\,2}}}\right] \\
u &= \tan^{-1}\left(\frac{\sin(u)}{\cos(u)}\right)
\end{aligned} \tag{5.26}$$

$$\begin{aligned}
r_k &= r + \frac{1}{4} J_2 \frac{r_e^{\,2}}{p_L} \sin^2(i_0)\cos(2u) \\
u_k &= u - \frac{1}{8} J_2 \frac{r_e^{\,2}}{p_L^{\,2}} \left(7\cos^2(i_0) - 1\right)\sin(2u) \\
\Omega_k &= \Omega_{so} + \frac{3}{4} J_2 \frac{r_e^{\,2}}{p_L^{\,2}} \cos(i_0)\sin(2u) \\
i_k &= i_0 + \frac{3}{4} J_2 \frac{r_e^{\,2}}{p_L^{\,2}} \sin(i_0)\cos(i_0)\cos(2u)
\end{aligned} \tag{5.27}$$

Therefore, all the perturbating and oscillatory factors have now been considered. The major orbit propagation parameters resulting from incorporation of the secular effects are described as follows:

- t: time since epoch
- r_k : orbital radius in the PQW frame post R3(-ω) rotation and orbit propagation
- u_k : unit orientation angle post R3(-ω) rotation and orbit propagation
- Ω_k : RAAN post R3(-ω) rotation and orbit propagation
- i_k : Inclination post R3(-ω) rotation and orbit propagation

The final step is to convert the quantity in terms of TEME. A 3-1-3 rotational sequence is employed, where the argument of perigee is already considered through equations (5.11) to (5.27) [88] and shown in Figure 5.2. This aligns the satellite's orbital plane with the line of nodes. Finally, rotations with respect to the inclination and RAAN is required. This results in the desired output

of position and velocity expressed in the TEME reference frame, shown in (5.28) and (5.29).

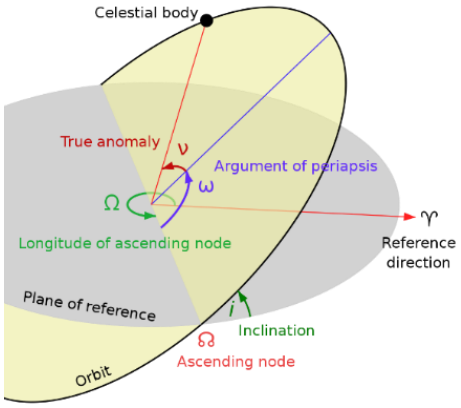


Figure 5.2: Demonstrates relation between rotation R3(-Ω) R1(-i) R3(-ω) [89].

$$r_{TEME} = r_k \left[\begin{bmatrix} -\sin(\Omega_k)\cos(i_k) \\ \cos(\Omega_k)\cos(i_k) \\ \sin(i_k) \end{bmatrix} \sin(u_k) + \begin{bmatrix} \cos(\Omega_k) \\ \sin(\Omega_k) \\ 0 \end{bmatrix} \cos(u_k) \right] \tag{5.28}$$

$$v_{TEME} = \dot{r} \left[\begin{bmatrix} -\sin(\Omega_k)\cos(i_k) \\ \cos(\Omega_k)\cos(i_k) \\ \sin(i_k) \end{bmatrix} \sin(u_k) + \begin{bmatrix} \cos(\Omega_k) \\ \sin(\Omega_k) \\ 0 \end{bmatrix} \cos(u_k) \right] + r\dot{v} \left[\begin{bmatrix} -\sin(\Omega_k)\cos(i_k) \\ \cos(\Omega_k)\cos(i_k) \\ \sin(i_k) \end{bmatrix} \cos(u_k) - \begin{bmatrix} \cos(\Omega_k) \\ \sin(\Omega_k) \\ 0 \end{bmatrix} \sin(u_k) \right] \tag{5.29}$$

where,

- r_{TEME} : Position in the TEME reference frame
- v_{TEME} : Velocity in the TEME reference frame

In order to prove the validly of these equations, an analysis was done using an external orbit analysis software called Systems Tool Kit (STK). Hence, it was chosen to be the validation medium to verify the results of the SGP propagator made in Simulink. The TLE described in Table 5.2 are the inputs to the STK scenario and the Simulink model. The simulation was done for 10 orbits to observe the variations in position and velocity in the TEME reference frame. The results are shown in Figures 5.3 and 5.4, respectively.

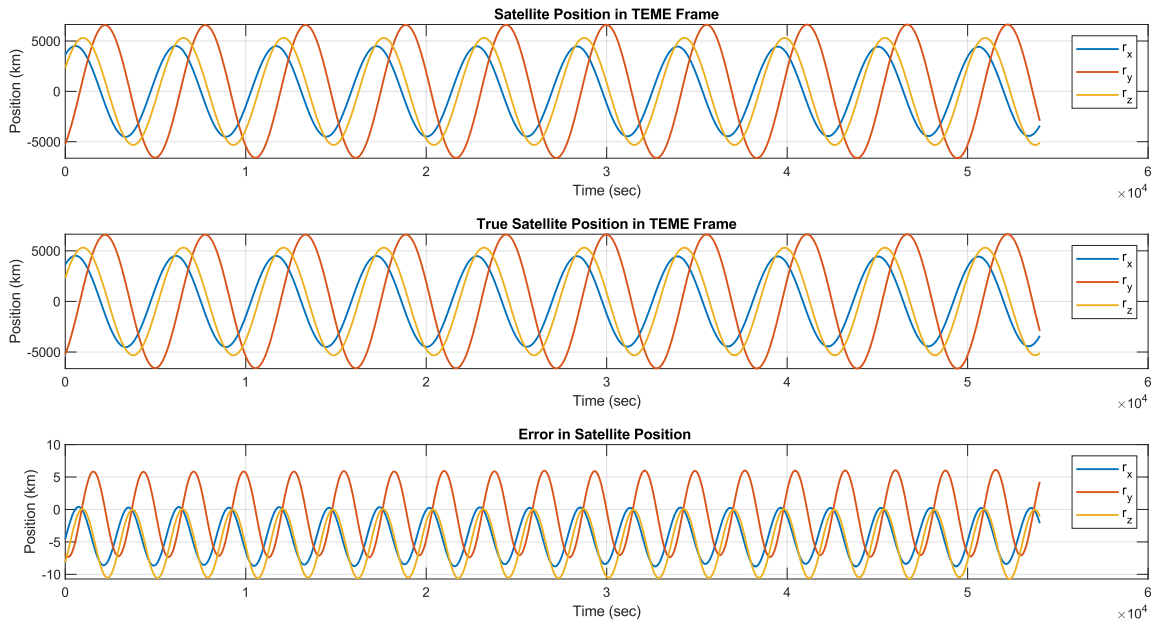


Figure 5.3: Position in TEME frame generated from the Simulink SGP model, STK orbit propagator and the difference between the true and simulated quantity.

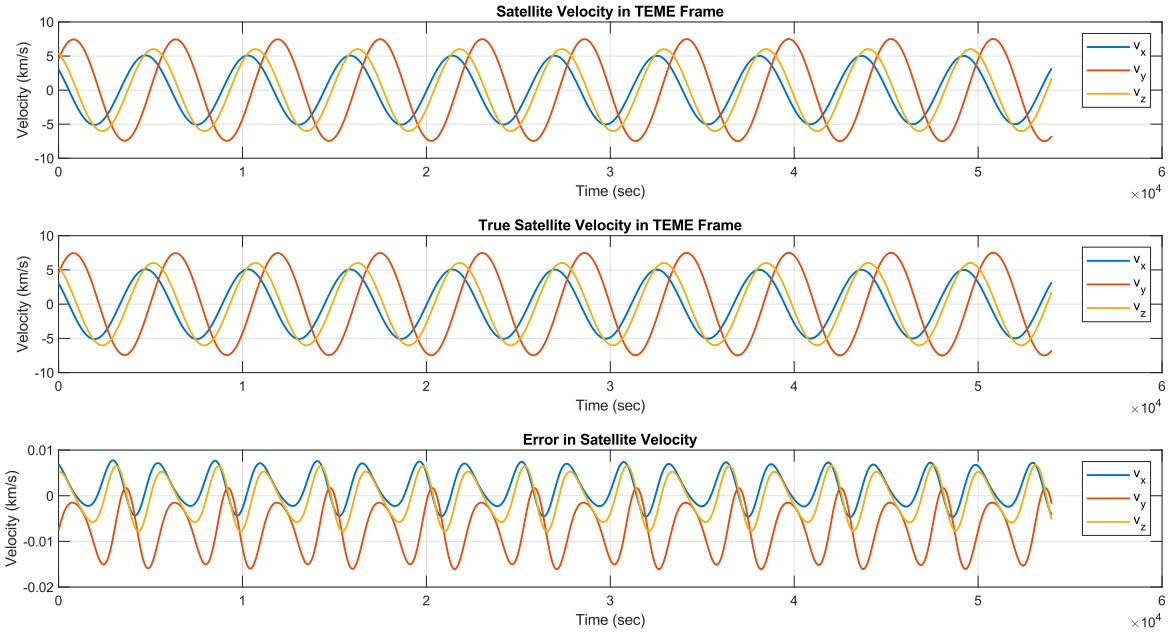


Figure 5.4: Velocity in TEME frame generated from the Simulink SGP model, STK orbit propagator and the difference between the true and simulated quantity.

The RMS error associated with the position output in TEME demonstrates that the accuracy is ~9.472 km. Moreover, the error demonstrates an oscillatory motion and shows a linear growth as time progresses. This is because the reliability of the propagator model degrades over time, as the orbital parameters vary at every epoch. Similar outcome is present for the error found in the velocity vector, but the linear growth in this error is much more evident. For the velocity error is ~0.011 km/sec on average, showing the accuracy level is at controllable limit for long durations. Thus solidifying the need of incorporation the variations found due to a long operational period derived in equations (5.16) to (5.27).

TEME to J2000

Most applications such as ground based satellite tracking systems assume TEME as the desired ECI frame. However, this frame approximates a truly inertial coordinate frame as the intent was to serve efficiency. However, a well-regarded inertial reference frame, ICRF/J2000 Frame was formulated on January 1st, 1998, by the International Astronomical Union (IAU) [90]. Using the Hipparions star catalogue, the J2000 frame axes are defined and described with respect to the quasars within +/- 20 microseconds [90]. To conduct the transformation between TEME to J2000 three axis fluctuating factors are required to be considered, precession, nutation, and equation of equinox [67].

Earth has two oscillating effects present at the planet's rotational axis due to the gravitational effects caused by the Moon, Sun, and all large celestial bodies in the solar system. First, the nutation phenomenon is a nodding motion on the Earth's rotational axis caused by the Moon. This is a tilt noticed in the J2000 plane when compared to the ecliptic plane with the Sun at its barycentre. The IAU 1980 nutation theory considers the motion of the primary celestial bodies to model this effect [91]. There are constants that are incorporated into this theory to produce the

nutation obliquity and longitude terms. These work in conjunction with the Delaunay parameters (5.31) defined at the J2000 epoch [67]. This is iterated through the all the 106 constants using (5.32) and produce the rotational angles with the correction parameters from the IERS bulletin A [86] using (5.33) . The output of this process is used to define the transformation matrix specific to Earth's nutation effect, (5.34) [67].

$$\tilde{\varepsilon}_{1980} = 23.439291 - 0.0130042T_{TT} - 1.64\times10^{-7}T_{TT}^{2} + 5.04\times10^{-7}T_{TT}^{3} \tag{5.30}$$

where $\tilde{\varepsilon}_{1980}$ is the mean obliquity in degrees, and T_{TT} is the Julian Century in terms of terrestrial time standard.

$$\begin{aligned}
l &= 134.96340251 + \left(1325r + 198.8675605\right)T_{TT} + 0.0088553T_{TT}^{2} + 1.434\times10^{-5}T_{TT}^{3} \\
l' &= 357.52910918 + \left(99r + 359.0502911\right)T_{TT} - 0.0001537T_{TT}^{2} + 3.8\times10^{-8}T_{TT}^{3} \\
F &= 93.27209062 + \left(1342r + 82.0174577\right)T_{TT} - 0.0035420T_{TT}^{2} - 2.88\times10^{-7}T_{TT}^{3} \\
D &= 297.85019547 + \left(1236r + 307.1114469\right)T_{TT} - 0.0017696T_{TT}^{2} + 1.831\times10^{-6}T_{TT}^{3} \\
\Omega &= 125.0445501 - \left(5r + 134.1361851\right)T_{TT} + 0.0020756T_{TT}^{2} + 2.139\times10^{-6}T_{TT}^{3}
\end{aligned} \tag{5.31}$$

where

- l: Mean Anomaly of the Moon in degrees
- l': Mean Anomaly of the Sun in degrees
- F: Difference between the Mean longitude of the Moon and Ω in degrees
- D: Mean Elongation of the Moon from the Sun in degrees
- Ω: Mean Longitude of the Ascending Node of the Moon in degrees
- r: 360 degrees

$$\Delta\Psi_{1980} = \sum_{i=1}^{106} \left(A_i + B_i T_{TT}\right) \sin\left(a_{p_i}\right)$$
$$\Delta\varepsilon_{1980} = \sum_{i=1}^{106} \left(C_i + D_i T_{TT}\right) \cos\left(a_{p_i}\right) \tag{5.32}$$
$$a_{p_i} = a_{1i} l + a_{2i} l^{'} + a_{3i} F + a_{4i} D + a_{5i} \Omega$$

where

- $\Delta\Psi_{1980}$: Nutation longitude in degrees
- $\Delta\varepsilon_{1980}$: Nutation obliquity in degrees
- A-D: IAU 80 nutation theory integer constants
- a_1-a_5: IAU 80 nutation theory real constants

$$\Delta\Psi_{1980} = \Delta\Psi_{1980} + \delta\Delta\Psi_{1980}$$
$$\Delta\varepsilon_{1980} = \Delta\varepsilon_{1980} + \delta\Delta\varepsilon_{1980} \tag{5.33}$$
$$\varepsilon_{1980} = \tilde{\varepsilon}_{1980} + \Delta\varepsilon_{1980}$$

where

- $\delta\Delta\Psi_{1980}$: EOP correction term for the nutation longitude parameter in degrees
- $\delta\Delta\varepsilon_{1980}$: EOP correction term for the nutation obliquity parameter in degrees

$$[N] = R_1\left(-\tilde{\varepsilon}_{1980}\right) R_3\left(\Delta\Psi_{1980}\right) R_1\left(\varepsilon_{1980}\right) \tag{5.34}$$

Precession is the next effect that is required to be considered when transforming to the J2000 frame. It is a wobble motion experienced around the rotational axis, similar to the rotation seen when spinning a top. This phenomenon occurs once again due to the tidal forces and the Earth's equatorial buldge. This is demonstrated in (5.35) and is described in terms of the 3 angles ς, Θ, z [67]. Thus, using these quantities the transformation matrix resulting from the precession effect

is shown in (5.36) [67].

$$\begin{aligned}\varsigma &= 0.6406161389T_{TT} + 8.38555556\times10^{-5}T_{TT}{}^{2} + 4.999444444\times10^{-6}T_{TT}{}^{3}\\ \Theta &= 0.5567530278T_{TT} - 1.185138889\times10^{-4}T_{TT}{}^{2} - 1.162027778\times10^{-5}T_{TT}{}^{3}\\ z &= 0.6406161389T_{TT} + 3.040777778\times10^{-4}T_{TT}{}^{2} + 5.056388889\times10^{-6}T_{TT}{}^{3}\end{aligned} \tag{5.35}$$

$$[P] = R_3(\varsigma)R_2(-\Theta)R_3(z) \tag{5.36}$$

where

- z : First precession rotational about Z-axis in degrees
- Θ : Second precession rotational angle about Y-axis in degrees
- ς : Third precession rotational angle about Z'-axis in degrees

In addition to the nutation and precession effects, it is required to consider the variation in equinox. As hinted, the equinox defined in the TEME frame is does not directly align with the vernal equinox. This parameter is a result of both earth's precession and nutation effects. Hence, the combined effect is considered and formulated by the equation of equinoxes. Using (5.37), a final transformation matrix is formed, as shown in (5.38) [67].

$$E_{q_{EQ}} = \Delta\Psi_{1980}\cos(\tilde{\varepsilon}_{1980}) \tag{5.37}$$

$$\begin{aligned}r_{J2000} &= [P][N]R_3(-E_{q_{EQ}})r_{TEME}\\ v_{J2000} &= [P][N]R_3(-E_{q_{EQ}})v_{TEME}\end{aligned} \tag{5.38}$$

Thus, using these effects, TEME is transformed using all three of these time varying quantities to the stable J2000 frame. These outputs were compared using STK for a time span of 10 orbits. Figure 5.5 and Figure 5.6 shows the position and velocity plots for the J2000 ECI frame.

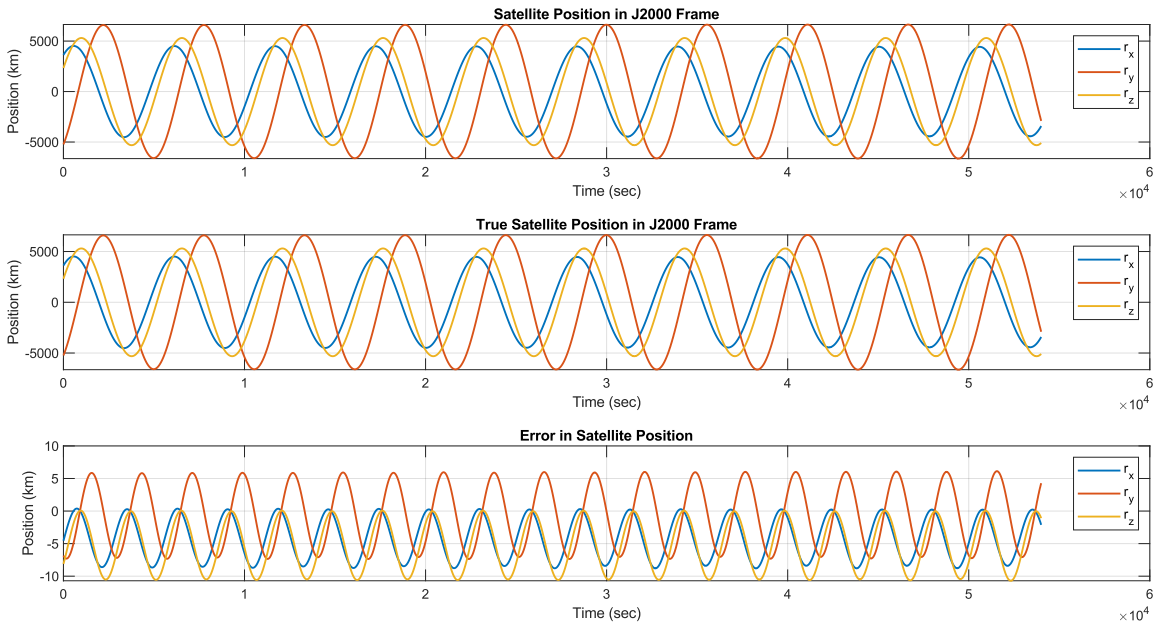


Figure 5.5: Position in J2000 frame generated from the Simulink SGP model, STK orbit propagator and the difference between the true and simulated quantity.

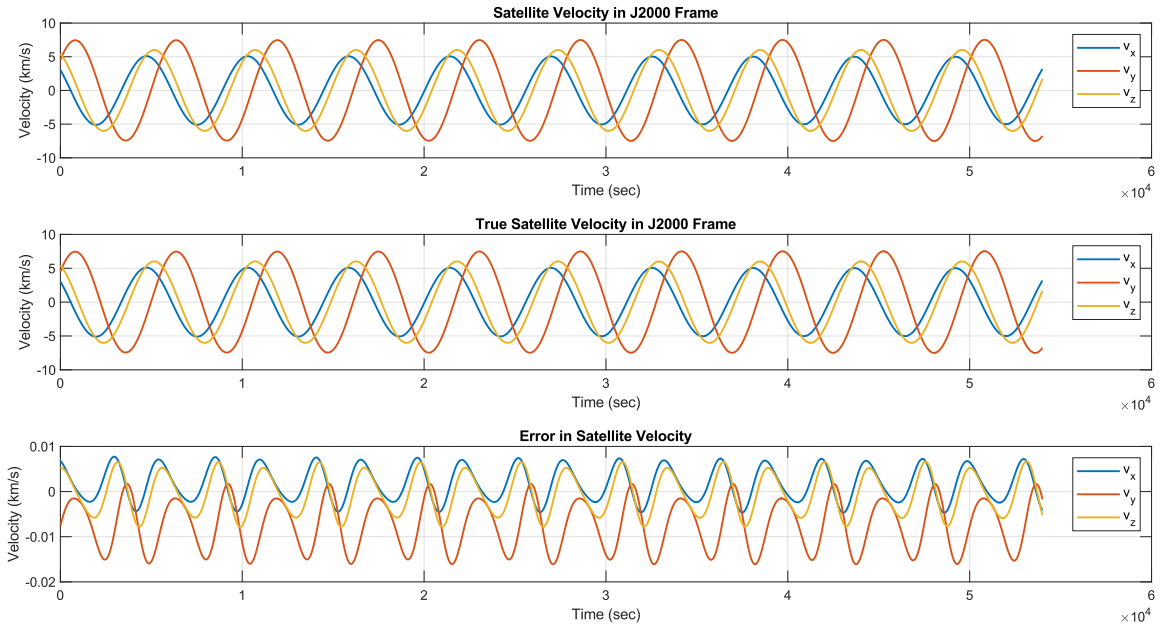


Figure 5.6: Velocity in J2000 frame generated from the Simulink SGP model, STK orbit propagator and the difference between the true and simulated quantity.

The analysis demonstrates that the error in both position and velocity oscillates and grows at a slow rate. This is similar to the plots seen in for the outputs when analyzing the TEME frame. The error in position and velocity is ~9.472 km and ~0.011 km/sec respectively. Due to the similarity in the TEME and J2000 frames, it is assumed both position and velocity derived in these frames are approximately the same. This is expected since the precession and nutation effects are driven by the Julian Century calculation. Thus, a variation in the SGP model will be noticeable if it was continuously operated for several years. This entails equating the TEME and J2000 frames is reasonable within an acceptable operational timespan.

J2000 to ECEF

The ECEF frame is one of the most critical reference frames for LEO satellite applications as environmental models such as the magnetic field model are realized in this frame. From the J2000 frame, two more transformations parameters are required to be considered. Because this is a moving frame, a geographic location is referenced and represented in terms of an angle, and polar effects are considered. The location is Greenwich, England at it is located at 0 longitude. Thus, the angle used to represent these effects are defined as the Greenwich apparent sidereal time (GAST).

In addition to the wobble noticed, the Earth's rotational axis moves along a circular path over time. Polar motion describes this motion and is modelled in terms of two angles, x_p and y_p, and are measured along the prime meridian and 90^o W meridian [92]. To formulate this motion, IERS provides coefficients that form the functions for these angles. Similar to the previously mentioned quantities, they are dependent on the Julian date. However, a slight modification is made as the time measurement is described in terms of a Modified Julian date (MJD). MJD calculation is shown in (5.39), and is incorporated in (5.40) to calculate these angles. Following these derivations, the transformation is displayed using (5.41).

$$MJD = JD - 2400000.5 \tag{5.39}$$

$$\begin{aligned} x_p &= \frac{1}{3600}\left(0.1250 - 0.0667\cos(A) - 0.0833\sin(A) + 0.0144\cos(C) + 0.0162\sin(C)\right) \\ y_p &= \frac{1}{3600}\left(0.3653 - 0.0760\cos(A) + 0.0597\sin(A) + 0.0162\cos(C) + 0.0144\sin(C)\right) \end{aligned} \tag{5.40}$$

$$A = \frac{2\pi(MJD - 59214)}{365.25}$$
$$C = \frac{2\pi(MJD - 59214)}{435}$$

$$[PM] = R_1(y_p)R_2(x_p) \tag{5.41}$$

Since the ECEF frame if fixed to the Earth, it is required to have a reference point between the J2000 and the ECEF frames. This is done by defining angle between the J2000 and ECEF frame's X axis. The angle is known as the Greenwich Meridian Sidereal Time (GMST). Due to the presence of the variation of equinox correction term, GAST is incorporated. The GAST is derived from the GMST, hence it is a function of UT1 time standard. Thus, using (5.1), the GAST calculation is shown in (5.42) and the rotational matrix is shown in (5.43) [67].

$$\begin{aligned} \theta_{GMST} &= \frac{1}{240}\left[67310.54841 + (876600 + 8640)T_{UT1} + 0.093104T_{UT1}^2 - 6.2\times10^{-6}T_{UT1}^3\right] \\ \theta_{GAST} &= \theta_{GMST} + E_{q_{EQ}} \end{aligned} \tag{5.42}$$

$$[R] = R_3(-\theta_{GAST}) \tag{5.43}$$

where

- θ_{GMST} : Greenwich Meridian Sidereal Time in degrees
- θ_{GAST} : Greenwich Apparent Sidereal Time in degrees

- T_{UT1}: Julian Century calculated in terms of UT1 time standard

Due to the similarities between the TEME and J2000 frames, the polar motion and the GAST transformation matrices are considered along with the three matrices expressed in (5.38). All the angles are converted to radians after calculation, because STK utilizes radians when deriving position and velocity outputs. Once all the mentioned matrices are formed, J2000 position is transformed into the ECEF frame using (5.44). However, to form the velocity in the ECEF, further derivation is required to express the change with respect to the J2000 inertial frame. Using the value of length of day (LOD) ~-0.2431 ms, from the IERS bulletin B [86], earth's angular velocity is calculated using (5.45) [67]. Therefore, velocity from the inertial frame is formed to convert to the ECEF frame (5.46) [67].

$$r_{ECEF} = [PM]^T [R]^T [N]^T [P]^T r_{J2000} \tag{5.44}$$

$$\omega_e = 7.292115146706979 \times 10^{-5} \left[1 - \frac{LOD}{86400} \right] \tag{5.45}$$

$$v_{ECEF} = [PM]^T \left\{ [R]^T [N]^T [P]^T v_{J2000} - \omega_e \times \left\{ [R]^T [N]^T [P]^T r_{J2000} \right\} \right\} \tag{5.46}$$

To validate these values, a comparison with STK was again conducted for a duration of 10 orbits. The position and velocity plots are shown in Figures 5.7 and 5.8, respectively.

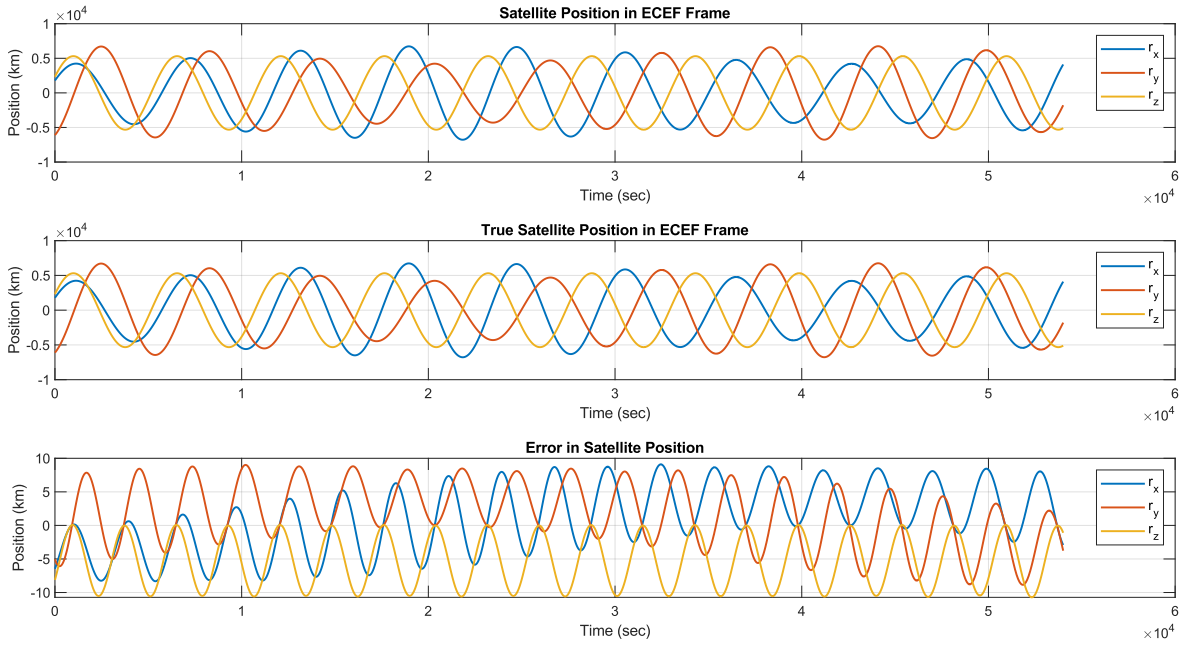


Figure 5.7: Position in ECEF frame generated from the Simulink SGP model, STK orbit propagator and the difference between the true and simulated quantity.

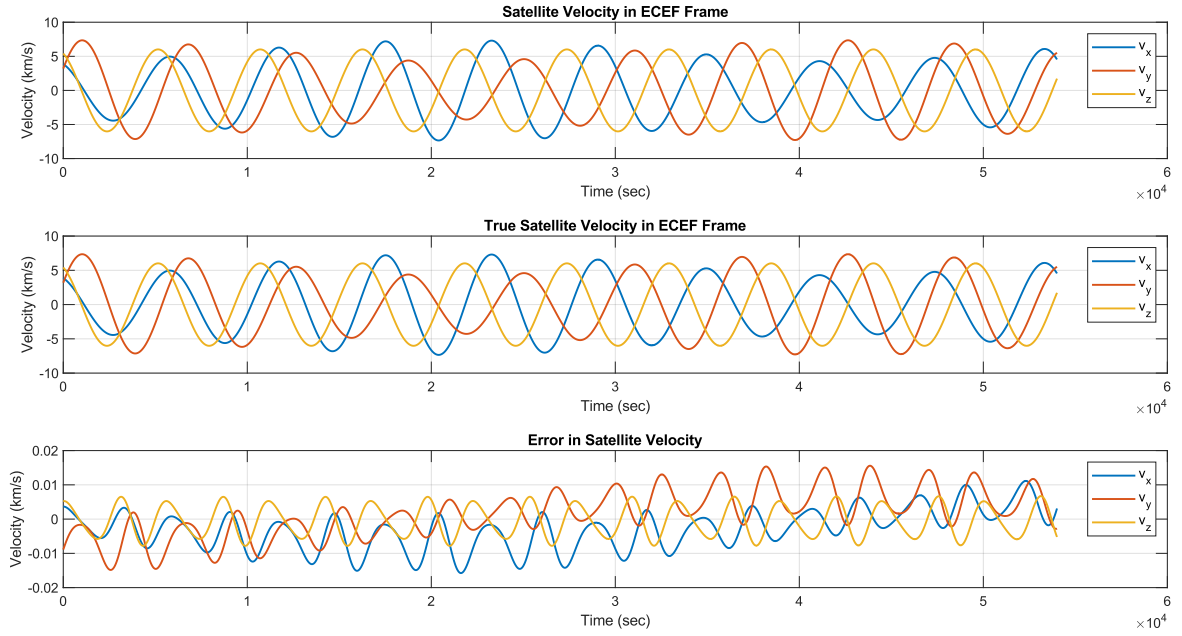


Figure 5.8: Velocity in ECEF frame generated from the Simulink SGP model, STK orbit propagator and the difference between the true and simulated quantity.

Due to the secular variations, the error in both position and velocity oscillate and grow slowly over time. When the error plots are compared to the TEME and J2000 coordinate frames, the RMS error in both quantities is approximately the same. The errors values for position and velocity are ~9.472 km and ~0.011 km/sec respectively. The error tends to occur due to the variation of equinox as the GMST angle is dependent on the J2000 frame's X-axis. Thus, all the EOP parameter mentioned are critical to produce accurate transformations over multiple reference frames.

In conclusion, the determination of position and velocity of a LEO satellite was successfully performed using SGP model and tested with STK to ensure it is within a controllable error value. The error in position on average is observed to be within 10 km for all geocentric reference frames. Thus, making the error negligible considering the distance to the satellite in ~422 km in altitude. This further increases the confidence in the usage of this propagator on the satellite.

5.3 Onboard GPS receiver centric Kalman Filter

GPS receivers are often used in satellite applications to produce position and velocity measurements. A method of trilateration in utilized which uses a minimum of four GPS satellites to accurately produce the desired quantities. The measurements are always in the ECEF frame as the GPS satellites are also used to locate places, objects, etc. In addition to the mentioned orbit propagator, a linear Kalman filter is used to produce real time orbit determination method. This robust process is used on rovers, UAV, etc. Figure 5.9 shows the general software algorithm used for a typical Kalman Filter to model the translational motion of a rigid body.

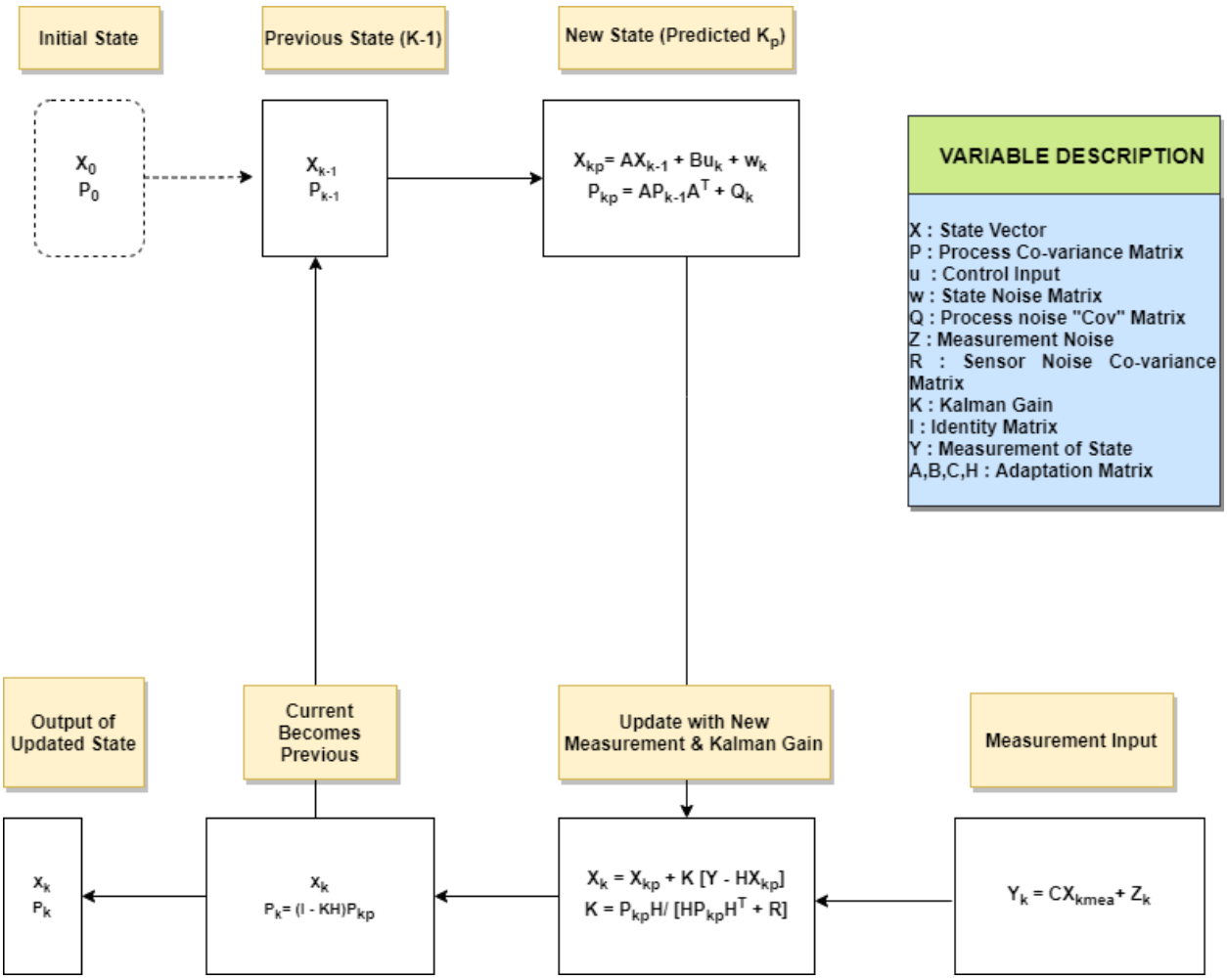


Figure 5.9: GPS Receiver based Kalman Filter's Software Algorithm Schematic.

The filter requires an initialization step which consists of initial estimates that are corrected in a continuous fashion. The GPS receiver measurement model discussed in Chapter 4 is incorporated in this filter, as it is the sensor relevant to this problem. To describe a translation motion of a rigid body three quantities are required, position, velocity, and acceleration. These are further characterized as to form system of equations for the GPS filter. (5.47) and (5.48) shows the equations for translational motion for position and velocity. To derive acceleration, it is critical to understand the motion. Due to the conservation of momentum and energy laws, it is interpreted as

a two-body problem. Thus, using (5.49) the acceleration of the satellite is modelled, as the force of gravity is the primary force manipulating the motion [93].

$$r_i = r_{i-1} + v_{i-1}\Delta t + \frac{1}{2} a_{i-1}\Delta t^2 \tag{5.47}$$

$$\dot{r}_i = v_i = v_{i-1} + a_{i-1}\Delta t \tag{5.48}$$

$$\ddot{r}_{i-1} = a_{i-1} = \frac{-Gm_e}{\left\| r_{i-1} \right\|^3} r_{i-1} \tag{5.49}$$

where

- r : 3 x 1 position vector in the ECEF frame in km
- v : 3 x 1 velocity vector in the ECEF frame in km/sec
- a: 3 x 1 acceleration vector in the in the ECEF frame in km/sec^2
- Δt : filter time interval/timestep
- m_e: Mass of the Earth, 5.973 x 10^{24} kg
- G: Universal Gravitational Constant: 6.674 x 10^{-11} m^3/kg sec^2

The state vector for the Kalman filter is shown in (5.50) which defines the position and velocity of the satellite. Moreover, the control vector that describes the acceleration of the system is found in (5.51).

$$X = \begin{bmatrix} r_x & r_y & r_z & \dot{r}_x & \dot{r}_y & \dot{r}_z \end{bmatrix} \tag{5.50}$$

$$u = \begin{bmatrix} \ddot{r}_x & \ddot{r}_y & \ddot{r}_z \end{bmatrix} \tag{5.51}$$

There are 3 main adaptation matrices incorporated shown in Figure 5.9. The first adaption matrix

is called the state transition matrix (A) which is one of two matrices to propagate the state forward in time. It is formulated using the system of equations shown in (5.47) to (5.49), and the matrix is shown in (5.52). It can be seen the contents in the matrix represent the coefficients of the position and velocity equations, because this is a time varying algorithm.

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.52}$$

Furthermore, the control matrix (B) used to propagate the control vector to propagate the motion is outline in (5.53). This works in conjunction with the A matrix as the state and the control vectors define the translation motion.

$$B = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 & 0 \\ 0 & \frac{1}{2}\Delta t^2 & 0 \\ 0 & 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \tag{5.53}$$

The final adaptation matrix is used to relate the measurements to the states. Fortunately, the states have a direct correlation with the measured quantities. Thus, the quantity is simply an identity matrix as shown in (5.54). Lasty, the process noise matrix is used to characterise unmodelled noise that are present in the system, found in (5.55). This parameter is often tuned to ensure the filter does not reach singularity.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.54}$$

$$Q = \begin{bmatrix} 1\times10^{-8} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1\times10^{-8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1\times10^{-8} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1\times10^{-8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1\times10^{-8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1\times10^{-8} \end{bmatrix} \tag{5.55}$$

Table 5.3 outlines the initialized parameters which are fed to the Simulink model where the Kalman filter is formed. Figure 5.10 shows the Simulink model which contains the blocks created to propagate the state vector and update it using the GPS receiver. The sensor noise covariance matrix is driven off the RMS noise values taken from the GPS receiver's datasheet, as shown in Chapter 4. The Kalman Filter shown operates at 10 Hz to compare the results with true position and velocity produced by the satellite simulator. This update frequency was chosen, as the ESSENCE mission required such sensor polling rate to prove the effectiveness of the filter. The relevant requirement for this selection is defined as: RQR-SYS-CDH-161 ESSENCE ADCS shall be operating at a frequency of 10 Hz.

Table 5.3: Initial parameters defined for the GPS- based Kalman Filter

Parameters	**Values**
Initial Position State (X_p)	X_p = [1794 ;-6109 ;2323] km
Initial Velocity State (X_v)	X_v = [3.87 ;3.19 ;5.41] km/sec
Initial Covariances (P_{i-1})	$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
Sensor Noise Covariance (R_{GPS})	$R_{GPS} = \begin{bmatrix} 6.4\times10^{-9} & 0 & 0 & 0 & 0 & 0 \\ 0 & 6.4\times10^{-9} & 0 & 0 & 0 & 0 \\ 0 & 0 & 6.4\times10^{-9} & 0 & 0 & 0 \\ 0 & 0 & 0 & 9\times10^{-10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 9\times10^{-10} & 0 \\ 0 & 0 & 0 & 0 & 0 & 9\times10^{-10} \end{bmatrix}$
Filter timestep (Δt)	10 Hz or 0.1 sec

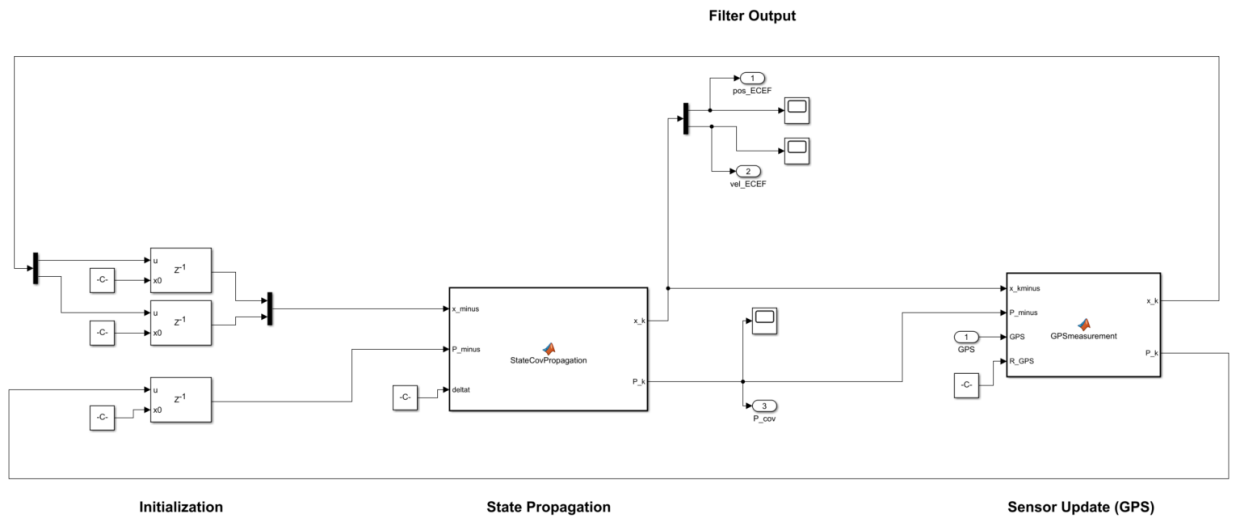


Figure 5.10: Simulink Model generated for the GPS Receiver based Kalman Filter.

Figures 5.11 and 5.12 show the position and velocity plots respectively in the ECEF frame. True values are also presented along with the error associated.

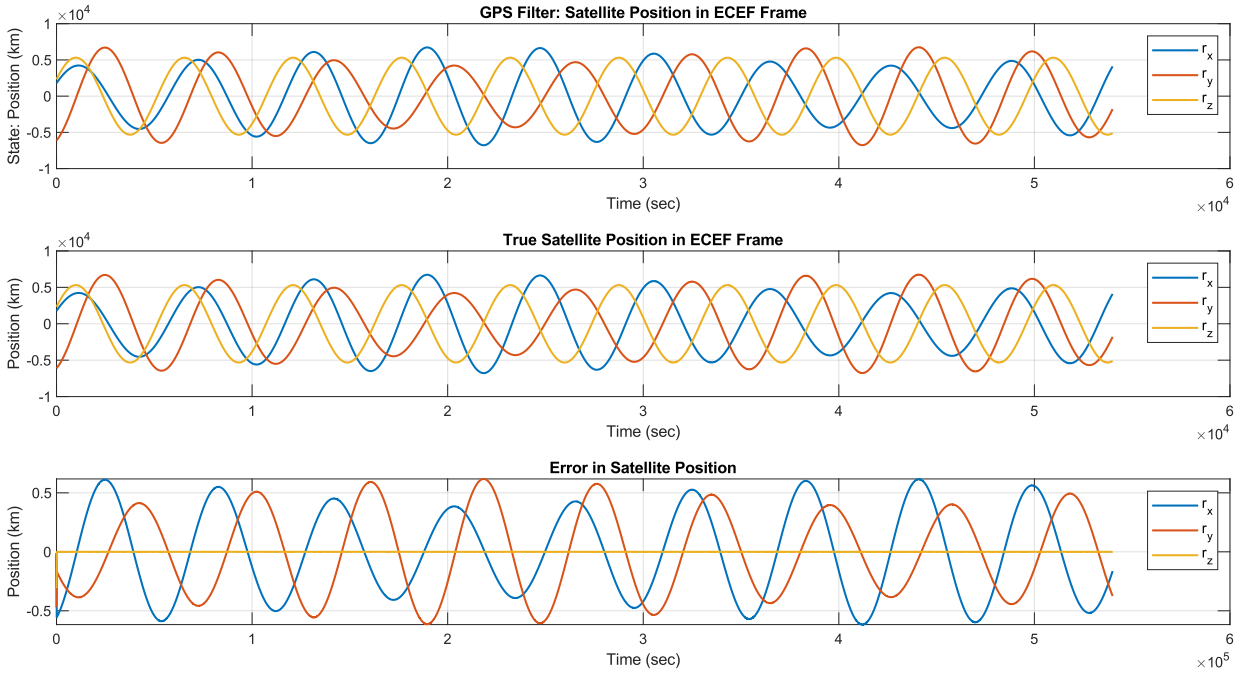


Figure 5.11: Position in ECEF frame generated from the Simulink GPS-based Kalman Filter, STK orbit propagator and the difference between the true and simulated quantity.

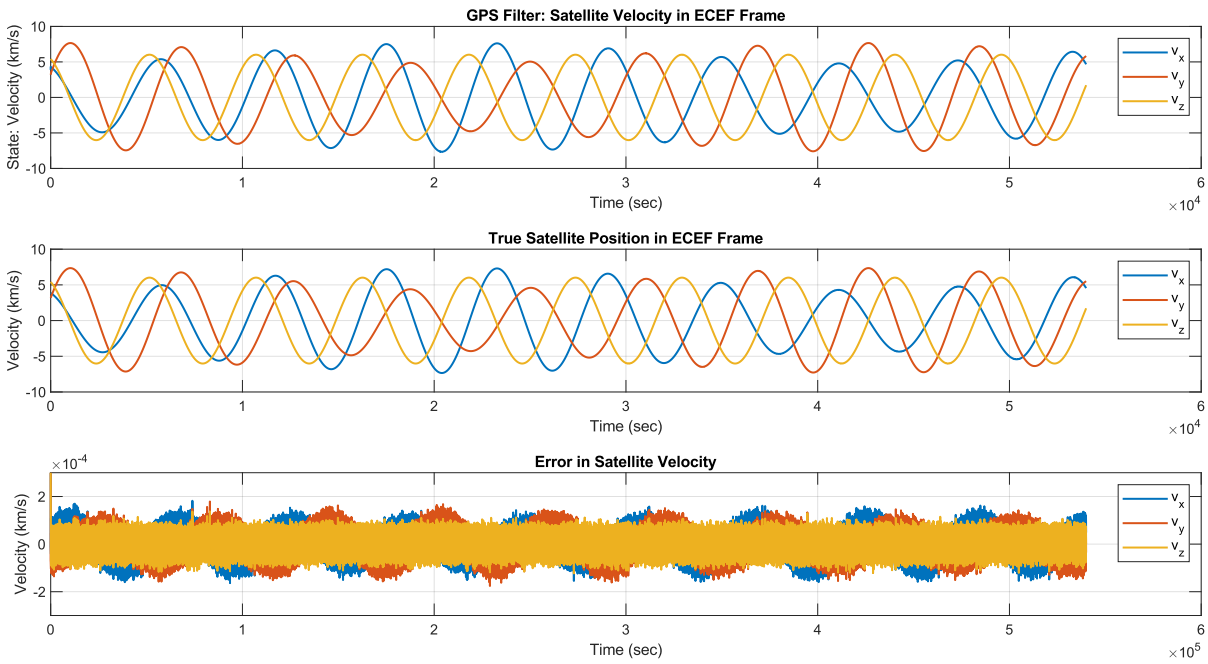


Figure 5.12: Velocity in ECEF frame generated from the Simulink GPS-based Kalman Filter, STK orbit propagator and the difference between the true and simulated quantity.

The above plots demonstrate a clear increase in accuracy for both position and velocity vectors in the ECEF frame. The RMS error in terms of magnitude for position is found to be 0.5133 km which is significantly more accurate than the SGP model. Similarly, velocity has a RMS error of 6.2318 x 10^{-4} km/sec. However, the error does not grow over time as the Kalman Filter improves the accuracy of the system model as measurement updates reduce the error in the states. The SGP propagator does not have the capability to account for the drift, unless the TLEs are updated at a daily rate. This is a potential single point failure, as there are instances for various amateur satellite missions where initial contact may occur after several weeks. This is the primary advantage of using a Kalman Filter, as GNSS/GPS receivers produce accurate results to conduct real time orbit propagation.

5.4 Environmental Models

The main purpose of determining the satellite's position and velocity is to aid the attitude determination system, as these vectors are required to be used to produce key outputs from environmental models. The environmental models that are used are the Sun position and the Magnetic field vector models.

5.4.1 Sun Position Model

The sun position model is defined with respect to the Mean-equator of Date (MOD) reference frame [67]. This is also an intermediate frame used in modelling the IAU-76/FK5 standard [67]. The only effect considered when transforming is precession. The sun position is first calculated using (5.56), in the MOD frame. Therefore, further transformation is required as defined in (5.57) to convert the quantity in the J2000 frame.

$$\begin{aligned}
&\lambda_{l'} = 280.460 + 3600.771 T_{UT1} \\
&M_{l'} = 357.5291092 + 35.99905034 T_{UT1} \\
&\lambda_{ecliptic} = \lambda_{l'} + 1.914666471 \sin\left(M_{l'}\right) + 0.019994643 \sin\left(2M_{l'}\right) \\
&\varepsilon = 23.439291 - 0.0130042 T_{UT1} \\
&r_{mag} = 1.000140612 - 0.016708617 \cos\left(M_{l'}\right) - 0.000139589 \cos\left(2M_{l'}\right)
\end{aligned} \tag{5.56}$$

$$r_{sun,MOD} = \begin{bmatrix} r_{mag} \cos\left(\lambda_{ecliptic}\right) \\ r_{mag} \cos\left(\varepsilon\right) \sin\left(\lambda_{ecliptic}\right) \\ r_{mag} \sin\left(\varepsilon\right) \sin\left(\lambda_{ecliptic}\right) \end{bmatrix} AU$$

$$r_{sun,J2000} = \left[P\right] r_{sun,MOD} \tag{5.57}$$

where

- $\lambda_{l'}$: Mean longitude of the Sun in MOD frame

- $M_{\dot{l}}$: Mean anomaly of the Sun in MOD frame
- $\lambda_{ecliptic}$: Ecliptic latitude of the Sun in MOD frame
- ε : Obliquity of the ecliptic plane
- $r_{sun,MOD}$: Sun position in AU defined in MOD frame
- $r_{sun,J2000}$: Sun position in AU defined in J2000 frame
- AU: Astronomical Unit 1 AU = 1.50 x 10^{11} m

The procedure to determine sun position is function of Julian Century. In addition, eclipse times are also required to be calculated to dictate the use of the sun sensors illustrated in Chapter 4. The eclipse zone is distributed in two domains penumbra and umbra, as shown in Figure 5.13. Umbra is when the orbiting satellite is in a total eclipse by the primary body, Earth. Whereas penumbra is when it is in a partial eclipse in which the satellite is either entering or exiting the umbra region. Using algorithm 34 in [67] it is defined as, if the dot product between the sun position and satellite position is less than 0, it is in an eclipse region.

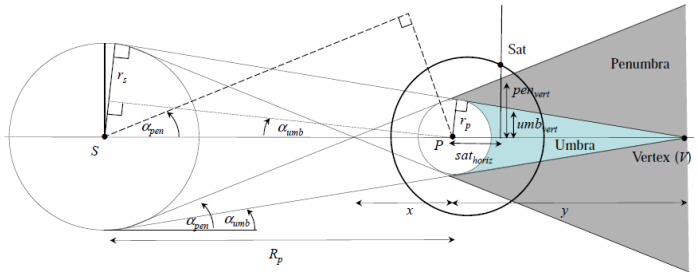


Figure 5.13: Illustrations of the penumbra and umbra regions while the orbiting satellite enters an eclipse zone [67].

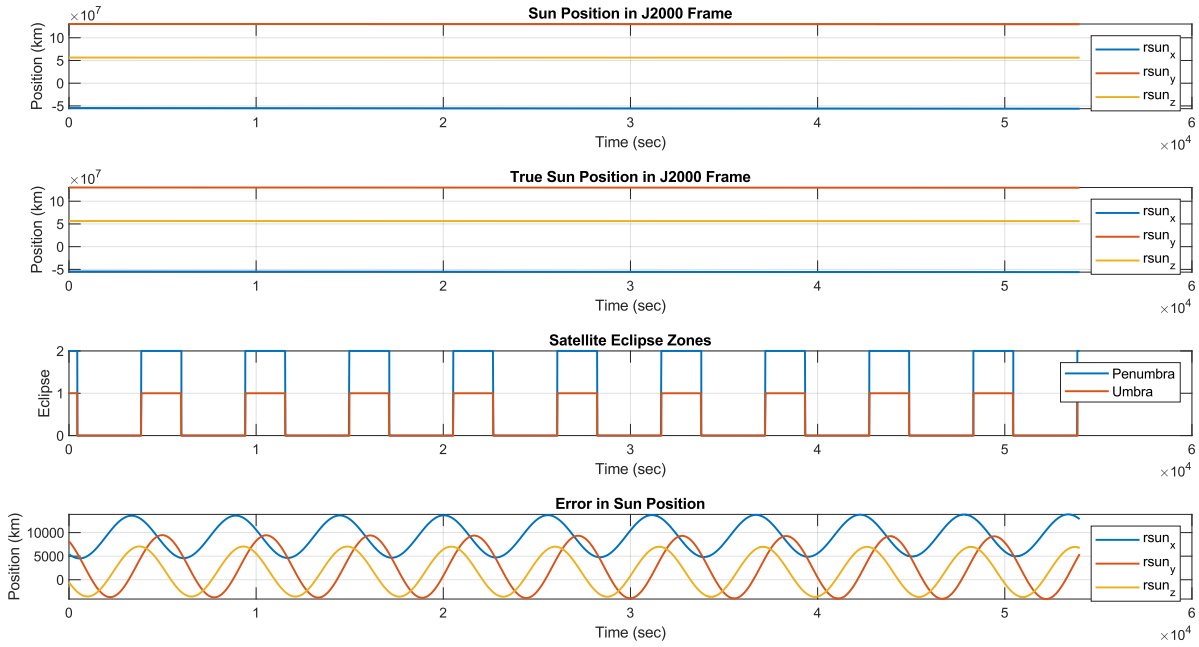


Figure 5.14: Sun position using equations (5.56) and (5.57), True Sun position from STK, eclipse zones, and the error present in calculation.

Figure 5.14 shows the sun position and its comparison with the STK simulation for a 10-orbit timespan. It can be seen the position seems to be constant, but slight variations are found as shown in the error plot. The error in terms of magnitude is 9.7275 x 10^3 km. This a negligible amount considering the distance from Earth to the Sun is 1 AU or 1.50 x 10^{11} m. Lastly, the eclipse flags show when the value is equal to 1 it is in umbra and when value is 2 it is in penumbra. An overlap is common as when the satellite enter the umbra region is also invariantly in the penumbra region. The angular difference between the two regions is ~0.005 degrees, which is why a common assumption is to conclude when the satellite enters a partial eclipse, it is to be interpreted as a full eclipse. The total duration in eclipse lasts for ~35.72 mins in an orbit, due to it high orbiting speeds and orbit placement. Within that window the umbra regions is for ~35.45 mins, hence validating the small angular difference shown in Figure 5.13.

5.4.2 Magnetic Field Vector Model

Earth can be perceived as an extremely large magnet, consisting of a north and south magnetic pole. These poles differ drastically from the geographic north and south poles, as they align with Earth's rotational axis. Low earth orbit missions such as MAGSAT, SAC-C and Swarm were lauched to observe the Earth's magnetic field [94]. These continuous earth surveying missions, help succeed magnetic field models such as the World Magnetic Model (WMM) and the International Geomagnetic Reference Field (IGRF) model. The models are formulated using a spherical harmonics mathematical model to produce magnetic potential vector. Both WMM and IGRF contain magnetic spherical or gaussian coefficients (g and h), but of different values. The IGRF model is used for this this, but both models produce acceptable magnetic field values. The International Association of Geomagnetism and Aeronomy (IAGA) supports the IGRF model and is updates the coefficients every 5 years. The current model is the 13th version referred to as IGRF-13 model for the duration of 2020-2025.

The mathematical model of IGRF-13 uses a potential function of the magnetic field, shown in (5.58) [95].The earth's magnetic field varies based on the location of this ellipsoid. Thus, geodetic measurements are employed, implying this calculation is conducted in the ECEF frame.

$$B = -\nabla V$$
$$V = a\sum_{n}^{N}\left(\frac{a}{R}\right)\sum_{m=0}^{n} g_n^m \cos\left(m\phi\right) + h_n^m \sin\left(m\phi\right) P_n^m \cos\left(\theta\right) \tag{5.58}$$

where

- B: Magnetic Field vector differentiated with respect to R in T
- a: Earth's magnetic reference radius 6371.2 km

- R: Magnitude of the satellite's position in ECEF frame
- g_n^m, h_n^m : IGRF spherical harmonics Gaussian coefficients in nT
- $P_n^m \cos(\theta)$: Schmidt quasi-normalized Legendre function
- ϕ: Geodetic longitude
- θ: Geodetic co-latitude
- N: Max degree for the current IGRF-13 model
- n,m : represent the degree and order of the series

However, this can be further simplified as shown in (5.59) to simplify the complex means of interpreting the Legendre functions [96]. These equations make the entire magnetic field vector calculation dependent on the latitude and the ECEF position vector.

$$B_{ECEF} = -\nabla V = \sum_{n}^{N}\sum_{m}^{n} B_{n,m}$$

$$B_{n,m} = \frac{K_{n,m} a^{n+2}}{R^{n+m+1}} \begin{Bmatrix} \frac{g_{n,m}C_m + h_{n,m}S_m}{R}\left[\left(\sin(\lambda) A_{n,m+1} + (n+m+1) A_{n,m}\right)\hat{r} - A_{n,m+1}\hat{e}_3\right].. \\ -mA_{n,m}\left[\left(g_{n,m}C_{m-1} + h_{n,m}S_{m-1}\right)\hat{e}_1 + \left(h_{n,m}C_{m-1} - g_{n,m}S_{m-1}\right)\hat{e}_2\right] \end{Bmatrix} \quad (5.59)$$

where

- B_{ECEF}: Magnetic Field vector in ECEF frame in T
- K : Schmidt function coefficient
- A : Derived Legendre polynomials defined in [96]
- C,S : Coefficients associated with the recursive Legendre polynomial defined in [96]
- $\hat{e}_1, \hat{e}_2, \hat{e}_3$: Unit vectors along the ECEF reference frame axis
- $\hat{r}$: ECEF position unit vector

- λ : Geodetic latitude

Therefore, using (5.59) the output results in the magnetic field vector in the ECEF. The coefficient are required to be propagated as time progresses. The IGRF coefficients found in [95] also contain rate of change in both g and h harmonic coefficients. This is modelled in (5.60), as it is assumed to vary linearly as a function of time [97].

$$\begin{aligned} g_n^m(t) &= g_n^m(T_0) + \dot{g}_n^m(T_0)(t - T_0) \\ h_n^m(t) &= h_n^m(T_0) + \dot{h}_n^m(T_0)(t - T_0) \end{aligned} \tag{5.60}$$

where

- $\dot{g}_n^m, \dot{h}_n^m$: Rates of change of the two Gauss spherical harmonics coefficients given in nT/yr
- t : Current Gregorian Year, time since epoch
- T_0 :Base Gregorian Year (2020)

However, this measurement is made relevant for attitude estimation by transforming this to the J2000 frame. (5.61) shows the transformation matrix which is the inverse of the product of matrix multiplication expressed in (5.44).

$$\begin{aligned} TM_{ECEF}^{J2000} &= [P][N][R][PM] \\ B_{J2000} &= TM_{ECEF}^{J2000} B_{ECEF} \end{aligned} \tag{5.61}$$

The IGRF-13 model was modelled in Simulink for both SGP orbit propagation model and the GPS-based Kalman Filter. Figure 5.15 shows the incorporation of this in the SGP method to calculate the position of a LEO satellite. In addition, Figure 5.16 shows the model with the GPS-based Kalman Filter.

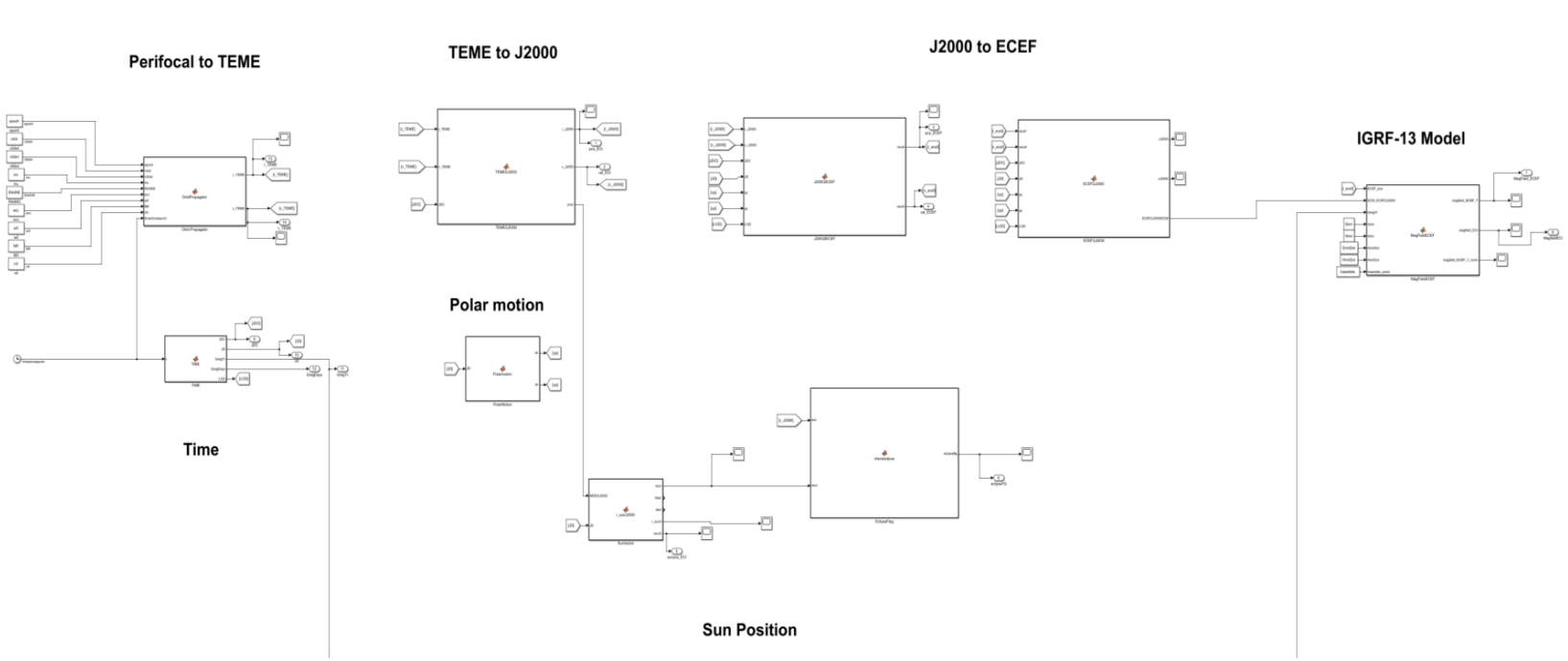


Figure 5.15: SGP model built in Simulink to produce Magnetic Field and Sun position vectors.

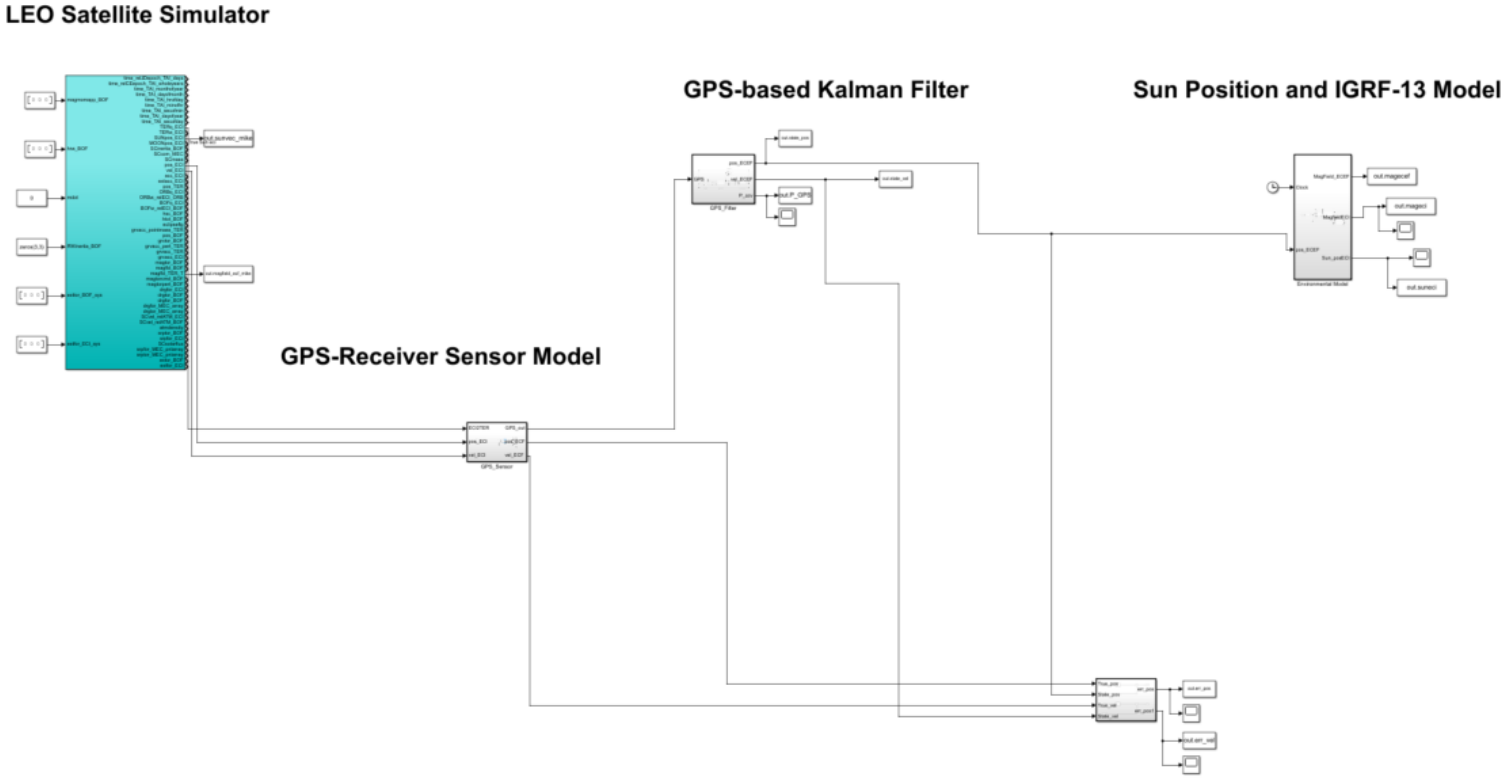


Figure 5.16: GPS-based Kalman Filter built in Simulink to produce Magnetic Field and Sun position vectors.

Both of the presented methods produced the required inputs for the IGRF-13 model to generate the magnetic field vector. Figure 5.17 shows the predicted magnetic field, true magnetic field, and the error in the magnetic field vectors. Moreover, Figure 5.18 shows the magnetic field vector in the J2000 frame which is used for various attitude determination algorithms.

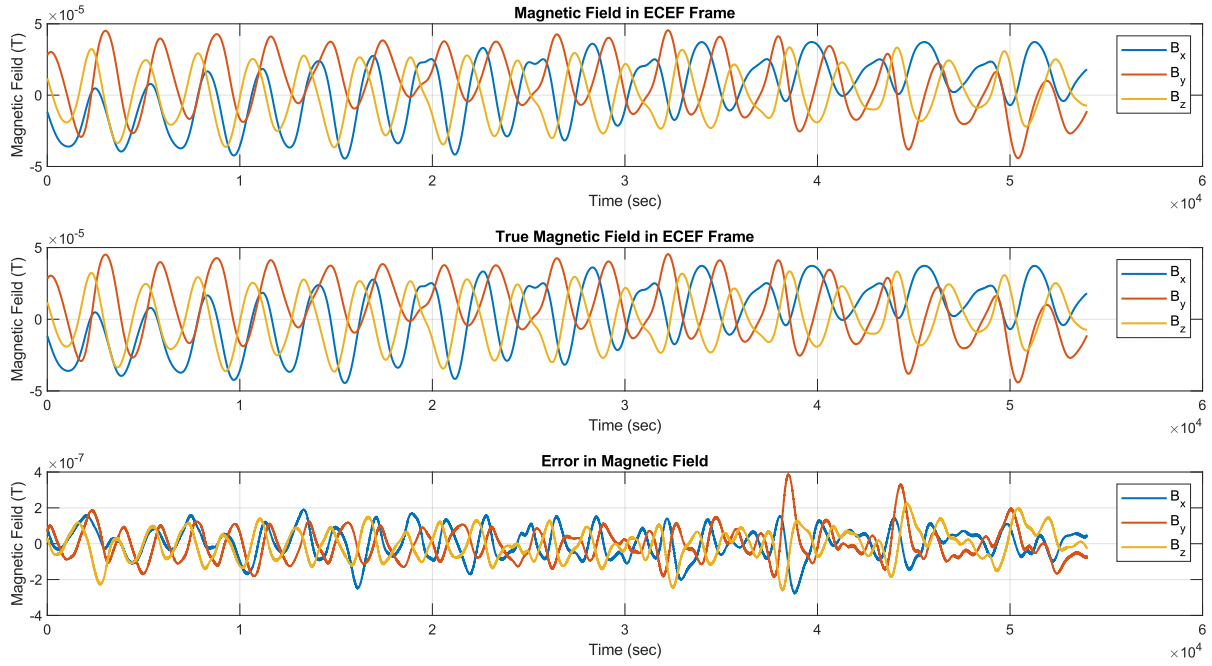


Figure 5.17: Shows the Magnetic field generated by both orbit determination methods in the ECEF frame, True Magnetic Field, and error in the measured quantity.

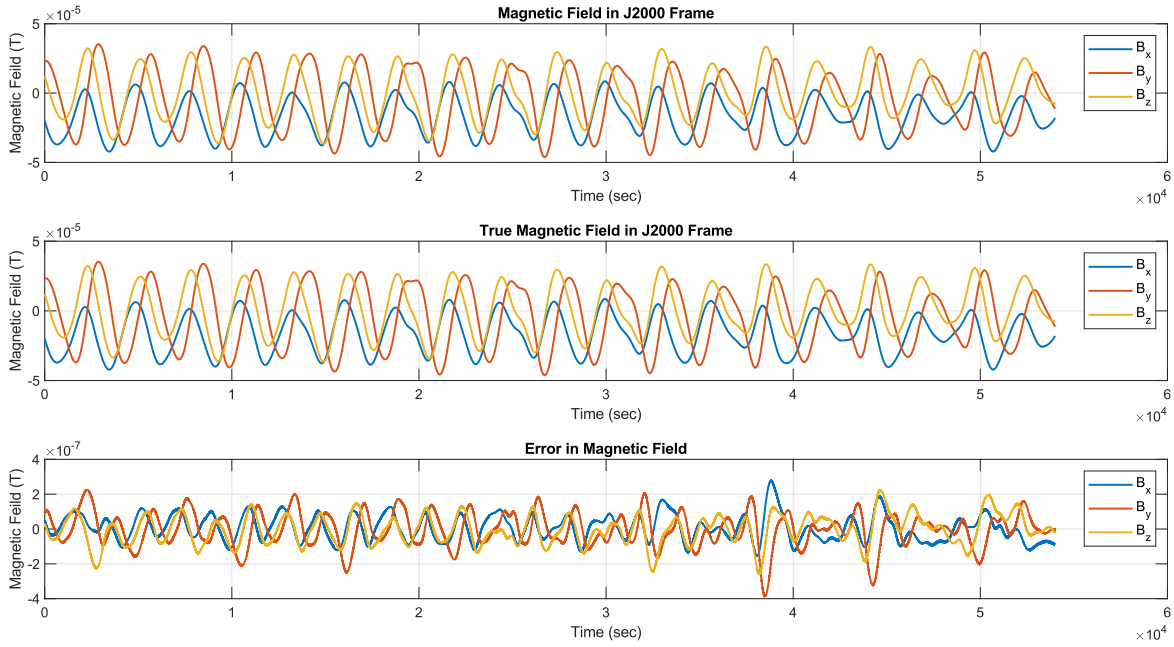


Figure 5.18: Shows the Magnetic field generated by both orbit determination methods in the J2000 frame, True Magnetic Field, and error in the measured quantity.

It was observed that the error in position using both instances was at a negligible value. The same narrative is proposed when converting the magnetic field in the J2000 frame. Therefore, the plots are approximately deemed identical for the magnetic field vectors. The magnetic field in the ECEF frame and the J2000 frame have an error of ~6.3354 x 10^{-9} T and ~8.0658 x 10^{-9} T, respectively. These error values are expected due to the issues revolving the south Atlantic anomaly, and various ionospheric effects affecting the nature of Earth's magnetic field. Therefore, showing the correct interpretation of this environmental model.

Thus, the solutions for orbit determination are compared ana analyzed to deem the better option for amateur CubeSat missions. Furthermore, the process defined to find the sun position and magnetic field vectors is also shown, which aids the ADCS for any LEO satellite missions.

CHAPTER 6 STATIC ATTITUDE DETERMINATION ALGORITHMS RESULTS AND ANALYSIS

In this chapter, three types of static attitude determination algorithms are explored and analyzed. Static attitude determination utilizes the concept of vector measurements to produce attitude estimates. These vectors are expressed in body fixed frame and an inertial reference frame. Due to the nature of the proposed solutions, quantities that are easily comparable are selected. Thus, the sun position and magnetic field vectors are utilized for this type of attitude estimation.

6.1 Attitude Estimation using Optimized TRIAD Algorithm

The optimized TRIAD method is the simplest means to determine the orientation of a spacecraft. The sensors used in this algorithm are CSS, FSS, and magnetometer. The unit vectors of the sun position and magnetic field are taken from the inertial and body frame for further processing, as the magnitude of the vector is not significant. The direction cosine matrix (DCM) is the end result and is used to represent the attitude profile is outlined in (6.1) [27].

$$Ar_i = b_i \tag{6.1}$$

where

- r_i : Vector in the inertial J2000 frame,
- b_i : is a vector in the body fixed frame and
- A: DCM representing the transformation between the inertial to the body frame.

The Optimized TRIAD algorithm forms intermediate axis constructed by the vectors in two

reference frames. A key thing to note is that this frame is restricted to dual vector measurements. This is because the TRIAD method is an orthogonal system which conducts a series of operations to produce a relevant orthogonal matrix. Prior to applying the mathematical operations required to transform this into a TRIAD solution, the optimization aspect is performed. The main purpose behind optimizing the TRIAD solution is to mitigate the noise endured in the measured sun and magnetic field vectors from the respective sensors. Thus, the optimization is conducted for both pairs of vectors as outlined in (6.2) and (6.3) for the inertial frame vectors. Similarly, (6.4) to (6.5) shows the optimized vectors measured in the body fixed frame.

$$\hat{\chi} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\hat{V}_{i1} + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\hat{V}_{i2} \tag{6.2}$$

$$\hat{\varsigma} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\hat{V}_{i1} - \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\hat{V}_{i2} \tag{6.3}$$

$$\hat{\kappa} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\hat{V}_{b1} + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\hat{V}_{b2} \tag{6.4}$$

$$\hat{\xi} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\hat{V}_{b1} - \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\hat{V}_{b2} \tag{6.5}$$

where

- $\hat{\chi}, \hat{\varsigma}, \hat{\kappa}, \hat{\xi}$: Optimized vectors using the noise variance values to apply proper weightage for sensor reliability
- σ_1, σ_2 : Noise variance values taken from the sun sensors and magnetometer datasheets respectively
- $\hat{V}_{i1}, \hat{V}_{i2}$: Represent the sun position and magnetic field vectors expressed as unit vectors in

the inertial frame

- $\widehat{V}_{b1}, \widehat{V}_{b2}$: Represent the sun position and magnetic field vectors expressed as unit vectors in the body fixed frame

Once both vector pairs are optimized, the TRIAD algorithm can now be applied. The concept is built on the orthogonal relationship between the two vectors. This process is outlined in (6.6) to (6.8) [27] [98], where the final outcome is the DCM (A).

$$\begin{aligned} r_1 &= \hat{\chi} \\ r_2 &= \frac{\left(\hat{\chi} \times \hat{\varsigma}\right)}{\left\|\hat{\chi} \times \hat{\varsigma}\right\|} \\ r_3 &= \frac{\left(\hat{\chi} \times \hat{r}_2\right)}{\left\|\hat{\chi} \times \hat{r}_2\right\|} \end{aligned} \tag{6.6}$$

$$\begin{aligned} w_1 &= \hat{\kappa} \\ w_2 &= \frac{\hat{\kappa} \times \hat{\xi}}{\left\|\hat{\kappa} \times \hat{\xi}\right\|} \\ w_3 &= \frac{\hat{\kappa} \times \hat{w}_2}{\left\|\hat{\kappa} \times \hat{w}_2\right\|} \end{aligned} \tag{6.7}$$

$$A = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}^T \tag{6.8}$$

This algorithm is modelled in Simulink where the environmental models discussed in Chapter 5 are utilized to produce vectors in the J2000 inertial frame. Figure 6.1 shows the Simulink model containing the LEO satellite simulator in conjunction with the Optimized TRIAD Algorithm and the sensor measurement models.

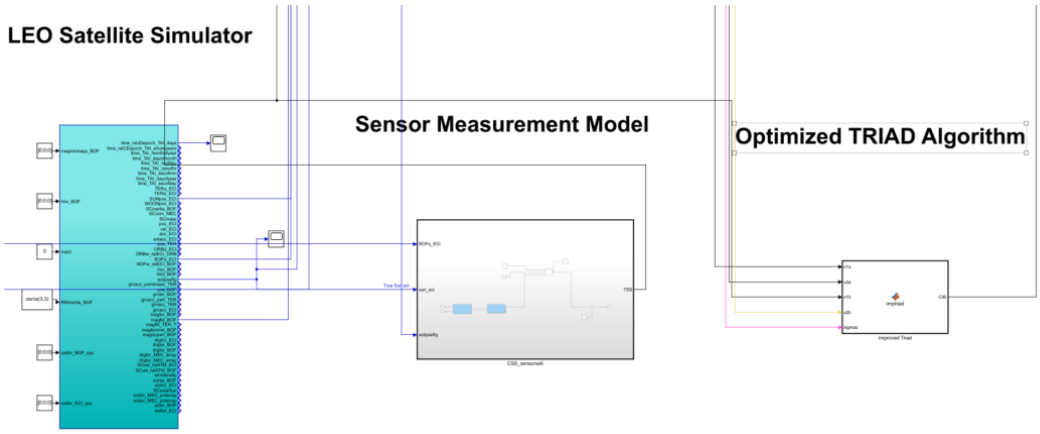


Figure 6.1: Simulink Model of Optimized TRIAD algorithm, Sensor Measurement model and the LEO Satellite Simulator.

6.1.1 Optimized TRIAD Algorithm: Simulated Results and Analysis

The simulated test required inputs defined in Table 6.1 as the values for the variances and the sensor polling rate were required to be defined. Figure 6.2 shows the plots reporting the outputs converted from the DCM to Roll, Pitch and Yaw angles, true angles, and eclipse flag. Moreover, Figure 6.3 shows the error in angles and the eclipse flag.

Table 6.1: Optimized TRIAD and Davenport's Q-Method Algorithm parameters.

Parameter	Values
Operating Frequency	10 Hz
Sun Sensor noise (σ_{sun}) FOV>60^o	2 deg
Magnetometer (σ_{mag})	7 x 10^{-7} T

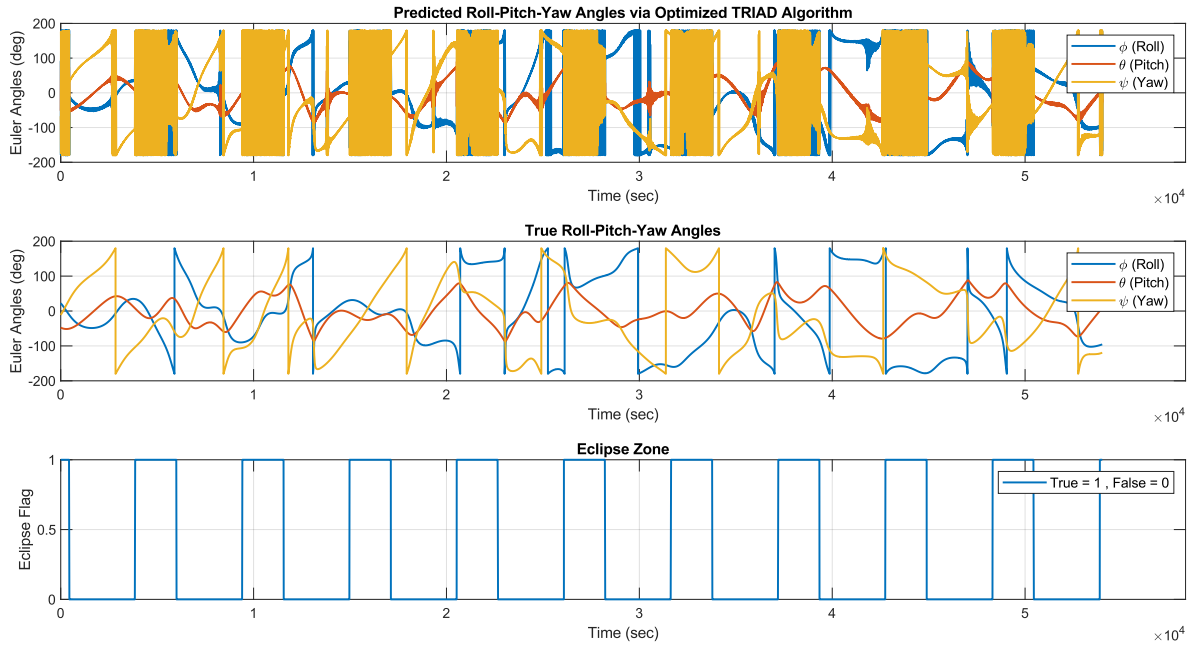


Figure 6.2: Predicted Euler angles produce via Optimized TRIAD algorithm, True Euler angles and Eclipse Flag.

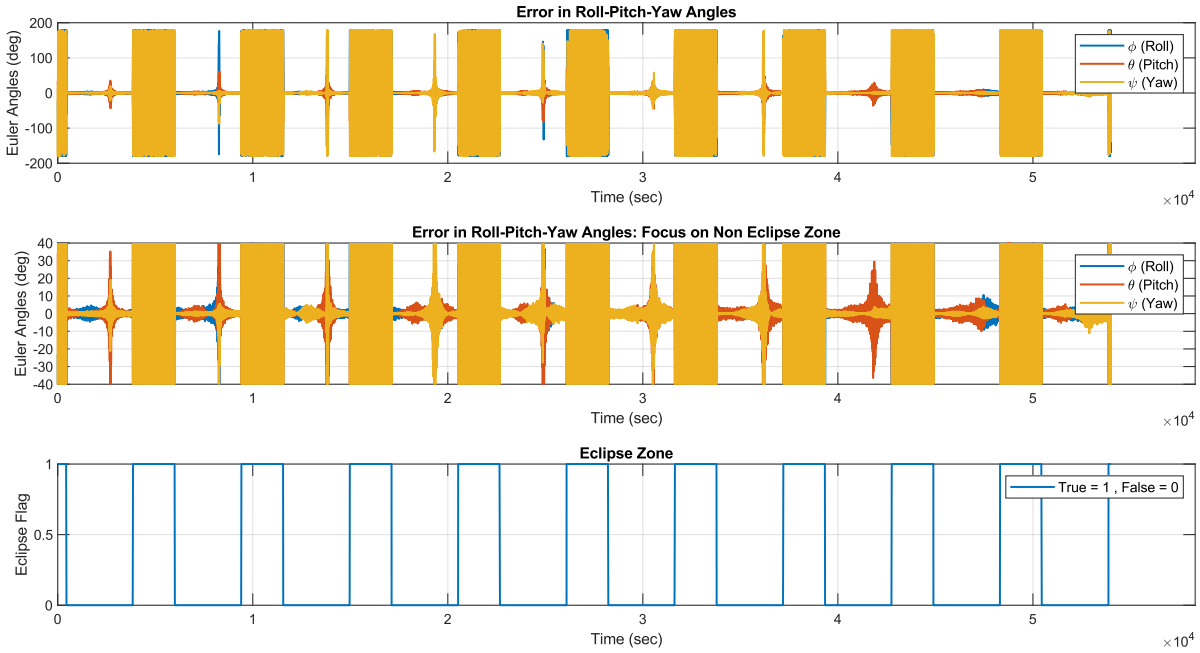


Figure 6.3: Error plots shown in terms of Euler Angles, Error in Euler Angles focusing on Non-Eclipse Zone and the eclipse flag for the Optimized TRIAD algorithm.

This experiment ran for 10 orbits to ensure the test was conducted for the majority of the 24-hour day cycle. The static estimation method demonstrates the use of vectors which was measurable for a LEO orbit. Based on the equations relevant to this algorithm, it can be hypothesized that this is heavily dependent on the sensor suite present on a satellite. The algorithm interprets the eclipse with an assumption that the satellite is in total eclipse even when it enters the penumbra region, regardless of it being a partial eclipse zone. The predicted orientation angle plots suggest a direct relation with the eclipse zones, as there are large oscillations between the angles. This is expected as the sun vector in the body frame cannot be determined if visibility of the sun is zero. Therefore, indicating a potential single point failure if this algorithm is its primary means to determine the satellite's attitude.

The error values expressed in terms of the Euler angles are extracted after the comparison of DCM with the true DCM. The error plots solidify the notion that this method is highly dependent on sensor availability. Initially, the satellite is seen to be eclipse causing a high value in error as the solution is not able to determine the orientation. For simulation timespan, the satellite enters eclipse 10 times for a duration of approximately ~35.72 minutes. During these regions it is seen the error is uncontrollable, due to the unavailability of the sun sensors. The initial convergence demonstrates it takes ~448.9 seconds for the solution to converge to a controllable attitude state. However, this is explained as the satellite is initially in eclipse. The error plot shows the RMS error in Roll, Pitch and Yaw angles are 54.6791 deg, 23.6688 deg, 59.0153 deg respectively. However, when not in eclipse, the average mean error expressed in Roll, Pitch and Yaw angles are 3.8740 deg, 3.4206 deg, 4.3621 deg. There are small impulses present during this phase as well, but this because the simulator's true quaternion output is restricted to a positive scalar value. However, this is not a testament to the algorithm's validity, because that event last for a small instance. Thus,

it can be concluded that this method is sufficient for attitude determination when not in eclipse. However, for Earth observation and surveying missions, the lack of attitude knowledge can affect the completion of mission objectives greatly.

6.2 Attitude Estimation using Davenport's Q-method Algorithm

The next static estimation technique analyzed is the Davenport's Q-method Algorithm. The optimization approach employed in the TRIAD method uses a density function to formulate an optimized solution. However, this method uses a weighted least squared approach, and attempts to produce a better attitude estimate. Moreover, it also addresses the singularity issues associated with DCMs by using quaternions. The Davenport's Q-method uses to a well-known optimization technique called a loss function. This loss function is often referred to as Wahba's problem and is shown in (6.9) [33]. The loss function is optimized by minimizing J to ensure the quaternion estimate produced meets the quaternion constraint (6.10). However, this problem is modified by Paul Davenport to transfer this into a maximization problem by converting the loss function to a gain function. This gain function is derived by expanding (6.9) to (6.11), which results in a gain function shown in (6.12) [33]. It can be seen to minimize the loss function, (6.12) is maximized, because the w_k is a constant defined by the sum of sensor variances [33].

$$\text{Min } J(A(q)) = \frac{1}{2}\sum_{k=1}^{N} w_k r_k^T r_k$$
$$\text{Where} \tag{6.9}$$
$$r_k = \hat{s}_{b,k} - A(q)\hat{s}_{i,k}$$

$$A(q)^T A(q) = q^T q = 1 \tag{6.10}$$

$$J(A(q)) = \left(\sum_{k=1}^{N} w_k \right) - \left(\sum_{k=1}^{N} w_k \hat{s}_{b,k}^{T} A(q) \hat{s}_{i,k} \right) \tag{6.11}$$

$$G(A(q)) = \sum_{k=1}^{N} w_k \hat{s}_{b,k}^{T} A(q) \hat{s}_{i,k} \tag{6.12}$$

where

- J (A(q)): Loss function defined in the Wahba's problem
- G (A(q)): Gain function defined as the Wahba's problem
- N: Number of unit vectors measured in the body fixed frame
- $\hat{s}_{b,k}$: Measured unit vector in the body fixed frame
- $\hat{s}_{i,k}$: Unit vector expressed in the inertial J2000 frame
- w_k : Weight for the vectors defined in the body fixed frame are defined by $1/\sigma^2$
- A(q): Transformation matrix to relate a vector from the inertial frame to the body fixed frame

The optimal attitude estimate used to maximize the gain function is found by solving for the eigenvalue and eigenvector of a symmetric Davenport's matrix. This is analogous to finding a minima and maxima of a standard quadratic equation. The solution is initiated by using the trace operation on (6.12) to produce the gain function in terms of a quaternion as outlined in (6.13) [33] [98]. In order to do so, the transformation matrix A(q) is expressed in terms of quaternion multiplication by expanding the terms as outlined in (6.14) [98]. Furthermore, the attitude profile and the Davenport's matrices expressed in the gain function are defined in (6.15) [98].

$$G(q) = q^T K(B) q \tag{6.13}$$

$$A(q) = \left(q_0^2 - \vec{q}^T\vec{q}\right) I_3 + 2\vec{q}\vec{q}^T - 2q_0\vec{q}^{\times} \tag{6.14}$$

$$B = \sum_{k=1}^{N} w_k \hat{s}_{b,k} \hat{s}_{i,k}^T$$

$$K(B) = \begin{bmatrix} B + B^T - I_3 tr\left(B\right) & z \\ z^T & tr\left(B\right) \end{bmatrix} \tag{6.15}$$

$$z = \begin{bmatrix} B_{23} - B_{32} \\ B_{31} - B_{13} \\ B_{12} - B_{21} \end{bmatrix}$$

where

- G(q): Gain function derived from the Wahba's problem expressed as a quaternion multiple
- K(B): Symmetric Traceless Davenport's matrix
- B: Attitude profile matrix
- q: Quaternion relating a vector in the inertial frame to the body fixed frame

Finally, the concept of a Lagrange multiplier is used along with the constraint defined in (6.10) to form a modified gain function (6.16) [33]. To express this in the standard eigenvector and eigenvalue form as shown in (6.17), (6.16) is differentiated in terms of the quaternion and the Lagrange multiplier. This results in the final form (6.18), where the maximum eigenvalue is taken as the final solution. This is simply because it maximizes the gain functions leading to produce an optimal quaternion estimate.

$$G_2\left(q, \lambda\right) = q^T K\left(B\right) q + \lambda\left(1 - q^T q\right) \tag{6.16}$$

$$A\vec{u} = \lambda\vec{u} \tag{6.17}$$

$$K(B)q = \lambda q \tag{6.18}$$

where,

- A: Symmetric matrix, interpreted as Davenport's matrix K(B)
- λ: Eigenvalue of the Davenport's matrix, K(B)
- $\vec{u}, q$: Eigenvectors of the Davenport's matrix, K(B)

The Davenport's Q-method in Simulink shown in Figure 6.3 which incorporates this static attitude determination algorithm defined in (6.9) to (6.18). The sensors used are kept consistent with the optimized TRIAD algorithm as these are present on the ESSENCE CubeSat.

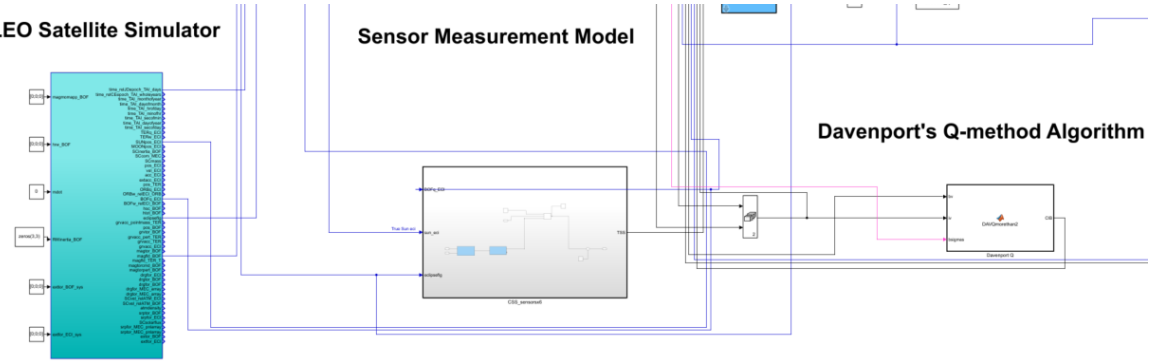


Figure 6.4: Simulink Model of Davenport's Q-method, Sensor Measurement model and LEO Satellite Simulator.

6.2.1 Davenport's Q-method Algorithm: Simulated Results and Analysis

The parameters required to initialize this algorithm are defined in Table 6.1. This attitude estimation algorithm places more faith on the magnetometer, as this sensor is not affected by an eclipse event. The output of this algorithm is in terms of a quaternion. Therefore, a comparison is made in terms of quaternions and Euler angles. Figures 6.4 shows the predicted quaternion, true

quaternion, and eclipse flag. Figure 6.5 demonstrates the error quaternion, error in Euler angles and eclipse flag.

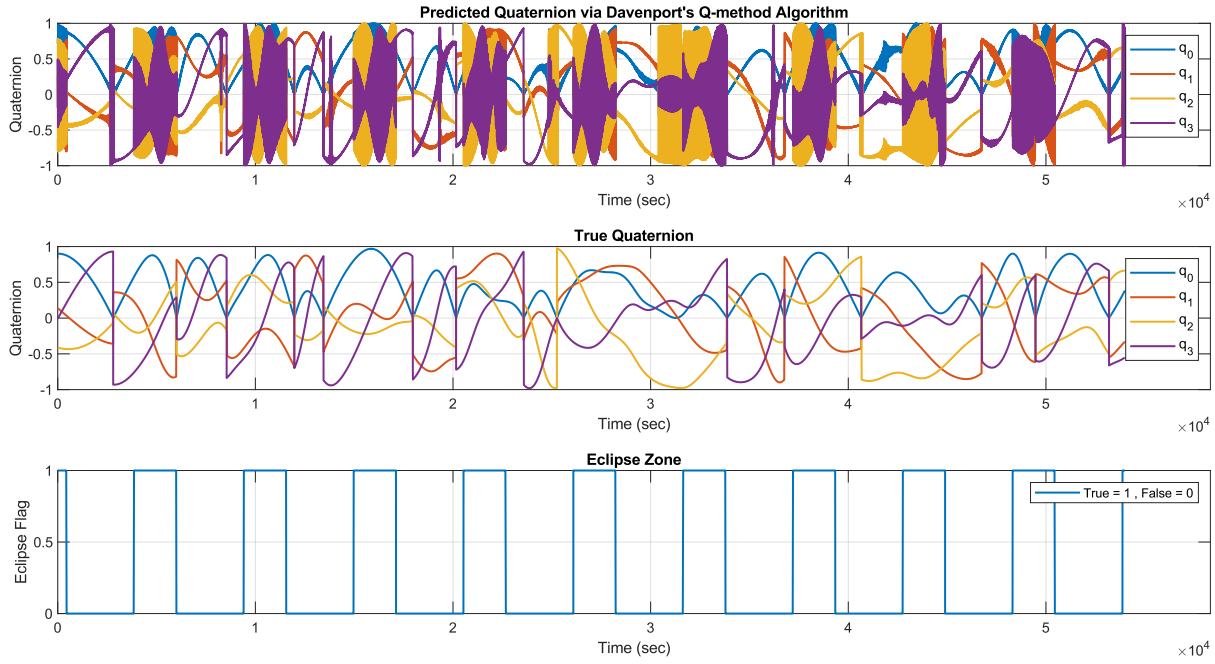


Figure 6.5: Shows the predicted quaternion generated via Davenport's Q-method Algorithm, true quaternion, and the eclipse flag.

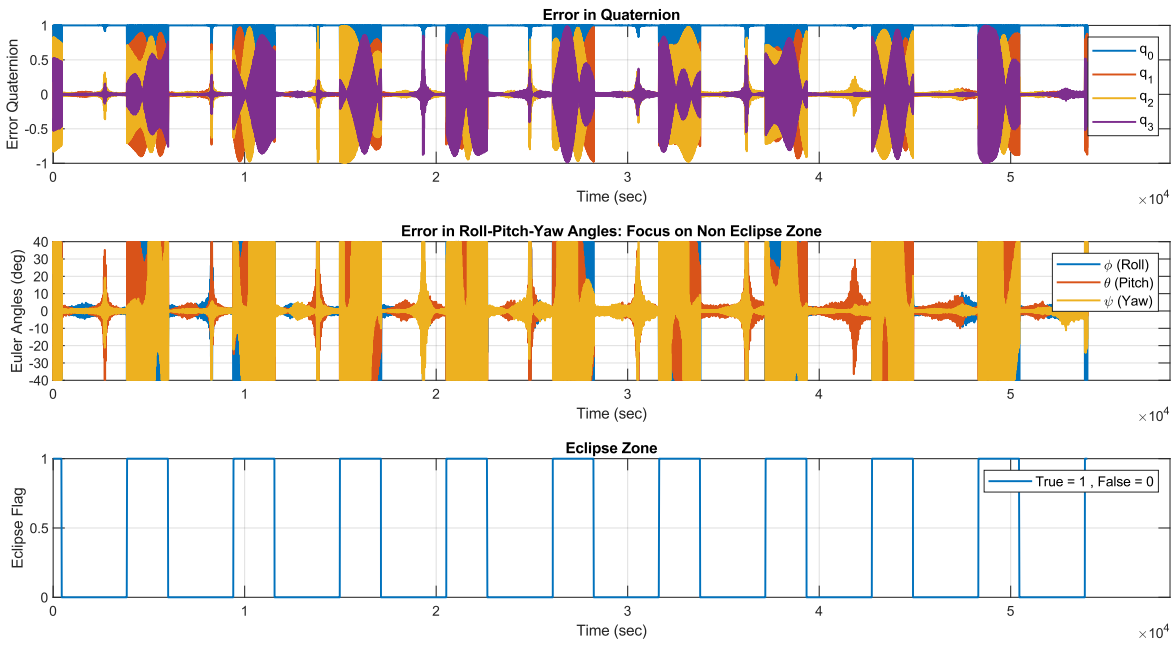


Figure 6.6: Error plots shown in terms of error quaternion, error in Euler angles focusing on the non-eclipse zones and the eclipse flag for the Davenport's Q-method Algorithm.

The simulation ran for 10-orbits seeking to improve and stabilize the error seen in the optimized TRIAD method. Moreover, the operating frequency was set to 10 Hz as defined by **RQR-SYS-CDH-161** for the ESSENCE mission. For the Davenport's Q-method, the use of Wahba's problem produces an optimal solution in terms of an attitude quaternion. This unique approach attempts to stabilize the uncertainties relevant to the sensors used, leading to a converged result across all three axes. Figure 6.5 produces the same narrative as the optimized TRIAD algorithm, that the stability in the solution diminishes for ~35.72 minutes per orbit. The improvement in the convergence time is ~1 mins, as the error density function used for the previous method is observed to be marginally worse. The error values expressed in the Euler angles are extracted from the formulation of the error quaternion when compared to the true value. This was also done with the QUEST algorithm that will be discussed in the next section.

Like the TRIAD method, Figure 6.6 shows that the error is not controllable, but does show minor improvements. The RMS error found throughout the simulation lifetime in terms of Roll, Pitch and Yaw are 49.8746 deg, 21.7432 deg, 50.2953 deg. When compared to the Optimized TRIAD Algorithm, this shows an improvement in all three Euler angles especially in the Yaw angle. During the non-eclipse periods, the error in Roll, Pitch and Yaw are found to be 3.8749 deg, 3.4215 deg, 4.3630 deg. There is no significant difference as these static approaches are expected to behave the same. Thus, an improvement in the attitude estimation is achieved using the Davenport's Q-method algorithm. These error values can be explained by the drawback of the loss function when in eclipse as the magnetometer is the only source of attitude estimate. This would entail when the sun sensors are not available the algorithm tends to correct for incorrect CSS and FSS outputs.

6.3 Attitude Estimation using QUEST Algorithm

As seen from the previous two methods of static attitude estimators, the error growth tends to settle down once the eclipse period has lapsed. The QUEST algorithm is meant to develop an efficient and optimal quaternion solution. Due to computational constraints present on satellites, the need for this static estimation technique is evident. The output of this method is the same as the Davenport's Q-method, but an efficient approach is taken while solving for it. The process begins by further manipulating the eigenvalue and eigenvector defined in the Davenport's Q-method. The Davenport's matrix K(B) is manipulated by first expanding it in terms of the vector and scalar components of the quaternion, shown from (6.19) to (6.22) [33].

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} \vec{q} \\ q_0 \end{bmatrix} = \lambda \begin{bmatrix} \vec{q} \\ q_0 \end{bmatrix} \tag{6.19}$$

$$\begin{bmatrix} \left(B+B^T\right)-I_3K_{22} & K_{12} \\ K_{12}^T & K_{22} \end{bmatrix}\begin{bmatrix} \vec{q} \\ q_0 \end{bmatrix}=\lambda\begin{bmatrix} \vec{q} \\ q_0 \end{bmatrix} \tag{6.20}$$

$$\begin{aligned} &\vec{q}\left[S-I_3K_{22}\right]+K_{12}q_0=\lambda\vec{q} \\ &S=B+B^T \end{aligned} \tag{6.21}$$

$$\vec{q}K_{12}^T+K_{22}q_0=\lambda q_0 \tag{6.22}$$

The next step is to use the concept of Rodrigues Parameter Representation and convert the unit quaternion in terms of the **Gibbs Vector (g),** shown in (6.23) . Thus, (6.21) is configured to produce (6.24), which is grouped in terms of "**g"**. The Gibbs vector can be transformed as a quaternion as shown in (6.25). The operation takes place by dividing both sides by the scalar component of the quaternion. This allows the derived quantity to be directly influenced by the measured sun position and magnetic field vectors. To produce an optimized solution for the attitude problem, a function in terms of eigenvalue is developed by modifying the coefficient of the Gibb's vector in (6.24). Therefore, the QUEST algorithm quantity defined in (6.26) is obtained [33]. Finally, substituting this back into (6.24) produces the Gibbs vector which is used to find the quaternion solution using the QUEST algorithm.

$$g=\frac{\vec{q}}{q_0} \tag{6.23}$$

$$\left[I_3\left(\lambda+K_{22}\right)-S\right]g=K_{12} \tag{6.24}$$

$$q=\frac{1}{\sqrt{1+\|g\|^2}}\begin{bmatrix} 1 \\ g \end{bmatrix} \tag{6.25}$$

$$\left[I_3\left(\lambda + K_{22}\right) - S\right]^{-1} = \frac{1}{\gamma}\left(\alpha I_3 + \beta S + S^2\right)$$

$$\begin{aligned} \alpha &= \lambda^2 - K_{22}^2 + tr\left[adj\left(S\right)\right] \\ \beta &= \lambda - K_{22} \\ \gamma &= \left(\lambda + K_{22}\right)\alpha - \det\left(S\right) \end{aligned} \tag{6.26}$$

To find the correct eigenvalue (6.22) is utilized and is further represented in terms of the Gibb's vector shown in (6.27). This allows the algorithm to equate the two equations by solving for the eigenvalue to used to in (6.26). Therefore, using (6.23) to (6.27) and using multiple algebraic identities shown in [33] **the eigen value function ($f\left(\lambda\right)$)** (6.28) is formed. To find the roots of this function, it is solved using a recursive numerical method called the Newton-Raphson technique. This is illustrated in (6.29) where it is initialized by setting the first eigenvalue to be the inverse sum of sensor variances [33]. A **threshold value (ε_{thr})** is defined for the estimator to satisfy. Thus, leading to conclude the solved eigenvalue is exclusive and is solved efficiently. This eigenvalue is then substituted into (6.26) to form Gibb's vector, resulting in the desired attitude quaternion, using (6.25). Furthermore, the loss function is also minimized without using an analytical approach defined in the Davenport's Q-Method Algorithm. The QUEST algorithm is assembled in Simulink as shown in Figure 6.7.

$$K_{12}^T g = \lambda - K_{22} \tag{6.27}$$

$$f\left(\lambda\right) = \lambda^4 - \left(a + b\right)\lambda^2 - c\lambda + \left(ab + cK_{22} - d\right)$$

$$\begin{aligned} a &= K_{22}^2 - tr\left[adj\left(S\right)\right] \\ b &= K_{22}^2 + K_{12}^T K_{12} \\ c &= \det\left(S\right) + K_{12}^T S K_{12} \\ d &= K_{12}^T S^2 K_{12} \end{aligned} \tag{6.28}$$

$$\lambda_{i+1} = \lambda_i - \frac{f(\lambda_i)}{f^{'}(\lambda_i)}, \text{ until } \left|\lambda_{i+1} - \lambda_i < \varepsilon_{thr}\right|$$

$$\lambda_1 = \sum_{i=1}^{N} w_i \tag{6.29}$$

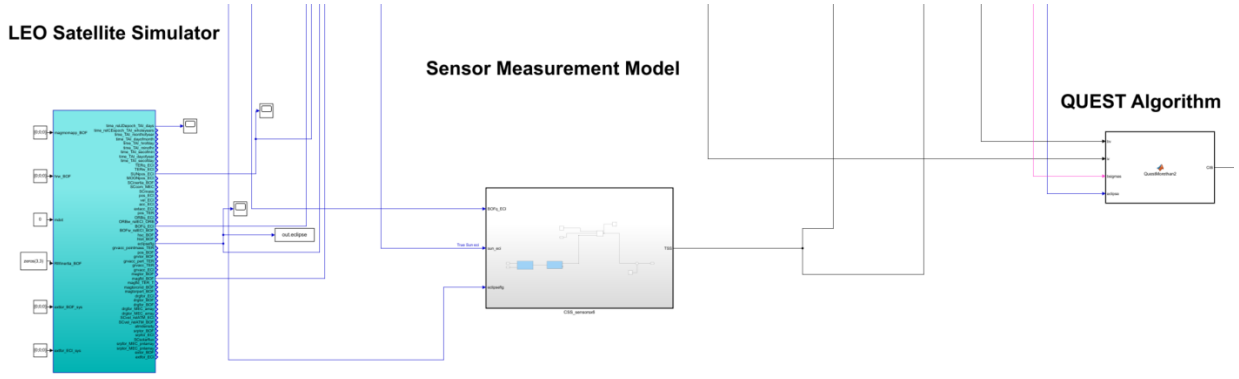


Figure 6.7: Simulink Model of QUEST Algorithm, Sensor Measurement model and the LEO Satellite Simulator.

6.3.1 Quest Algorithm: Simulated Results and Analysis

It is required to initialize the QUEST algorithm by defining a few preliminary parameters, as shown in Table 6.2. Figures 6.8 shows plots consisting of the predicted quaternion, true quaternion, and eclipse flag. Figure 6.9 demonstrates error quaternion, Euler angles and eclipse flag plots.

Table 6.2: QUEST Algorithm parameters.

Parameter	Values
Operating Frequency	10 Hz
Sun Sensor noise (σ_{sun}) FOV>60^o	2 deg
Magnetometer (σ_{mag})	7 x 10^{-7} T
QUEST Threshold (ε_{thr})	1 x 10^{-8}

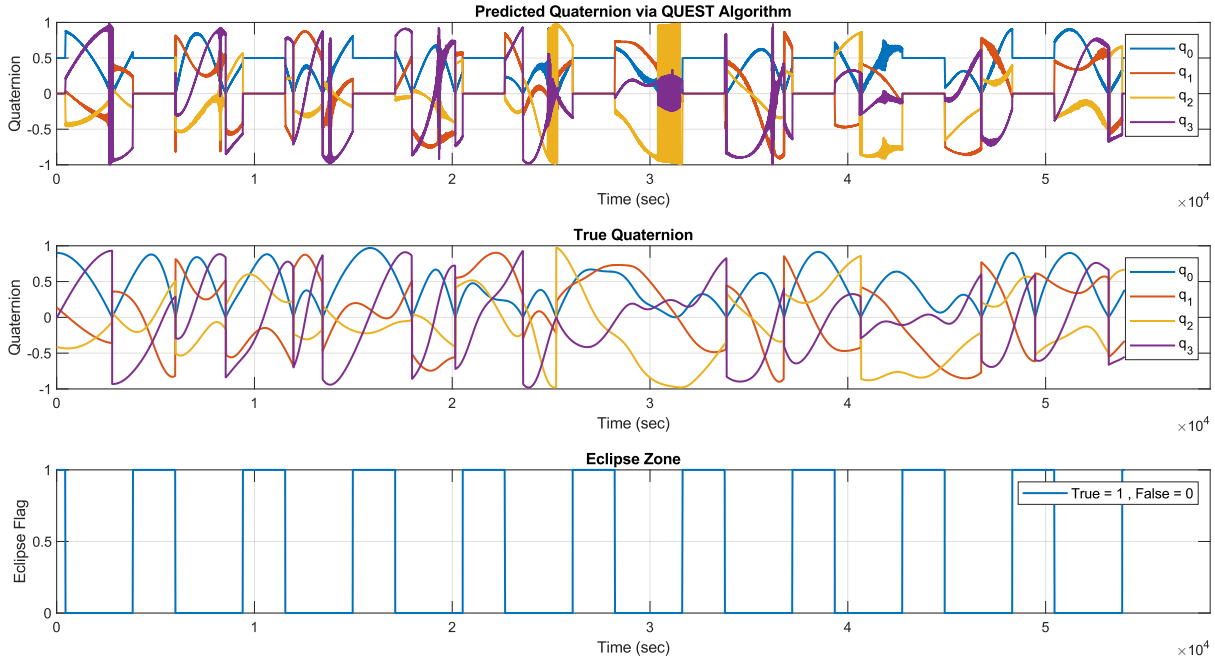


Figure 6.8: Shows the predicted quaternion generated from the QUEST Algorithm, true quaternion, and the eclipse flag.

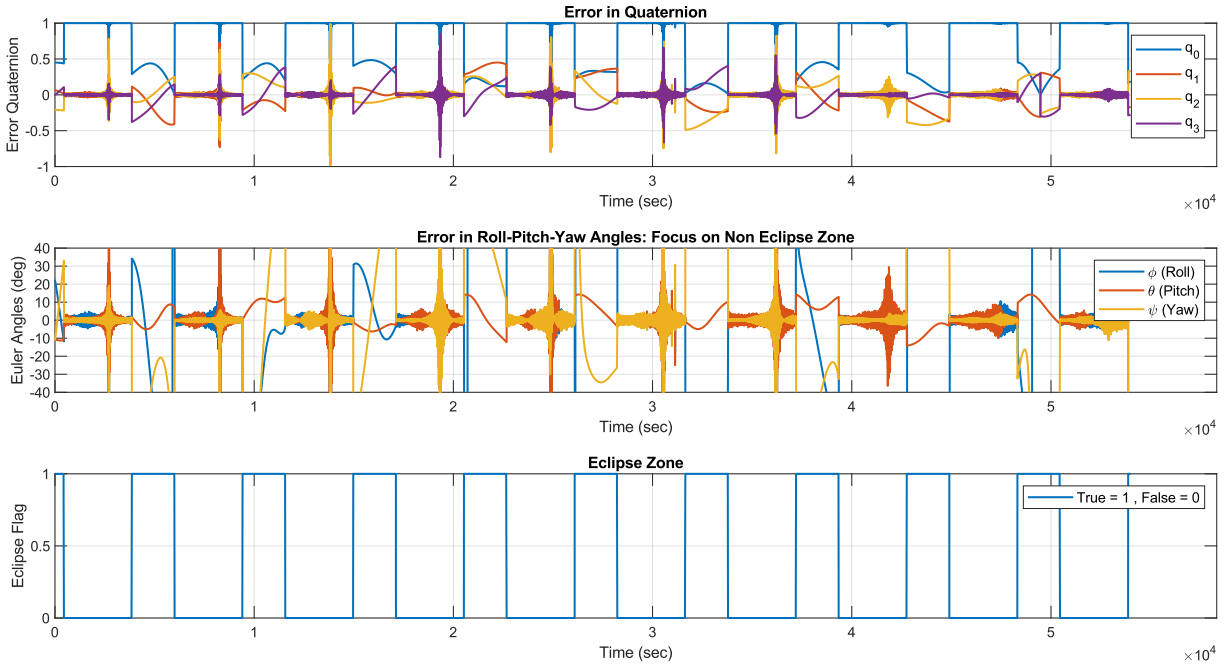


Figure 6.9: Error plots shown in terms of error quaternion, error in Euler angles focusing on the non-eclipse zones and the eclipse flag for the QUEST algorithm.

The recursive algorithm performs efficiently by using the threshold defined in Table 6.2. This benefits CubeSats because the onboard computational power is conserved. The tolerance value is a variable quantity which is altered based on the clock speed of the OBC and the sensor polling rate. It can be seen from Figure 6.8, the quaternion solution generated from QUEST deteriorates while in the eclipse phase. However, the error remains stable as it does not grow in a rapid fashion when compared to the previous two static estimators. This is better shown in Figure 6.8 as it shows the slow growth rate. Thus, this algorithm provides a great benefit in this regard. For the 10-orbit simulation timespan, the RMS error in Roll, Pitch and Yaw values are 61.3359 deg, 5.6481 deg, 48.0255 deg, respectively. These values have shown an improvement when compared to the Optimized TRIAD Algorithm but not the Davenport's Q-Method. This is mainly because of the approach taken to solve the attitude determination problem. The Davenport's Q-method produces a quaternion when the eigenvalue optimization loop reaches a 0 value to produce a minima value. However, the QUEST methods seeks to converge to a near 0 value defined via threshold. This results in a slight error increase when compared to the Davenport's Q-method, as shown in Table 6.3. The Roll angle improves but the Pitch and Yaw are worse.

Table 6.3: RMS Error expressed in Roll, Pitch and Yaw for all static attitude algorithms.

Static Attitude Estimation Algorithm	**Error in Roll (ϕ)**	**Error in Pitch (θ)**	**Error in Yaw (ψ)**
Optimized TRIAD	54.6791 deg	23.6688 deg	59.0153 deg
Davenport's Q-Method	49.8746 deg	21.7432 deg	50.2953 deg
QUEST	61.3359 deg	5.6481 deg	48.0255 deg

In conclusion, it is evident all three static attitude estimation methods, produce similar error values as per Table 6.3. A key benefit in using these types of attitude estimation is that they are easy to implement. For amateur CubeSat missions, the error tolerance is typically not strict when compared to large budget Earth Observation missions. For all three static estimators the error value rise when a satellite enters an eclipse. This is demonstrated in Figures 6.3, 6.6 and 6.9 as the error oscillates at an unmanageable rate. The Optimized TRIAD and the Davenport's Q-Method algorithms are prime examples of such a behaviour. However, the QUEST algorithm's error doesn't grow as rapidly. This is due to the nature of the solver as it assumes, the previous state is the next state once the satellite enters an eclipse. The QUEST approach is the closest of the three to online estimation techniques. Moreover, the setback between all three methods is that they are very sensor dependent. This leads to possibilities of single point failures and can result in a complete ADCS failure for a satellite. This is why static estimation should never be a primary attitude determination algorithm option. A gyroscope isn't typically incorporated into this form of attitude estimation because of the well-known gyroscope bias/drift effect. For the ESSENCE mission the QUEST Algorithm is defined to be the best approach among the three as a secondary estimator. This is because the error in eclipse is grows at a significantly slower rate in QUEST, and the use of a threshold value reduces the computational and power needs of the CubeSat's ADCS.

CHAPTER 7 ONLINE ATTITUDE DETERMINATION ALGORITHMS ANALYSIS AND RESULTS

In this chapter, online attitude determination algorithm is discussed and analyzed. The algorithm explored is a variation a standard Kalman filter, called the Multiplicative Extended Kalman Filter (MEKF). There are two system models explored which explore a traditional attitude estimation system model and a sensor reliant one. Moreover, the asynchronous sensor fusion strategies and a system analysis of both system models are examined as well.

7.1 Attitude Estimation using MEKF Algorithm: Dynamics and Kinematics

The development of a real time attitude determination algorithm an MEKF is established. This method is similar to a standard linear Kalman filter but is consisted of a few differences. The steps to build this algorithm are defined in Table 7.1. This is a recursive algorithm and loops through each of these steps at a particular timestep (Δt).

Table 7.1: MEKF steps used for online attitude determination.

MEKF Steps	Step Contents
Initialization	$\hat{x}_0^- = E\left[x(t_0)\right]$ $P_0^- = E\left[\left[x(t_0)-\hat{x}_j\right]\left[x(t_0)-\hat{x}_j\right]^T\right]$
Propagation	$\dot{x} = f(x,t)+G(t)w(t)$ $\dot{P} = F(t)P(t)+P(t)F(t)^T+G(t)Q(t)G(t)^T$ Where, $x = \hat{x}_j^+$
Measurement Update/Reset	$K_j = P_j^- H_j^T\left[H_j P_j^- H_j^T + R_j\right]^{-1}$ $\hat{x}_j^+ = \hat{x}_j^- + K_j\left[z_j - h\left(\hat{x}_j^-\right)\right]$ $P_j^+ = \left[I_9 - K_j H_j\right]P_j^-$

where

- (-) : Symbolizes the current quantity and indicates it is a priori state/covariance
- (+) : Symbolizes the current quantity and indicates it is a posteriori state/covariance
- (^): Symbolizes predicted state value
- E[■]: Expected value set for states and covariance

Initialization

The initialization step is the first step in the MEKF's recursive process. This step consists of defining the states relevant for the problem. Because the focus of this thesis is attitude determination, quantities such as quaternion, angular velocity and bias are defined as states. The

state values are shown in (7.1). Due to the unintuitive nature of quaternions, the state is internally reduced to make it in terms of an error quaternion for the state and covariance propagation, and update steps. This is shown in (7.2), where only the vector component of the attitude error quaternion, angular rates and bias are considered [47]. This is because the information lies only in the vector component of the quaternion, the error quaternion, and is referred to as an internal state vector. This vector is later handled by the update step but is derived form the global state vector (7.1). The covariance matrix uses this internal state vector as well to produce a relationship in terms of the error produced by the sensors used in the system. This is shown in (7.4) where "P" is the covariance matrix [47]. To start the process, *x*, and *P* are initialized by setting them as the expected mean values of the state vector and the covariance matrix. However, they are also interpreted as arbitrary values which are preconfigured to be bad estimates for the system.

$$x = \begin{bmatrix} q & \omega & \beta \end{bmatrix} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 & \omega_x & \omega_y & \omega_z & \beta_x & \beta_y & \beta_z \end{bmatrix} \tag{7.1}$$

$$\delta x = \begin{bmatrix} a & \Delta\omega & \Delta\beta \end{bmatrix} \tag{7.2}$$

$$\delta q(a) = \begin{bmatrix} 1 \\ a \end{bmatrix} \tag{7.3}$$

where

- x: global state vector used in the MEKF
- δx: internal state vector used for error correction
- P: Covariance matrix used in the MEKF
- q: Unitless 4 x 1 dimensional quantity referred to as quaternion
- ω: 3 x 1 angular velocity measured in rads/sec

- β: 3 x 1 bias acting on the system measured in rads/sec
- a : Vector component in the error quaternion and is similar to $\bar{q}$
- $\Delta\omega, \Delta\beta$: Error in the angular velocity and bias in the system

$$P = \begin{bmatrix} E\left[aa^T\right] & E\left[a\Delta\omega^T\right] & E\left[a\Delta\beta^T\right] \\ E\left[\Delta\omega a^T\right] & E\left[\Delta\omega\Delta\omega^T\right] & E\left[\Delta\omega\Delta\beta^T\right] \\ E\left[\Delta\beta\Delta a^T\right] & E\left[\Delta\beta\Delta\omega^T\right] & E\left[\Delta\beta\Delta\beta^T\right] \end{bmatrix} \tag{7.4}$$

Propagation

The next step in this iterative estimation process is to conduct state and covariance propagation. The global state vector is still used for state propagation, rather than the internal one, because this isn't an error centric step. The system model used for propagation consists of the Euler's dynamics and quaternion kinematics equations. The bias state is to factor in error from the MEMS and FOG gyroscopes as time progresses, so a standard system noise is used for propagation. In the propagation step shown in Table 7.1, it is outlined in terms of $\dot{x}$ which indicates it's a function of time. Thus, the system model used to propagate the current state to the next timestep is defined as shown in (7.5).

$$\dot{x} = \begin{bmatrix} \dot{q}_{BI} \\ \dot{\omega}_B^{BI} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\Xi\left(q_{BI}\right)\omega_B^{BI} \\ J_B^{-1}[\tau_B - \omega_B^{BI} \times (J_B\omega_B^{BI})] \\ \eta_u \end{bmatrix} \tag{7.5}$$

The final outcome of this step is to generate a priori state ($\hat{x}_{j+1}^-$) and covariance (P_{j+1}^-) to produce

from the current state ($\hat{x}_j^+$) and covariance (P_j^+). The extracted quantities defined as the state are substituted in (7.5) to produce $\dot{x}$. Similarly, the propagation of covariance requires the calculation of $\dot{P}$, and this is done by utilizing (7.6).

$$\dot{P} = F(t)P(t) + P(t)F(t)^T + G(t)Q(t)G(t)^T \tag{7.6}$$

Where F, G and Q are the state transition, disturbance torque error and process noise matrices respectively.

Moreover, they are also referred to as Jacobian matrices because, they are computed by taking the partial derivatives with respect to the internal state vector. Thus, F and G are outlined in (7.7) and (7.8) are used to propagate the covariance matrix [47]. However, the process noise matrix represents the unmodelled noise quantities such as disturbance torques. These are incorporated in the Q matrix and is outlined in (7.9).

$$F = \frac{\partial \dot{x}}{\partial x} = \begin{bmatrix} -\omega^{\times} & \frac{1}{2}I_3 & 0_3 \\ 0_3 & J_B^{-1}\left[-\left[\omega_B^{BI\times}\right]J + \left[J_B\omega_B^{BI}\right]^{\times}\right] & 0_3 \\ 0_3 & 0_3 & 0_3 \end{bmatrix} \tag{7.7}$$

$$G = \frac{\partial \dot{x}}{\partial \eta} = \begin{bmatrix} 0_3 & 0_3 \\ J_B^{-1} & 0_3 \\ 0_3 & 0_3 \end{bmatrix} \tag{7.8}$$

$$Q = \begin{bmatrix} \sigma_d^2 I_3 & 0_3 \\ 0_3 & \sigma_u^2 I_3 \end{bmatrix} \tag{7.9}$$

where η is the noise in the system which includes the disturbance torques (τ_B) mentioned in

Chapter 3, $\sigma_d \approx 5\text{x}10^{-8}$ is an accumulation of disturbance torques acting on the system and $\sigma_u \approx 1.16\text{x}10^{-5}$ is bias error resulting from gyroscope.

Upon the completion of this calculation, 4th order Runge-Kutta numerical integration method is employed. The method is outlined in (7.10) and (7.11). Where Δt is the timestep. For state propagation function g(x) is expressed and equated to (7.5), but for covariance propagation g(x) uses (7.6).

$$\hat{x}_{j+1}^{-} = \hat{x}_{j}^{-} + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \tag{7.10}$$

$$\begin{aligned} k_1 &= \Delta t \bullet g\left(x\right) \\ k_2 &= \Delta t \bullet g\left(x + \frac{1}{2}k_1\right) \\ k_3 &= \Delta t \bullet g\left(x + \frac{1}{2}k_2\right) \\ k_4 &= \Delta t \bullet g\left(x + k_3\right) \end{aligned} \tag{7.11}$$

Measurement Update

The measurement update step follows the process described in Table 7.1 but requires the derivation of the observation matrix (H). This is exclusive to each sensor and is derived from the observation function, h(x). This function uses the sensor's mathematical model to convert the states in terms of the measured quantities. For the Fibre Optic and MEMS gyroscopes, the outputting quantity is angular velocity in BFF. The benefit in using this sensor is that no modification is required for to the state vector. The function for these types of sensors are outlined by (7.12), which leads to the formulation of the observation matrix (7.13). It can be seen the function is similar to the measurement model of a gyroscope but doesn't represent the measured quantities. The gyroscopes correct the angular velocity and bias states using the state values defined in the system. The

observation matrix (H) is formed by taking the partial derivatives in terms of the internal state vector, because the covariance is required to calculate the Kalman Gain. In addition, the sensor noise matrix takes the RMS error value to form the error covariance matrix for the gyroscopes. These matrices are interpreted using an integrated ADC which has a 16-bit word count. Using the full scale, least significant bit and word length values from Table 4.1 and 4.2, the R matrices for the MEMS and Fibre Optic Gyroscopes are formed in (7.14) and (7.15).

$$h_{gyro}(q_{BI}, \omega_B^{BI}, \beta) = \omega_B^{BI} + \beta + \eta_v \tag{7.12}$$

$$H_{gyro} = \begin{bmatrix} \frac{\partial h}{\partial a} & \frac{\partial h}{\partial \omega} & \frac{\partial h}{\partial \beta} \end{bmatrix} = \begin{bmatrix} 0_3 & I_3 & I_3 \end{bmatrix} \tag{7.13}$$

$$R_{MEMS_Gyro} = \begin{bmatrix} \left(\frac{\pi}{180\left(2^{16-LSB}/FS\right)}\right)^2 & 0 & 0 \\ 0 & \left(\frac{\pi}{180\left(2^{16-LSB}/FS\right)}\right)^2 & 0 \\ 0 & 0 & \left(\frac{\pi}{180\left(2^{16-LSB}/FS\right)}\right)^2 \end{bmatrix} \tag{7.14}$$

FS=450°/s,LSB=1

$$R_{FOG_Gyro} = \begin{bmatrix} \left(\frac{\pi}{180\left(2^{16-LSB}/FS\right)}\right)^2 & 0 & 0 \\ 0 & \left(\frac{\pi}{180\left(2^{16-LSB}/FS\right)}\right)^2 & 0 \\ 0 & 0 & \left(\frac{\pi}{180\left(2^{16-LSB}/FS\right)}\right)^2 \end{bmatrix} \tag{7.15}$$

FS=585.15°/s,LSB=1

In addition to the gyroscopes, the two types of sun sensors used are CSS and FSS. Even though both of these sensors observe the sun's illumination, they don't report the same quantity as their outputs. The CSS produces and angular output which is accumulated to generate a sun vector.

However, the FSS provides a sun position vector by using the azimuth and elevation measurements and is later converted to a cartesian quantity. These sensors utilize the sun's position model and the quaternion state to form the observation function in BFF. The observation functions for the CSS and FSS is outlined by (7.16) and (7.17) respectively. Using the logic described in (7.13)the observation matrices for CSS and FSS are shown in (7.18) and (7.19) [48]. Lastly, the sensor covariance matrices are formed using the error values shown in Tables 4.3 and 4.4 and are outline in (7.20) and (7.21).

$$h_{CSS}(q_{BI}, \omega_B^{BI}, \beta) = \cos^{-1}\left(\hat{n}_B \bullet \left[q_{BI}\hat{s}_I q_{BI}^{-1}\right]\right) + \eta_v \tag{7.16}$$

$$h_{CSS}(q_{BI}, \omega_B^{BI}, \beta) = q_{BI}\hat{s}_I q_{BI}^{-1} + \eta_v \tag{7.17}$$

$$H_{CSS} = \begin{bmatrix} 2\hat{n}_B s_B^{\times} & 0_3 & 0_3 \end{bmatrix} \tag{7.18}$$

$$H_{FSS} = \begin{bmatrix} 2s_B^{\times} & 0_3 & 0_3 \end{bmatrix} \tag{7.19}$$

$$R_{CSS} = \begin{bmatrix} \left(\frac{1.85\pi}{180}\right)^2 & 0 & 0 \\ 0 & \left(\frac{1.85\pi}{180}\right)^2 & 0 \\ 0 & 0 & \left(\frac{1.85\pi}{180}\right)^2 \end{bmatrix} \tag{7.20}$$

$$R_{FSS} = \begin{bmatrix} (0.9998)^2 & 0 & 0 \\ 0 & \left(1.74\times10^{-2}\right)^2 & 0 \\ 0 & 0 & \left(3\times10^{-4}\right)^2 \end{bmatrix} \tag{7.21}$$

The final sensor used is the three-axis magnetometer. Similar to the sun sensors, a reference vector in the inertial J200 frame is required. The magnetometer uses the IGRF-13 reference model and

the quaternion state to form the observation function in BFF. The observation function and matrix is determined and shown in (7.22) and (7.23). Moreover, the sensor covariance matrix is outlined in (7.24), which takes the error values defined in Table 4.5.

$$h_{Mag}(q_{BI}, \omega_B^{BI}, \beta) = q_{BI} B_I q_{BI}^{-1} + \eta_v \tag{7.22}$$

$$H_{Mag} = \begin{bmatrix} 2B_B^{\times} & 0_3 & 0_3 \end{bmatrix} \tag{7.23}$$

$$R_{Mag} = \begin{bmatrix} \left(7\times10^{-7}\right)^2 & 0 & 0 \\ 0 & \left(7\times10^{-7}\right)^2 & 0 \\ 0 & 0 & \left(7\times10^{-7}\right)^2 \end{bmatrix} \tag{7.24}$$

The primary reason for developing the internal state vector concept is for error correction in the MEKF algorithm. Quaternions cannot be updated in an additive manner because it violates the unit quaternion norm constraint [27]. This is why the concept of an error quaternion in employed. The standard formation process of the internal state vector prior to resetting the state is shown in (7.25)

.

$$\delta x = \begin{bmatrix} a \\ \Delta\omega \\ \Delta\beta \end{bmatrix} = K_j \left[z_j - h\left(\hat{x}_j^-\right) \right] \tag{7.25}$$

where z is the measured quantity of a sensor, h($\hat{x}_j^-$) is the observation function relevant to that sensor and K is the Kalman gain.

However, the angular velocity and the bias states get updated in a tradition manner, as there are no restriction present for these quantities. However, the quaternion is a 4-dimensional quantity, so the update step requires a slight variation. Thus, the complete state and covariance updates are outlined

from (7.26) to (7.29).

$$\begin{aligned} \hat{x}_j^+ &= \hat{x}_j^- + \delta x \\ P_j^+ &= \left[I_9 - K_j H_j \right] P_j^- \end{aligned} \tag{7.26}$$

$$\hat{x}_{j,1:4}^+ = \hat{q}_j^+ = \delta q(a) \otimes \hat{q}_j^- \tag{7.27}$$

$$\hat{x}_{j,5:7}^+ = \hat{\omega}_j^+ = \hat{\omega}_j^- + \Delta\omega \tag{7.28}$$

$$\hat{x}_{j,8:10}^+ = \hat{\beta}_j^+ = \hat{\beta}_j^- + \Delta\beta \tag{7.29}$$

Upon the completion of this step process, the algorithm is looped in a continuous fashion producing real time state estimates. The system model discussed in this section is modelled in Simulink, as illustrated in Figure 7.1. It can be seen the steps defined in Table 7.1 are labelled as shown in that figure. The measurement update block make use of the observation functions, matrix along with relevant inertial frame vectors for the magnetometer and both sun sensor updates.

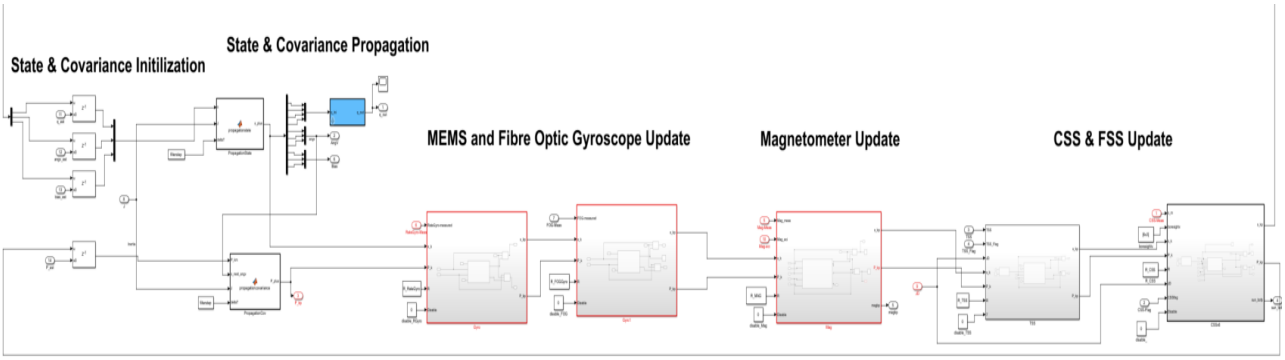


Figure 7.1: Simulink model developed as per the system model defined in (7.5).

7.2 Asynchronous Sensor Fusion Techniques

An important concept is implemented prior to operating the attitude determination filter to enhance all attitude sensors' asynchronous nature. Each sensor has its own output frequency range that it's capable of operating in. The gyroscopes and magnetometers are one of the fastest outputting sensors. However, both types of sun sensors don't have the ability to produce its measured quantities as frequently. Table 7.2 shows the frequency ranges of the sensors used for the ESSENCE mission and are explored in this thesis. However, to conduct analysis the frequency used in the simulated environment are shown in Table 7.3. Initially the values shown in Table 7.2 were used for the simulated tests. Due to the continuous crashing of Simulink caused by the limitations associated with the laptop used for the simulated tests, the values shown in Table 7.3 were used. These were chosen to simply demonstrate the concept of asynchronous in a computationally efficient manner. There are two approaches employed to mitigate the synchronization problem the zero-order hold, and a frequency update-based algorithm. The MEKF consisting of dynamics and kinematics-based system model is employed to analyze the accuracy for both approaches.

Table 7.2: Shows the frequency ranges for the attitude sensors used on the ESSENCE CubeSat.

Sensors	Frequency Ranges (Hz)
FSS	50 (max)
CSS	50 (max)
FOG	10 - 1000
MEMS Gyroscope	10 – 1000
Magnetometer	10 - 116

Table 7.3: Shows the frequencies used for the simulated testing environment.

Sensors	Frequency Ranges (Hz)
FSS	0.5
CSS	0.5
FOG	10
MEMS Gyroscope	10
Magnetometer	7.5

7.2.1 Zero Order Hold (ZOH) Technique: Results and Analysis

The Zero Order Hold (ZOH) technique is a common approach which synchronizes the sensors in any system to match the operational frequency of the MEKF. The approach utilizes a very simple logic, which uses a piecewise function. This function uses a base timestep that matches the operational frequency defined in Table 7.4. Therefore, the sensor output is set at a frequency of 10 Hz. The sensor updates the filter with a new value every 0.1 seconds or else the previous value is held until the next update. This is outlined in (7.30) which shows the function used in the zero-order hold technique where the inputs are the sensor outputs [99].

$$z_{ZOH}(u,t) = \sum_{i=1}^{\infty} u(i) \bullet rect\left(\frac{t - T/2 - iT}{T}\right) \tag{7.30}$$

where T is the ZOH's base timestep, t is the current timestep, "rect" is a rectangular/piecewise function and u is the input take from the sensor's output.

To initiate the MEKF process, several parameters are required to be defined for the attitude determination filter. These are outlined in Table 7.4 which is comprised of initial state estimates

and covariance values, sensor noise matrices and operational frequency.

Table 7.4: Outlines the initial parameters used for the Dynamics and Kinematics based MEKF.

Parameters	Values
Initial Quaternion Estimate (q_{est})	$\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$
Initial Angular Velocity Estimate (ω_{est})	$\begin{bmatrix}0.1 & 0.1 & 0.1\end{bmatrix}$ rad/sec
Initial Bias Estimate (β_{est})	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}$ rad/sec
Initial Covariance (P_{est})	$\begin{bmatrix} 3\left(\frac{\pi}{4}\right)^2 I_3 & 0_3 & 0_3 \\ 0_3 & \left(1\times10^{-3}\right)^2 I_3 & 0_3 \\ 0_3 & 0_3 & \left(\frac{15\pi}{3600\times180}\right)^2 I_3 \end{bmatrix}$
Filter Timestep (Δt) , Operational Frequency (Fs)	0.1 sec , 10 Hz

Nominal Conditions using ZOH technique

The condition defined as "nominal" is based off the operational frequency defined in Table 7.4. The simulation was set for a duration of 10 obits, where 1 orbit lasts 5400 seconds. The outputs of this filter are to convey various information within the attitude determination system. However, the relevant one for this thesis revolve around the quaternion and the angular velocity of the satellite. Thus, Figure 7.2 shows the plots for the quaternion and angular velocity estimates and, the true quaternion and angular velocity values. Figure 7.3 shows the error in both states is showed, the attitude error represented in Roll, Pitch and Yaw angles, and the eclipse flag. The MEKF shows the convergence takes place in approximately ~668 seconds or ~11 minutes. Following the initial convergence, the error in the states are observed to be significantly lower than any of the static

estimation algorithms. The use of gyroscopes reduces the error in states by measuring a time varying quantity. This is because the gyroscope's drift cannot be corrected in any of the static methods. However, the MEKF state estimation algorithm enhances this sensor well in the update step to reduce the bias in the system. All the error values expressed in terms of Euler angles are extracted after the formulation of the error quaternion. The error quaternion represents the comparison of the true and quaternion state. This is the case for every MEKF implementation mentioned in this thesis. Upon the completion of the simulation, the RMS error in Roll, Pitch and Yaw angles are 8.2563 deg, 5.1192 deg and 10.2282 deg respectively. Thus, showing an improvement in the attitude angles by ~70% when compared to static attitude estimation algorithms. In the eclipse periods, the error quaternion deviation from an approximate 0 convergence grows. The error does not grow rapidly and does not oscillate as time progresses, unlike the static estimators. The maximum error seen during these times aren't beyond ~±20 deg for all 3 Euler angles. This shows the filter's ability to rely on the system model while in eclipse.

The angular velocity state utilizes both types of gyroscopes to predict the bias in the system to generate an accurate angular rate outcome. The RMS error values are 9.281×10^{-4} rad/sec, 6.580×10^{-4} rad/sec, and 9.555×10^{-4} rad/sec. The main source of the error for this quantity is the MEMS gyroscope. The noise acting on these devices are higher than any decoupled sensor such as the Fibre Optic gyroscope. Therefore, the MEKF attitude algorithm is proven to be much more accurate when compared to previously analyzed vector-based algorithms.

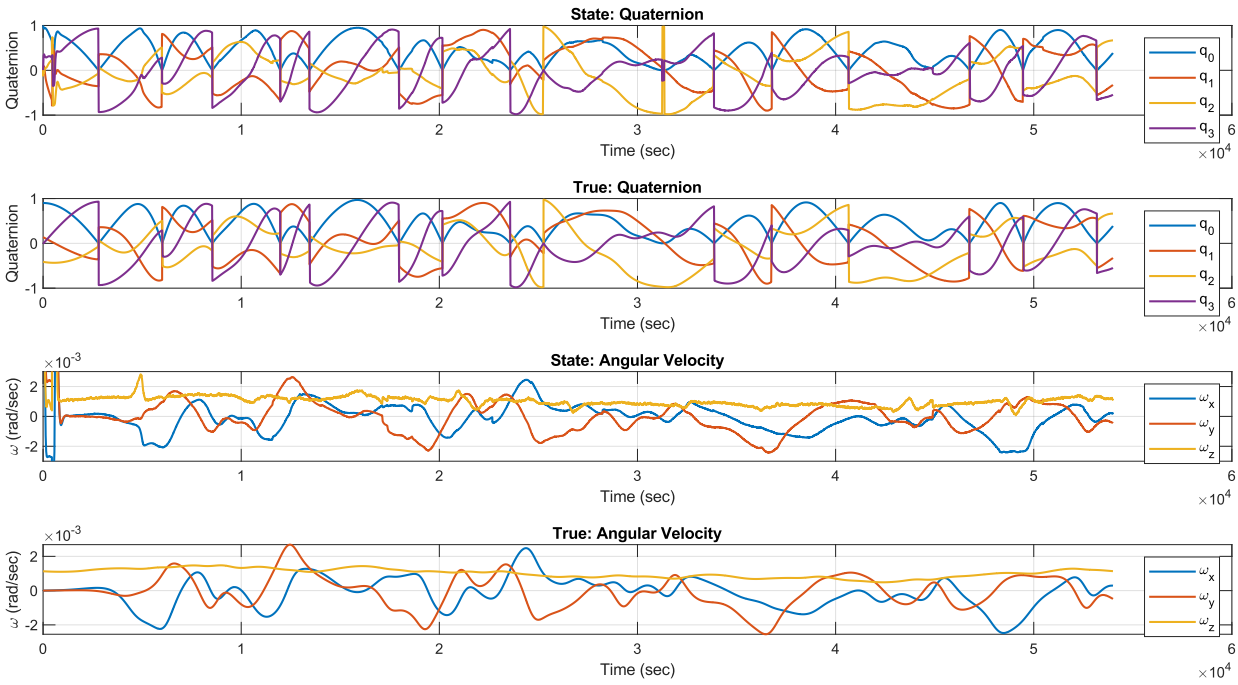


Figure 7.2: Plots displaying the quaternion and angular velocity states and the true quaternion and angular velocity quantities using the ZOH technique.

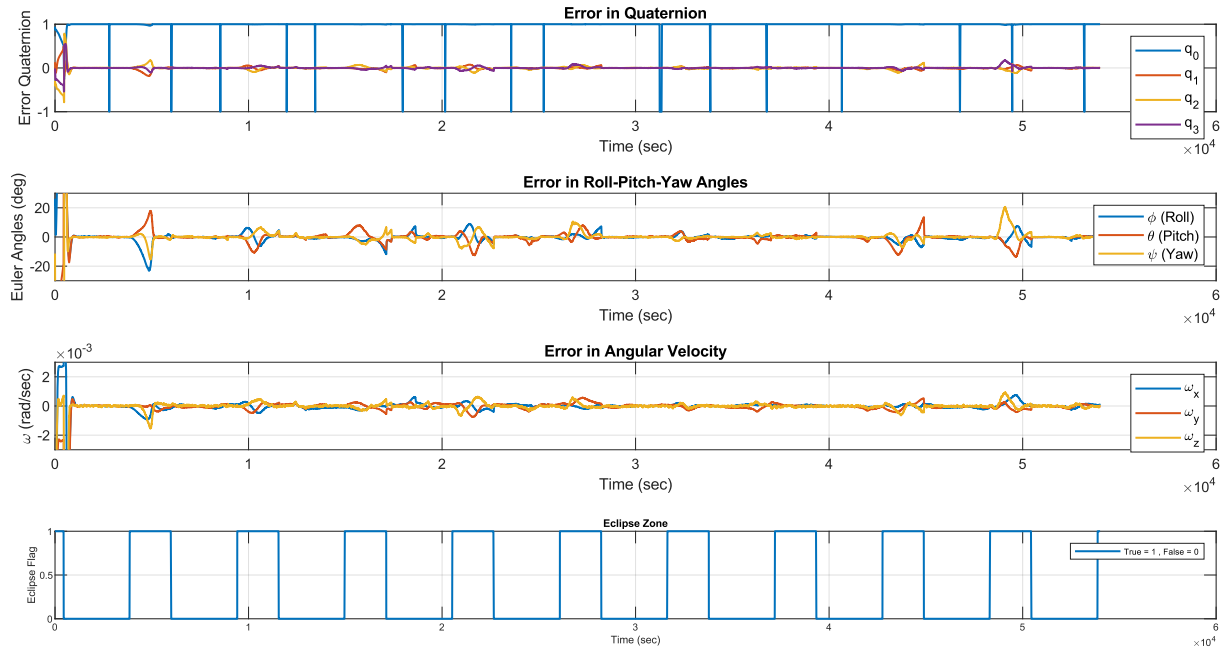


Figure 7.3: Plots displaying the error quaternion, Euler angles, angular velocity, and eclipse flags quantities using the ZOH technique.

7.2.2 Frequency Based Update Algorithm (FBUA): Results and Analysis

In addition to the ZOH technique, a frequency-based algorithm is developed. The concept behind the frequency-based update algorithm (FBUA) is that it simply checks if the sensor has an output that matches the time instance of the MEKF. This check is utilizing the frequencies defined in Tables 7.2 and 7.3. The FBUA logic shown in (7.31) produces flags for further processing. Thus, if the outcome of the logic is "0" it communicates to the MEKF that this sensor can be used for the filter update.

$$f(t) = \mathrm{mod}\left(t, \frac{F_s}{F_{sen}}\right) \tag{7.31}$$

where t is the current time instance, F_s is the operational frequency and F_{sen} is the sensor's outputting frequency.

Figure 7.4 shows the error plots using this approach where the assumption is that all sensor outputting frequencies are known and are taken from Table 7.2 for this simulation test. The RMS error in terms of the Euler angles are 9,8568 deg, 7.3336 deg and 8.6433 deg and the RMS error in angular rates are 1.5×10^{-3} rad/sec, 2.5×10^{-3} rad/sec, and 4×10^{-4} rad/sec. When compared to the ZOH method, the error difference between the Roll and Yaw axes aren't large. However, the Pitch axis shows a large increase in inaccuracy in the error values. Moreover, the angular rates also show an increase in error in all three axes, proving the ZOH technique is a better. Therefore, the ZOH method is selected for both system models applied in the MEKF in this thesis.

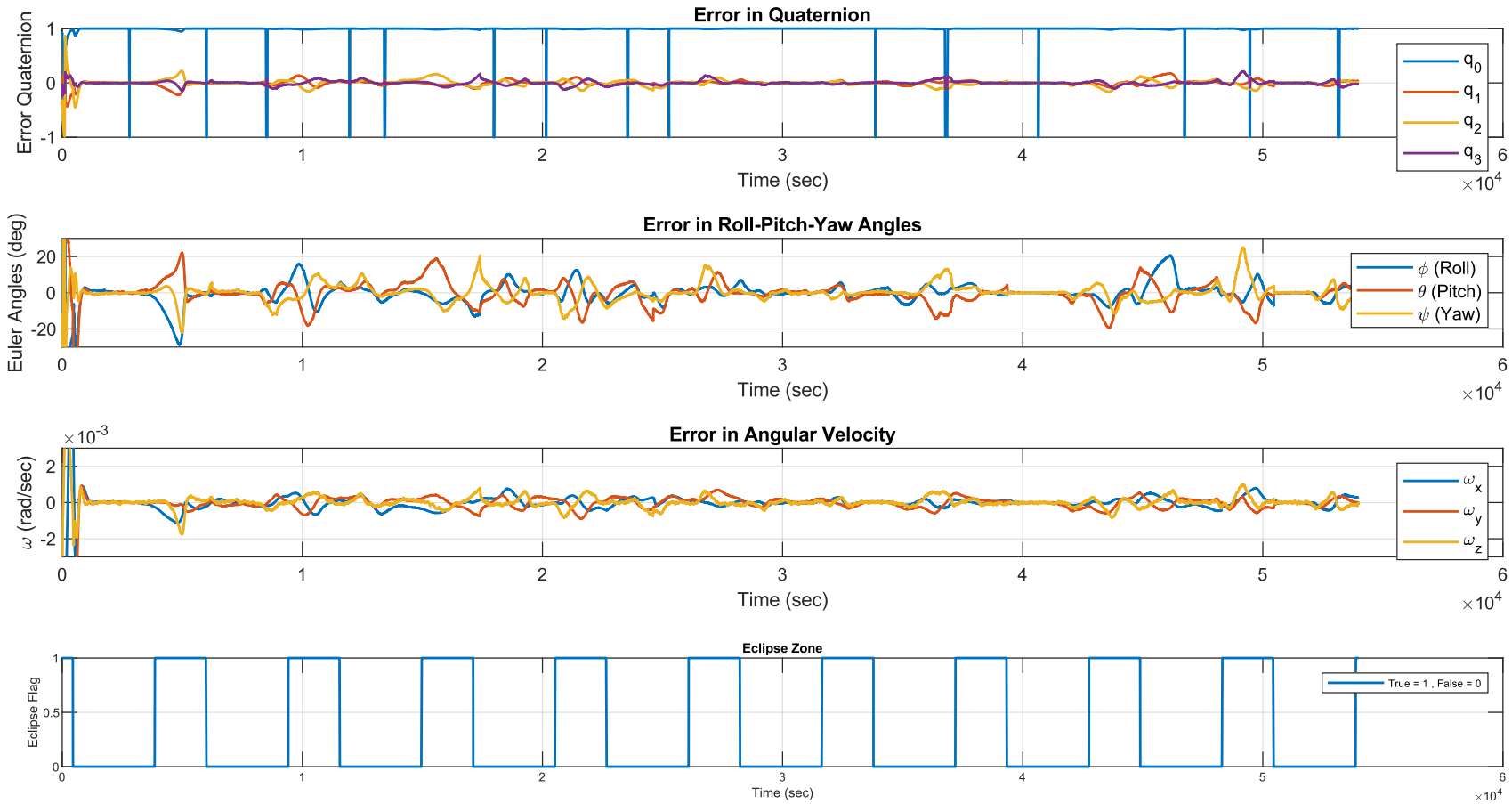


Figure 7.4: Plots displaying the error quaternion, Euler angles, angular velocity, and eclipse flags using FBUA.

7.3 System Analysis: Dynamics and Kinematics based MEKF

To further demonstrate the ability of the dynamics and kinematics based MEKF, a thorough system analysis is done by analyzing various forms of single point failures. Static attitude estimation requires each sensor to be available at all times, or else the error values grow uncontrollably. A glimpse of this is shown with the absence of the sun sensors during the eclipse periods. However, other single point failures also occur, and they are not caused by natural causes. Therefore, the effects of single and multiple sensor failures are analyzed. In addition, there are occurrences where human errors take place, in which the sensors are either misconfigured or misaligned. Lastly, accuracy is additionally analyzed by varying the operational frequency. All the mentioned tests are conducted at the same operational frequency defined in the nominal case, apart from the final test.

Individual Sensor Failure

In a low earth orbit, one of the most critical sensors is the magnetometer. This is because the earth's magnetic field is best measured in this environment. Thus, upon failure it can be seen large error fluctuations take place as shown in the error plots in Figure 7.5. The RMS error in terms of the Euler angles grows to 99.7271 deg, 36.9720 deg and 87.8691 deg. The sun sensors are not functional when the satellite is in eclipse and when the sun isn't in its FOV. To account for the second constraint, visibility flags are derived to communicate if the sensor can be used or not. This is a geometric constraint rather than an environmental one. Thus, during eclipse phase this affects the accuracy leads to a sudden growth upon even with the correction of states using the other sensors. Apart from the sun sensors, magnetometer is the only sensor which updates the quaternion state directly. This doesn't affect the modelling of the angular velocity as much due to the availability of both gyroscopes. However, a slight increase in error for the angular velocities is

experienced. This is because the MEKF draws relationship to all of the states from sensors. The angular rates error values are observed to be 2.8 x 10^{-3} rad/sec, 2.5 x 10^{-3} rad/sec, and 4.3 x 10^{-3} rad/sec in the angular velocity. Although an increase in mean error is concerning, this event is not likely to occur as it is a MEMS device. The event is treated as a single sensor failure but in reality, multiple failures in OBC would occur for this single event to happen. Thus, this demonstrates of the level of importance this sensor holds in a LEO mission. A common engineering practice of using redundant sensors can be used to mitigate a magnetometer failure. Small integrated circuit chips that are configured for the PC104 standard can be used, as they are compatible with any CubeSat. However, it would be critical to ensure that the backup magnetometer must be off when not in use. The results of not doing so leads to added sensor noise, potential subsystem failure and potential CubeSat failure.

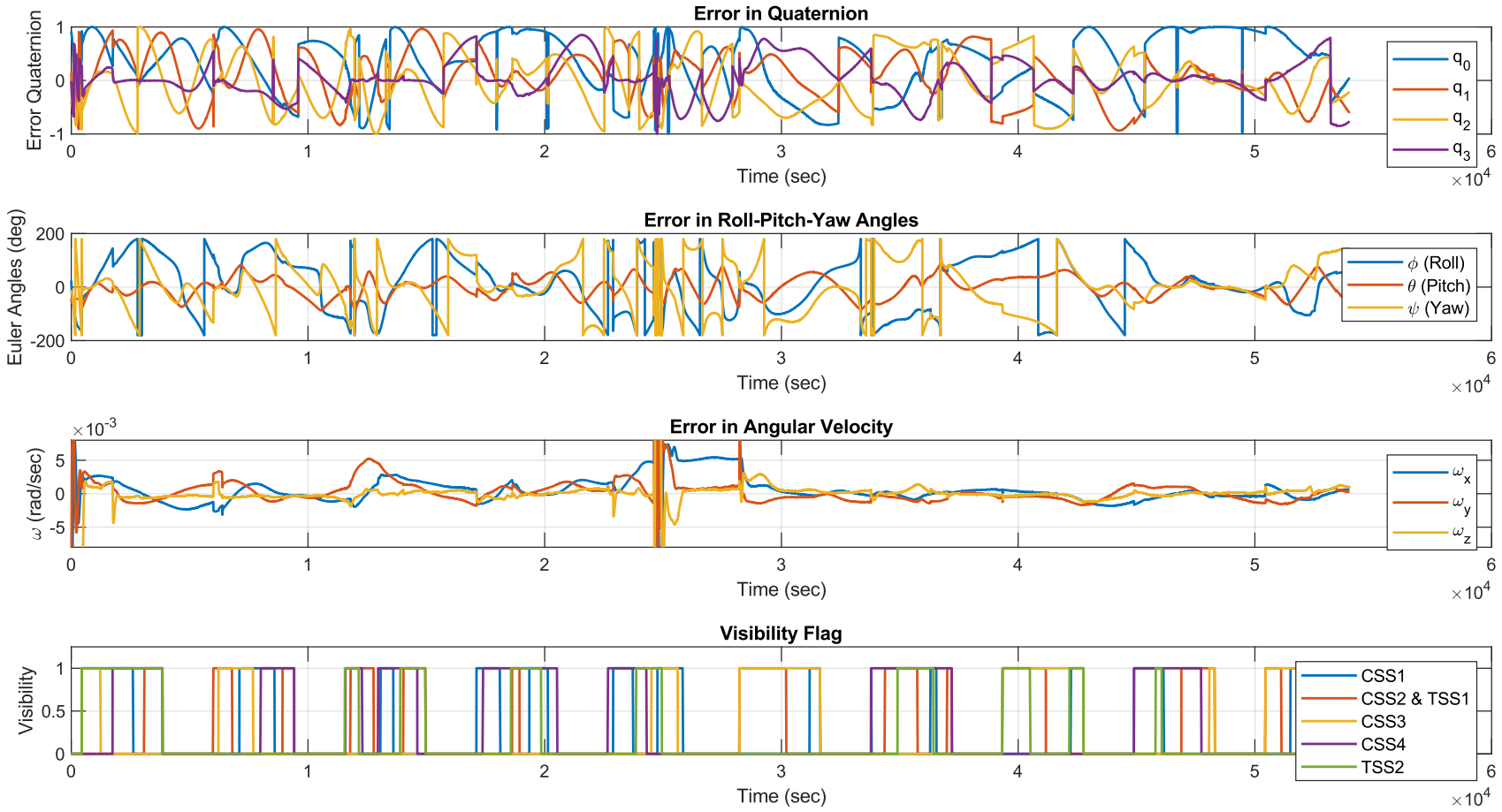


Figure 7.5: Magnetometer failure plots displaying the error quaternion, Euler angles, angular velocity, and visibility flags for the dynamics and kinematics based MEKF.

In addition to the magnetometer, a failure in gyroscope is also explored. The effects are expected to be seen in the angular velocity state rather than the quaternions. The Fibre Optic gyroscope was interpreted as the failed sensor. The plots shown in Figure 7.6 display the error seen in the quaternion and angular velocity states. The quaternions states are affected because of the quaternion kinematics equation used in for the system model incorporates the angular velocity vector. Thus, while propagating the state variables the increased error from the angular rates is carried further to the quaternion states as well. The RMS error shows an increase in terms of Euler angles are 12.6339 deg, 5.5378 deg and 15.1883 deg. The error in angular rates are 1.1×10^{-3} rad/sec, 1.8×10^{-3} rad/sec, and 8×10^{-4} rad/sec. It can be seen the error is lower than the magnetometer failure scenario. There are two key reasons for this result. First is that there is a redundancy present, as the MEMS gyroscope is utilized. Second, the magnetometer is available as the previous scenario showed its importance for the MEKF algorithm. Therefore, these outcomes show that a gyroscope failure is feasible in case singular sensor failure occurs. Similar to the magnetometer, redundancy can be used in this case as well. This is implemented in the current scenario as a MEMS gyroscope is used to account for the loss of FOG. FOGs are fairly expensive to purchase, so MEMS gyroscopes are often taken as the backup. However, if another FOG can be used for redundancy, it should be placed 45 deg between any two BFF axis. This would entail angular velocity can be measured for two axes, rather than one or even none.

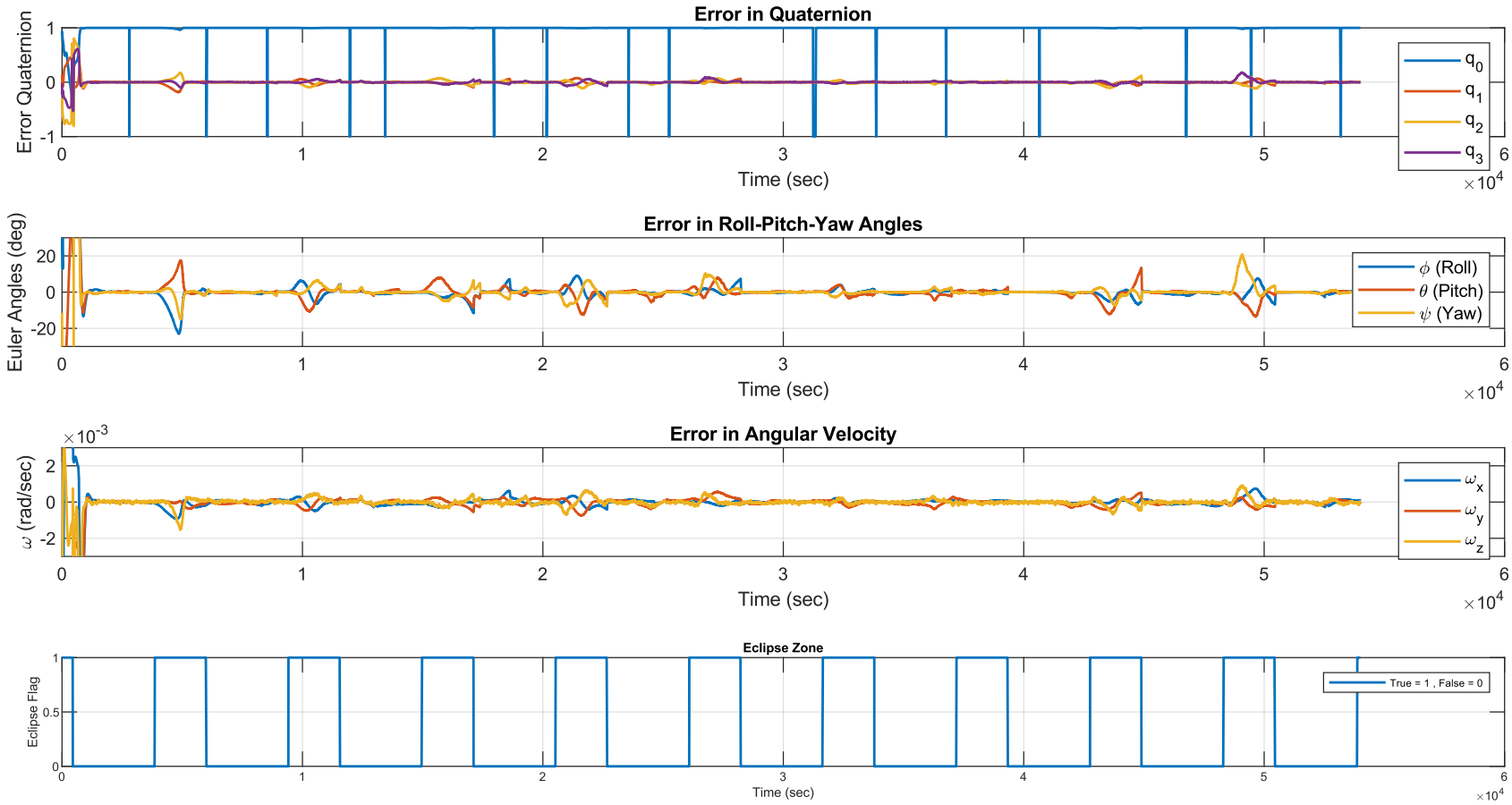


Figure 7.6: Fibre Optic gyroscope failure plots displaying the error in quaternion, Euler angles, angular velocity, and eclipse flag for the dynamics and kinematics based MEKF.

Multiple Sensor Failures

From the single sensor failure analysis, it is evident that failure in magnetometer affects the attitude estimates drastically. However, an accumulation of multiple sensor failure leads to a larger issue for attitude determination. An analysis of multiple points of failure was conducted, where CSS, FSS and FOG were assumed to have failed. It can be hypothesized the error increases across both quaternions and angular velocity states. Similar to the single sensor failure analysis, error plots were produced shown in Figure 7.7. It can be seen the error is handled by the dynamics and kinematics-based system model very well. This is because the system is able to utilize the magnetometer to the filter's advantage. The RMS error found in terms of Euler angles are 57.0284 deg, 25.1101 deg and 48.3372 deg, and the error in the angular velocity vector is 1.6×10^{-3} rad/sec, 9.0×10^{-4} rad/sec, and 1.7×10^{-3} rad/sec. These values are expected because, the MEMS gyroscope and magnetometer are the only available sensors for the update step. The error experienced is caused by the bias found in the MEMS sensors. Hence, largely affecting the quaternion estimates produced by the MEKF. The presented scenario is catastrophic as it demonstrates multiple points of failure. The only course of action is to produce attitude estimates of the available sensors. For this scenario the only available sensors were the MEMS gyroscope and magnetometer. Thus, the natural rise in RMS error was expected.

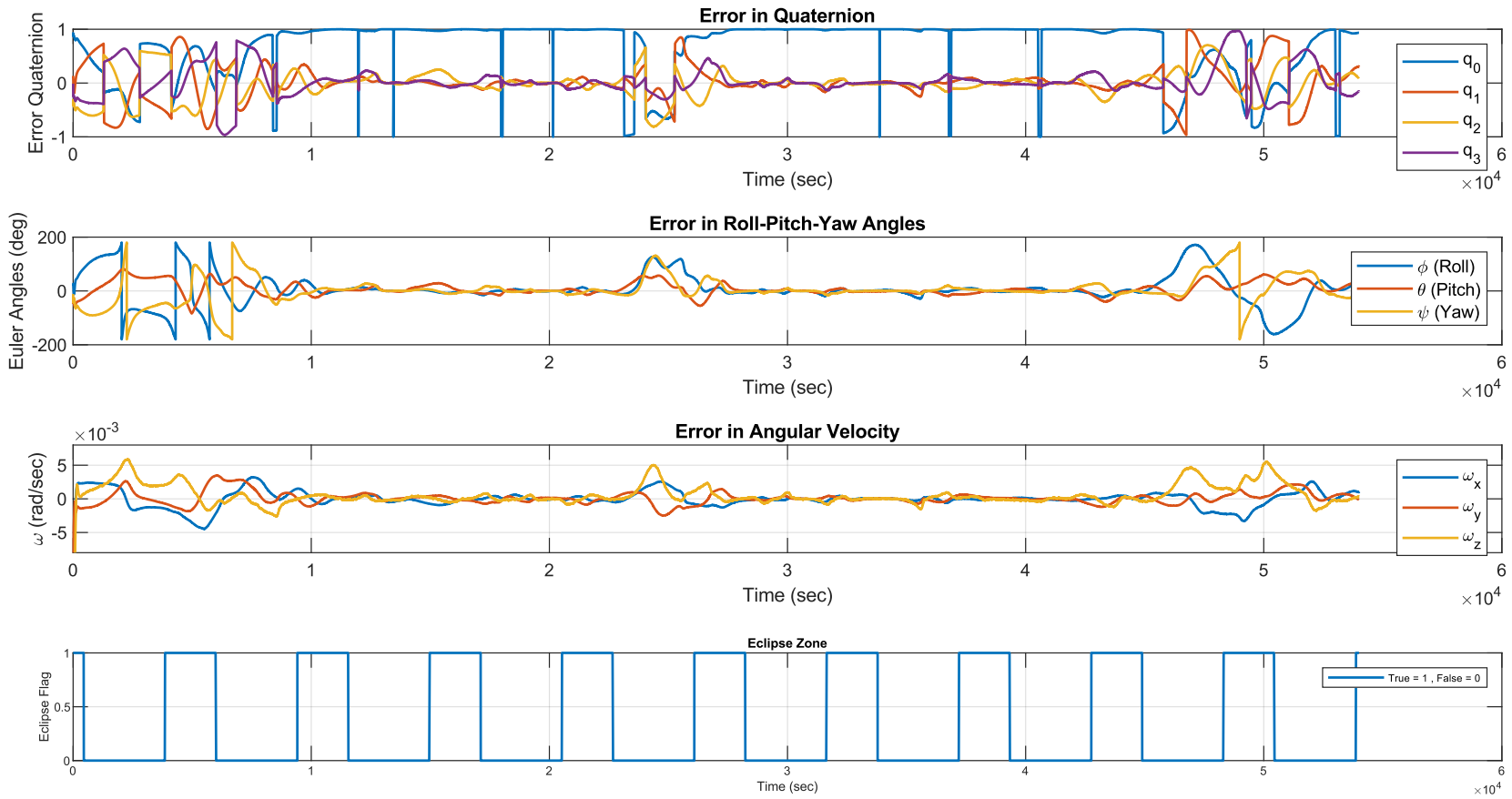


Figure 7.7: Multiple sensor failure plots displaying the error in quaternion, Euler angles, angular velocity, and eclipse flags for the dynamics and kinematics based MEKF.

Sensor Misalignment and Misconfiguration

The third system test investigates possible misalignment and misconfiguration of the decoupled sensors. First, a misalignment is considered for the FOG. This is typically case of human error when external components are incorporated in a CubeSat. For this experiment a 3-2-1 rotational sequence is performed to alter the optical gyroscope's alignment from the BFF. The rotational angle applied at each rotation is outlined in Table 7.5.

Table 7.5: Shows the angles used for a 3-2-1 rational sequence to produce a misaligned matrix.

Angles	Values
Alpha (x-axis)	-35 deg
Beta (y-axis)	48 deg
Gamma (z-axis)	60 deg

Figure 7.8 demonstrates the error plots caused by this human error. The attitude error in terms of Roll, Pitch and Yaw are 9.3114 deg, 5.3092 deg and 11.3075 deg, respectively. Moreover, the error in angular rates are 9.0×10^{-4} rad/sec, 1.3×10^{-3} rad/sec, and 1.0×10^{-3} rad/sec. The increase in error is expected as the misalignment causes the angular velocity being measured in its own axis rather than the body fixed frame. This is problematic for the MEKF as it assumes all the measured quantities are transformed in terms of the body fixed frame for the state update step. The current scenario also demonstrated a mitigation technique. The presence of MEMS gyroscope allowed the correction of the inaccuracies caused by the misaligned FOGs. Hence, showing how effective

having backup sensor is for attitude determination. Therefore, leading to an acceptable set of error values if such a human error occurs.

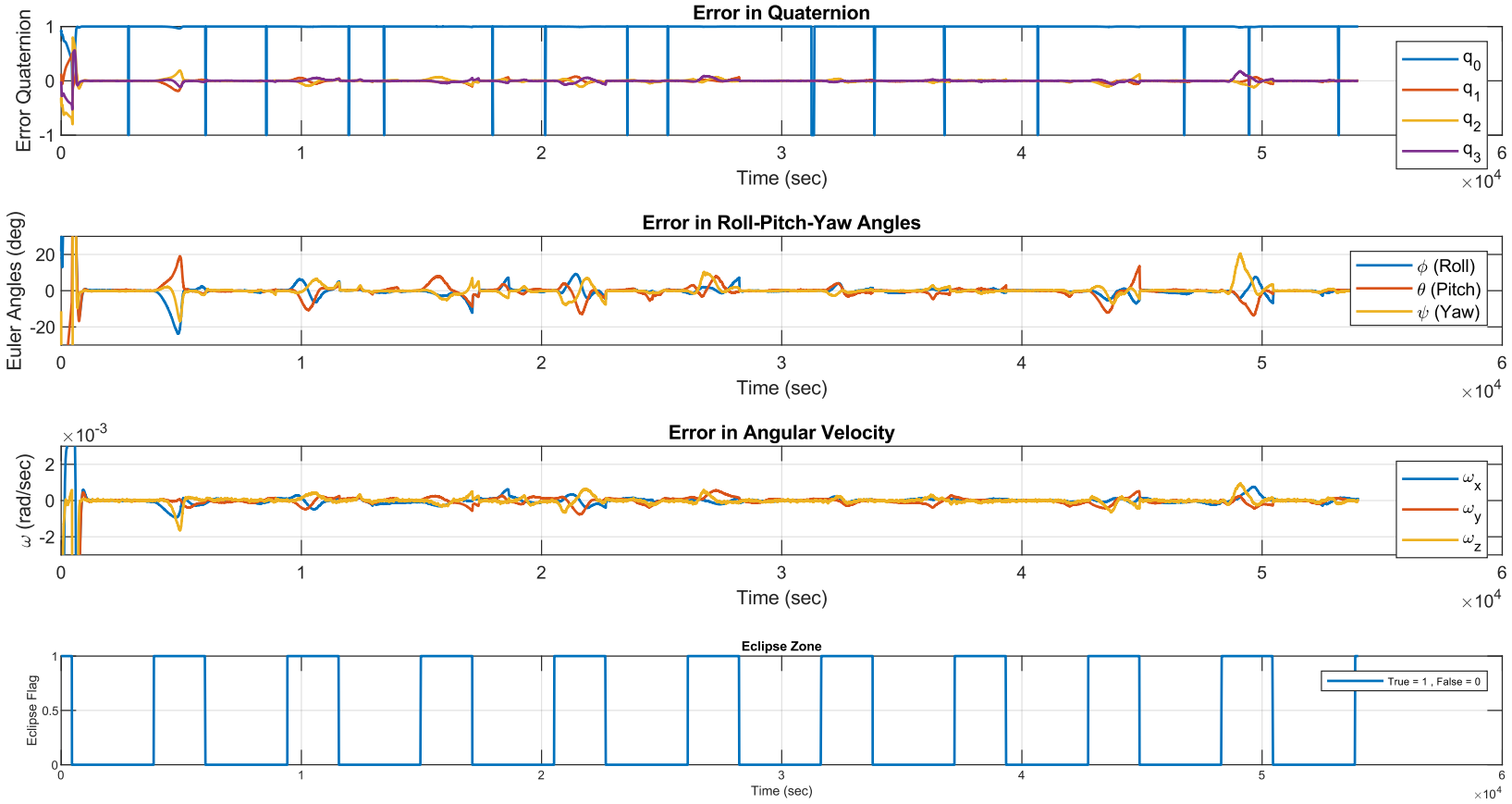


Figure 7.8: Sensor misaligned plots displaying the error in quaternion, Euler angles, angular velocity, and eclipse flag.

The misconfiguration error is considered in case of a possible misinterpretation on the placement of the CSS. As previously mentioned, the CSSs are placed along the body. For this simulated experiment, it is assumed that all CSSs placed on the +X face of the CubeSat are interpreted as if they are +Z. This entails there are sensors present on the +X side but are not used correctly. The mean error values in terms of Euler angles are 45.6783 deg, 23.6520 deg and 44.8150 deg, and the error in angular rates are 1.6×10^{-3} rad/sec, 2.0×10^{-3} rad/sec and 2.1×10^{-3} rad/sec. Figure 7.9 describes these error values in term of error quaternion, error in Euler angles and angular rates. It can be seen due to the lack of attitude knowledge in the +X direction, the visibility and the environmental effects present for the CSS increase significantly. Thus, the visibility flag for the +X sun sensors used in the filter is not interoperating the illuminated CSS correctly. Hence, a blind spot on one side of the satellite results from such a misconfiguration of sensors. Thus, the error while not in eclipse phases is equally or much worse than the ones observed in the eclipse zones. However, the mean error values are still less than multiple sensor failures as only one string of CSS is mistaken. Thus, making all the other sensors offset this error source and minimize the error using the MEKF. In case a misconfiguration of a CSS takes place, and external analysis can be done to correct these errors. This would entail downlinking the measured values of all CSSs for further interpretation. CSS are often placed along the body of the CubeSat; thus, duplicate measurements can easily be identified. Once this is done, the correction matrix can be uplinked to correct for the faulty sensor.

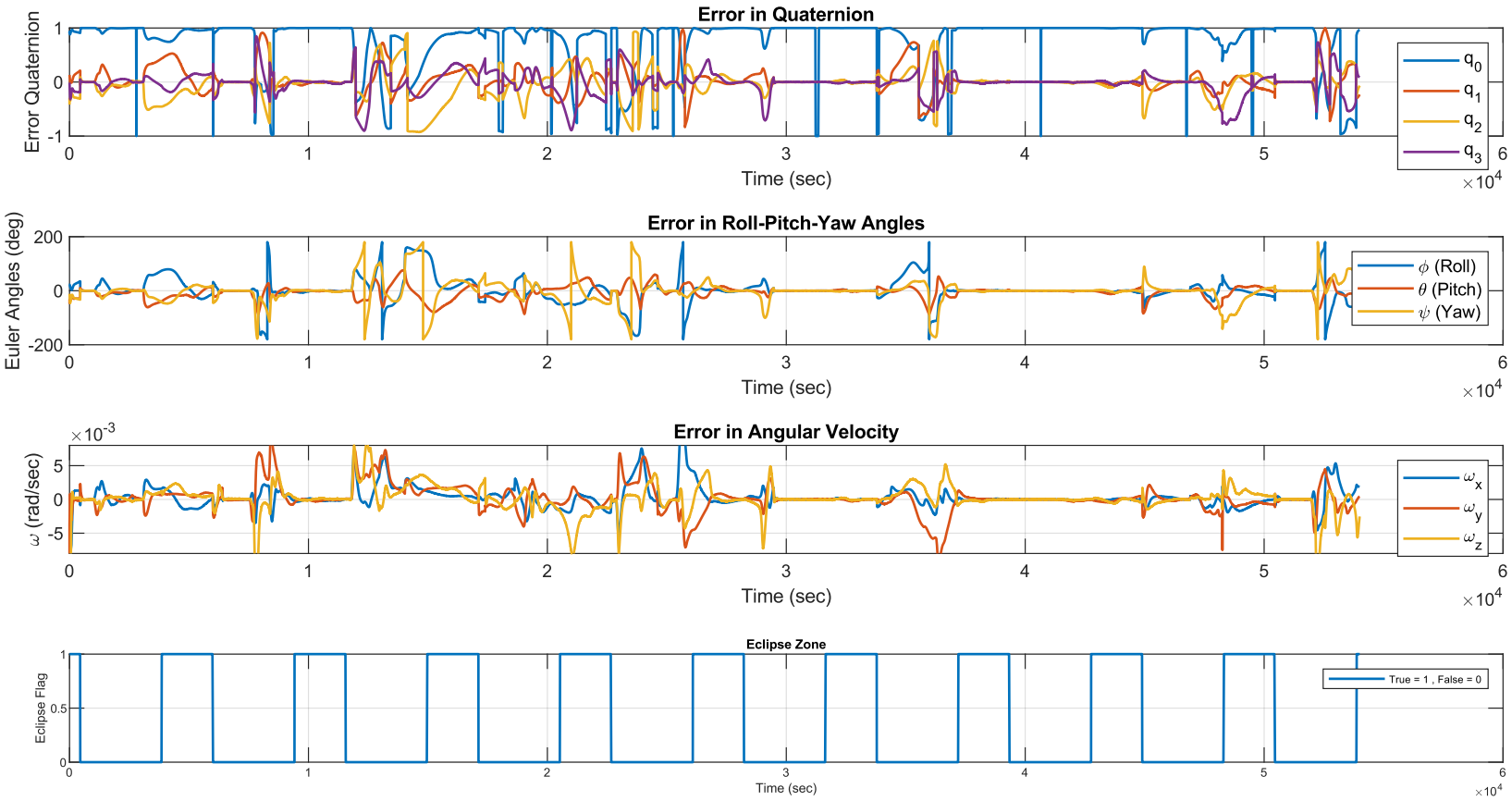


Figure 7.9: Sensor misconfiguration plots displaying the error in quaternion, Euler angles, angular velocity, and eclipse flag.

Various Update Frequencies

The final system analysis was done to examine how accurate the results get when the operational frequencies are varied. The two frequencies analyzed other than the nominal frequency showed in Table 7.4 are 1 Hz and 100 Hz. This is to simply to provide a low and high variation in the operational frequencies. It is theorized that the higher the frequency, the better the accuracy as the updates from the sensors come more often. Tables 7.6 and 7.7 shows the error in Euler angles and angular rates for 3 cases which included the ones seen in the nominal case from the zero-order hold technique. As seen in both the tables, the value show a decrease in pitch angle error as the frequency values are increased. Operating the ADCS at high frequencies affects the performance of the CubeSat. This is because key subsystems such as the power and command and data handling are overused for just a slight improvement. The in error at 100 Hz when compared to the nominal case the improvement is not very significant. Hence, the accuracy in terms of system performance cost isn't optimal. Therefore using 10 Hz is the suitable choice for the MEKF specific to the system model described in (7.5).

Table 7.6: Shows the RMS error in the Euler angles for all three operational frequencies.

	Error in Euler Angles		
Operational Frequencies	**Roll (ϕ)**	**Pitch (θ)**	**Yaw (ψ)**
1 Hz	10.4324 deg	7.3863 deg	8.8999 deg
10 Hz (Nominal)	8.2563 deg	5.1192 deg	10.2282 deg
100 Hz	1.0800 deg	-0.3434 deg	-0.6381 deg

Table 7.7: Shows the RMS error values in the angular velocity for all three operational frequencies.

	Error in Angular Velocity		
Operational Frequencies	**X-axis Angular Velocity (ω_x)**	**Y-axis Angular Velocity (ω_y)**	**Z-axis Angular Velocity (ω_z)**
1 Hz	1.6 x 10^{-3} rad/sec	2.5 x 10^{-3} rad/sec	6.0 x 10^{-4} rad/sec
10 Hz (Nominal)	9.281 x 10^{-4} rad/sec	6.580 x 10^{-4} rad/sec,	9.555 x 10^{-4} rad/sec
100 Hz	1.0 x 10^{-3} rad/sec	1.4 x 10^{-3} rad/sec,	1.4 x 10^{-3} rad/sec

7.4 Attitude Estimation using MEKF Algorithm: Kinematics Only

The system model proposed in section 7.1 is altered to reduce the computational load on the OBC. This model makes use of the quaternion kinematics equation and the bias equation but doesn't consider the Euler's equation of motion for its system model. However, it is required to have a source to determine the angular velocity of the CubeSat and use it for the quaternion kinematics. Thus, it is required to make utilize either the Fibre Optic Gyroscope or the MEMS Gyroscope directly in the propagation step. Furthermore, to account for remove the bias from the gyroscope it is required to be accounted before they are fed into the quaternion kinematics equation. The updated states, internal state and system model is described in (7.32), (7.33), and (7.34) respectively.

$$x = \begin{bmatrix} q & \beta \end{bmatrix} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 & \beta_x & \beta_y & \beta_z \end{bmatrix} \tag{7.32}$$

$$\delta x = \begin{bmatrix} a & \Delta\beta \end{bmatrix} \tag{7.33}$$

$$\dot{x} = \begin{bmatrix} \dot{q}_{BI} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\Xi\left(q_{BI}\right)\breve{\omega}_B^{BI} \\ \eta_u \end{bmatrix}$$
$$\breve{\omega}_B^{BI} = \omega_{B,meas}^{BI} - \beta \tag{7.34}$$

The process described in Table 7.1 of the previous system model is employed but is modified to the kinematics-based system model. The main purpose of both forms of MEKF is to communicate the orientation of the satellite. Thus, the elimination of the angular velocity state allows the filter to strictly produce quaternion as its output state. This also results in the reduction of the size of covariance matrix is from 9x9 matrix to 6x6. Thus, the state transition matrix (F) and the disturbance torque error matrix (G) are updated and shown in (7.35) and (7.36) respectively. The process noise matrix (Q) does not require modification, as the environmental effects don't change.

$$F = \begin{bmatrix} -\omega^{\times} & -\frac{1}{2}I_3 \\ 0_3 & 0_3 \end{bmatrix} \tag{7.35}$$

$$G = \begin{bmatrix} -\frac{1}{2}I_3 & 0_3 \\ 0_3 & I_3 \end{bmatrix} \tag{7.36}$$

In this model the sensors used are CSS, FSS and the magnetometer for the update step. Due to the lack of angular velocity correction, the bias state correction is dependent on the attitude sensors which is not ideal. This is because neither of the mentioned sensors contain the bias parameter in their measurement model. The selection of gyroscope is done through a comparison analysis between the Fibre Optic gyroscope and the MEMS gyroscope. The selection test is conducted in nominal conditions where Table 7.8 describes all the initial parameters for the MEKF. Figure 7.10 shows the formulation of the MEKF in Simulink using the system model described in (7.34).

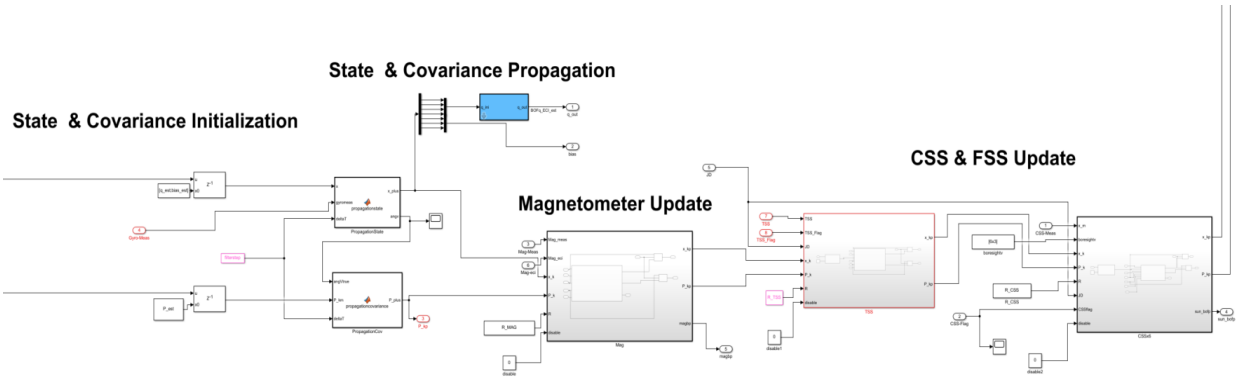


Figure 7.10: Simulink model developed as per the system model defined in (7.34) for the kinematics based MEKF.

Table 7.8: Shows the nominal conditions for the kinematics based MEKF.

Parameters	Values
Initial Quaternion Estimate (q_{est})	$\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$
Initial Bias Estimate (β_{est})	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}$ rad/sec
Initial Covariance (P_{est})	$\begin{bmatrix} 3\left(\frac{\pi}{4}\right)^2 I_3 & 0_3 \\ 0_3 & \left(\frac{15\pi}{3600\times 180}\right)^2 I_3 \end{bmatrix}$
Filter Timestep (Δt) , Operational Frequency (Fs)	0.1 sec , 10 Hz

Nominal mode – Fibre Optic Gyroscope

In this test, the angular velocity input to the propagation step is from the Fibre Optic gyroscope. Much like the all the other tests, the simulated test ran for 10 orbits under the nominal conditions.

Figure 7.11 shows the predicted quaternion, true quaternion, and the eclipse flags. The convergence of this filter occurs at ~550 seconds which is approximately 2 minutes faster than the previous system model. This MEKF system model utilizes the ZOH technique because it was proven to be the better approach to enhance the asynchronous nature of all the sensors. The RMS error is in terms of Roll, Pitch and Yaw angles are 9.5556 deg, 4.6762 deg and 11.1560 deg. Although the error values don't show a drastic improvement, but during eclipse the system model is much better equipped to produce accurate attitude estimations. This is illustrated in Figure 7.12 which shows the error plots of this test. When compared to the dynamics and kinematics-based system model, the accuracy in the quaternion state shows a significant improvement. The maximum error values across all three Euler angles reaches ~ ±7.5 deg, which is ~13.5 deg more accurate than the previous system model. Therefore, it shows that this model is better suited to handle attitude determination than the dynamics and kinematics-based system model.

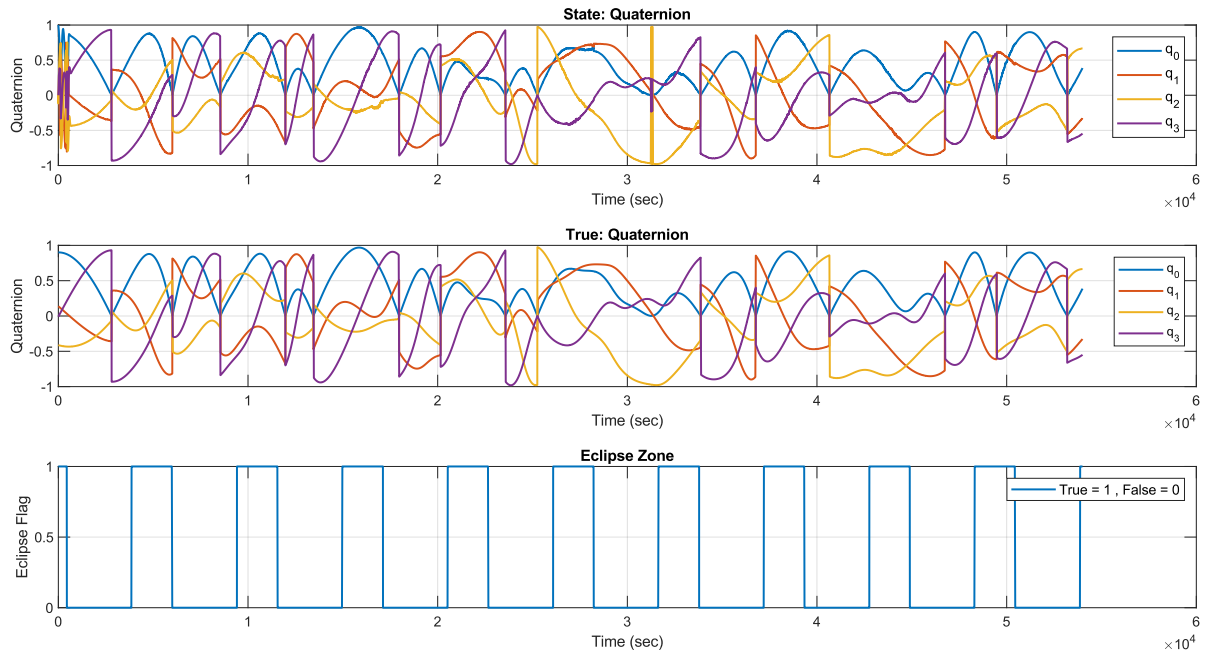


Figure 7.11: Kinematics based MEKF using FOG plots displaying the quaternion state, true quaternion, and eclipse flag.

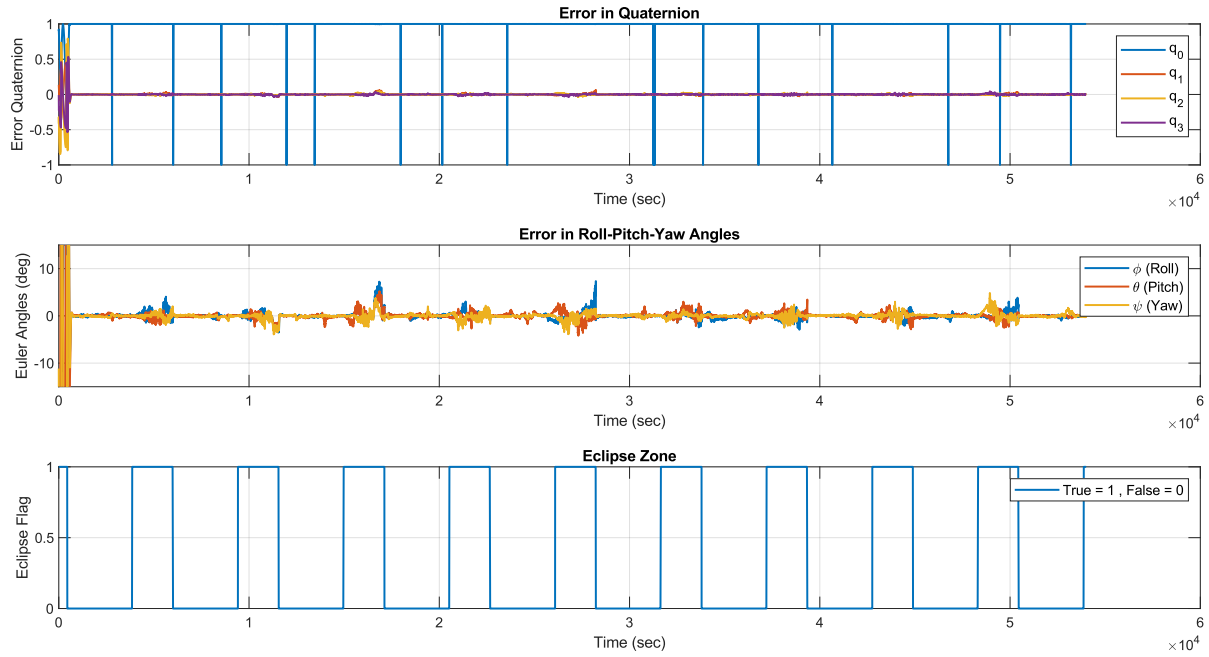


Figure 7.12: Kinematics based MEKF using FOG plots displaying the error quaternion, error in term of Euler angles, and eclipse flag.

Nominal mode – MEMS Gyroscope

Similar analysis was conducted with the MEMS gyroscope to determine if the accuracy improves by using this sensor. It is hypothesized that the MEMS gyroscope would produce inaccurate attitude estimates when compared to the Fibre Optic gyroscope. Figure 7.13 shows the state and true quaternion comparison along with the eclipse flag, and Figure 7.14 shows the error plots and eclipse flags generating from this test. The convergence time using the MEMS gyroscope increases to ~600 seconds which is 50 seconds slower than the Fibre Optic gyroscope. Moreover, the RMS error values in term of Roll, Pitch and Yaw are 10.1369 deg, 5.1509 deg and 11.6555 deg. The error value increases because of the significant increase of the gyroscopic bias in the system. However, the error quantities are still better than the ones seen in the dynamics and kinematics-based system model. When in eclipse the increased error is more prominent due to the lack of

correction in the bias state. This is seen as the values reach a maximum value of ~ ± 24.5356 deg. This worse than both dynamics and kinematics-based system model and the kinematics model which uses FOG. Therefore, using the FOG is the better choice as they have low noise and bias parameters when compared to MEMS gyroscope.

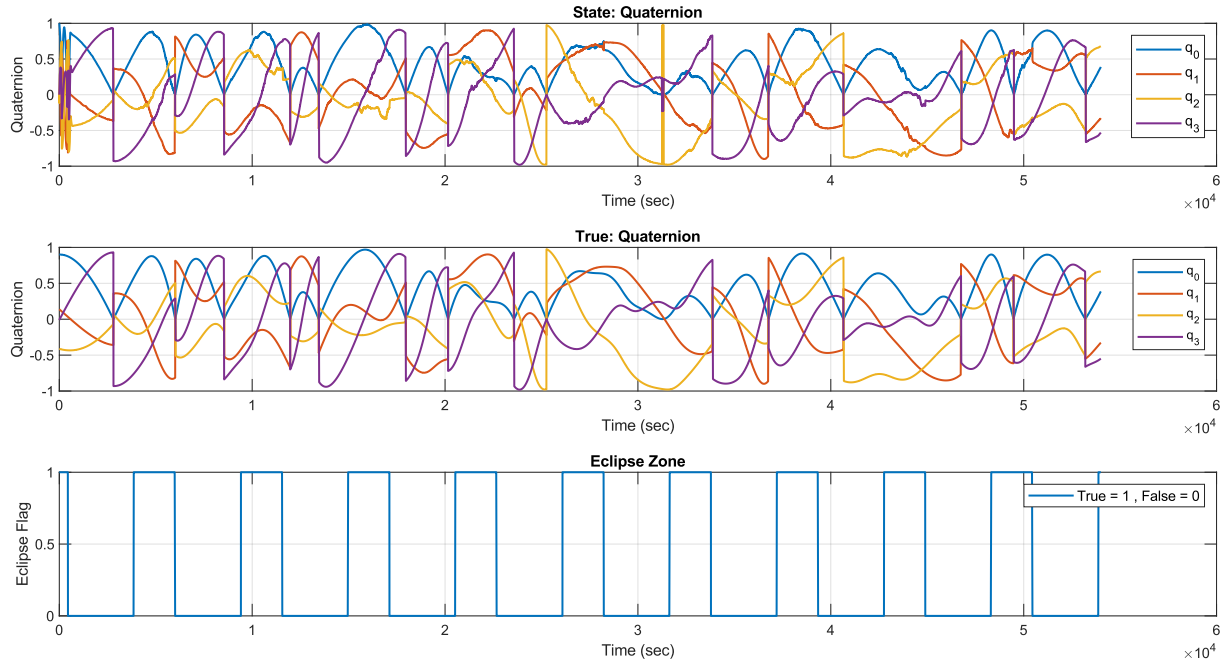


Figure 7.13: Kinematics based MEKF using MEMS Gyroscope plots displaying the quaternion state, true quaternion, and eclipse flag.

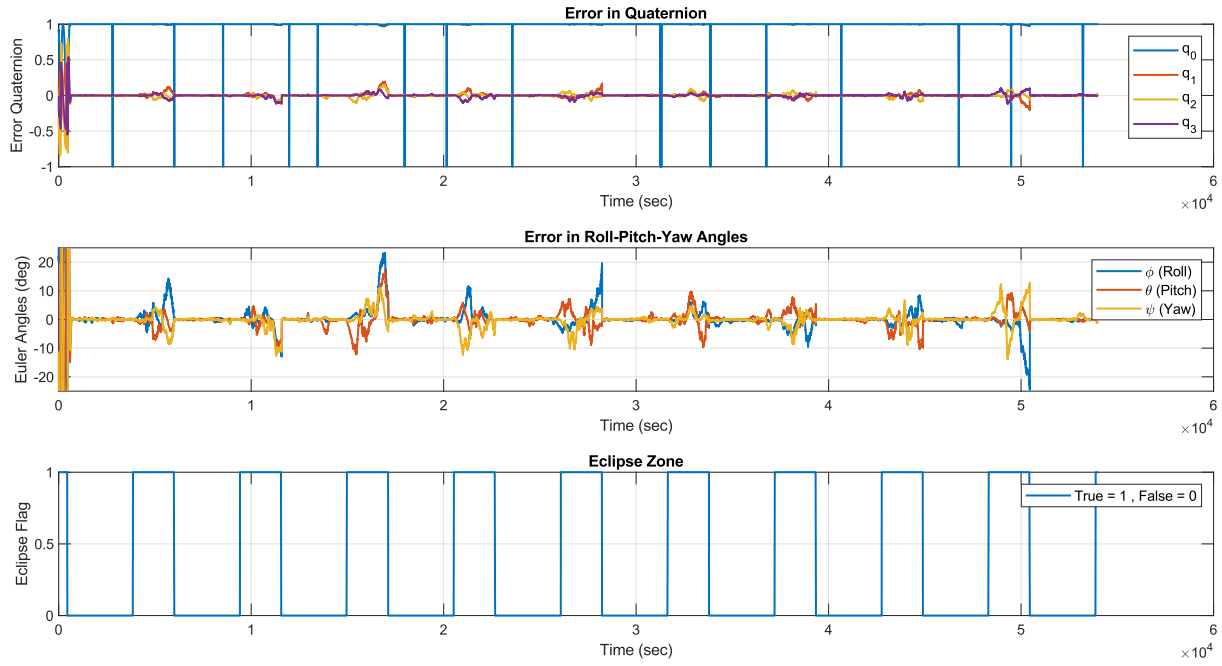


Figure 7.14: Kinematics based MEKF using MEMS Gyroscope plots displaying the error quaternion, error in term of Euler angles, and eclipse flag.

7.4.1 System Analysis: Kinematics only based MEKF

The system analysis conducted for the dynamic and kinematics based MEKF, is repeated for this system model as well. Because fewer sensors are used in this approach, the effects seen in attitude error upon failures are predicted to be higher than the previous system model. Since there are only 2 different types of sensors used for the MEKF update step, only single sensor failures are analyzed. However, issues such as misalignment, misconfigurations and variation in operational frequency will be analyzed in the same fashion as the previous system model.

Individual Sensor Failure

The first sensor failure investigated is the magnetometer utilized in this system model. This sensor failure is expected to take a toll on accuracy as seen in the previous MEKF system model. However, upon the failure in this sensor the kinematics based MEKF only updates itself by using

CSS and FSS. Thus, only the system is entirely dependent on FOG during the propagation when the CSS and FSS aren't available. Moreover, when not in eclipse the inaccuracy is still present as the state correction is not consistent as the CSS and FSS updates are varied based on their field of view of the sun. The error plots displayed in Figure 7.15 show an uncontrollable error value throughout the 10-orbit timespan. The RMS error in this state in terms of Euler angles are 86.0705 deg, 32.7957 deg and 70.7044 deg. The error values decrease when compared to the previous system model, the use of gyroscope in the propagation state aid the filter to correct the states. As the system model begins to correct the states under normal orbital conditions, it appears to worsen rapidly when the satellite is in eclipse. It is worth noting that this is sensor is embedded in the ESSENCE CubeSat's OBC making it less likely to fail when in use. This is because multiple points of failure would need to occur as it is integrated in another local system. The recommendation presented in the previous system model can be applied here. The use of redundant sensor such as an magnetometer on an integrated circuit, would help mitigate this form of single point failure.

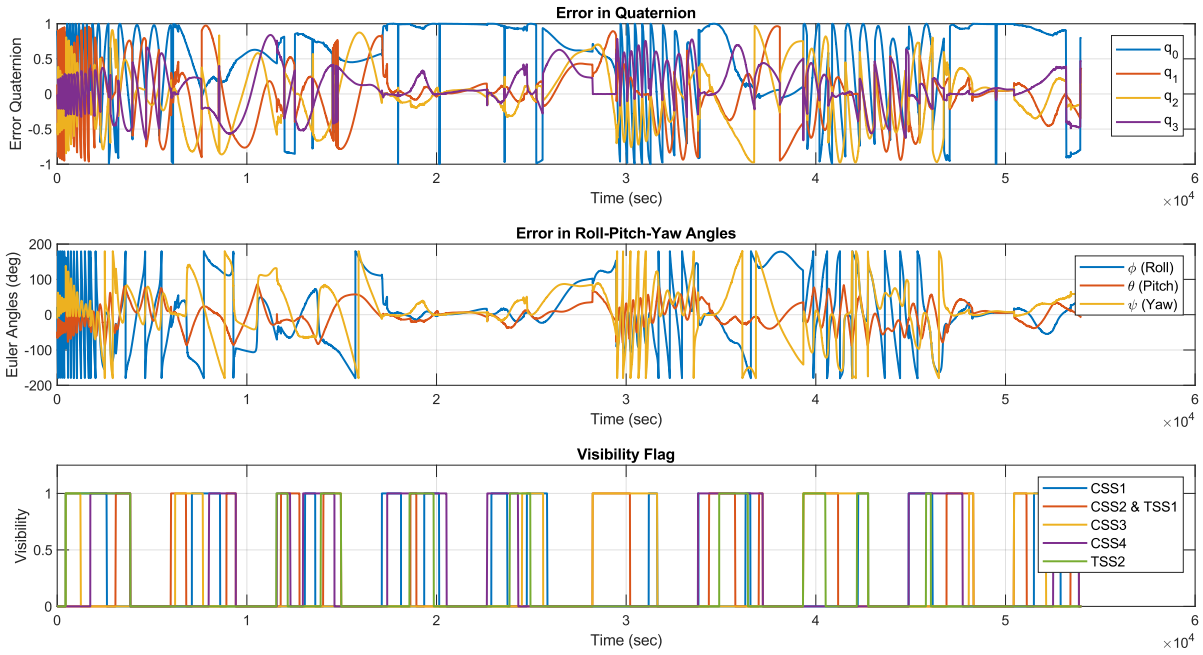


Figure 7.15: Shows plots displaying the error in quaternion, Euler angles and Visibility flag for magnetometer failure in the kinematics based MEKF.

In addition to the magnetometer failure, an analysis is also made where a failure in the sun sensor is experienced. Although these are two different types of sun sensors, and the test can be multiple sensors failure, they are grouped as one for this test. This is because both CSS and FSS produce angular measurements which are manipulated to form a sun position vector. Figure 7.16 shows the error plots which show that this system model can produce controllable error values. The mean error values found are similar to the ones seen in the FOG failure seen in the dynamics and kinematics-based system model. The RMS error in Roll, Pitch and Yaw are found to be 13.6897 deg, 7.3325 deg and 11.8845 deg. The main error source experienced in this test is once again the bias in the system. This source affects the propagation of the quaternion state, causing the inaccuracy to grow over time. However, the correction in this state is handled well as the error in the eclipse zones are not beyond ± 20 deg. Thus, the sensor detrimental to this system model is the

magnetometer, which confirms the notion concluded when a failure in magnetometer is experienced. A recommendation for the current situation would be to use the current values of each solar panel. These current values would provide insight as to where the sun is illuminating a particular panel. This would lead to providing CSS measurements which can be used in case the photodiode that form the CSS fails.

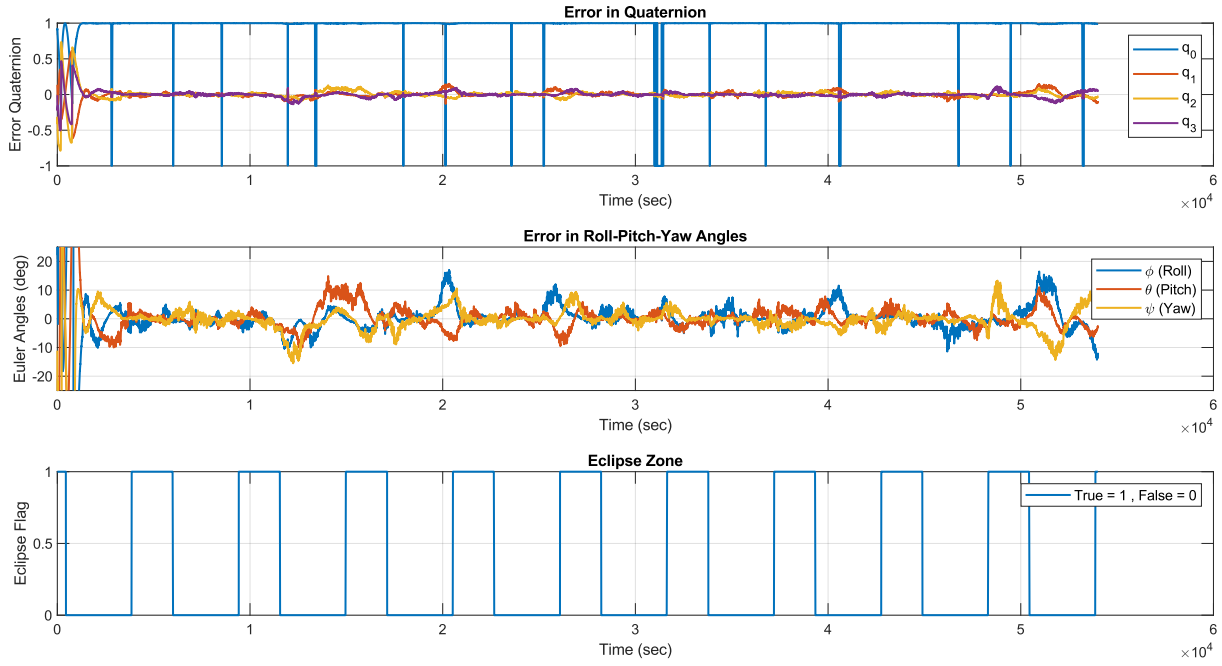


Figure 7.16: Shows the error in quaternion, Euler angles and eclipse flag for sun sensor failure in the kinematics based MEKF.

Sensor Misalignment and Misconfiguration

The misalignment test conducted in this system model follows the frame deviation angles described in Table 7.4. The error plots displaying the effects due to this form of failure are shown in Figure 7.17. As observed in these plots, the error increased when compared to the nominal mode of the kinematics-based system model. The mean error in Roll, Pitch and Yaw are 9.6182 deg, 5.2969 deg and 11.4376 deg. These values also shown an increase when compared to the same

experiment done with the dynamics and kinematics-based system model. However, the gyroscopes in the dynamics and kinematics-based system model were incorporated in the update step, where as in this model it is part of the propagation step. This entails the propagation of the quaternion state is being done in a local sensor frame rather than the body fixed frame. However, the error values are still corrected in an accurate manner as other attitude-based sensors such as the sun sensors and magnetometers are able to account for this misalignment issue. In the eclipse zone, the error shows an expected growth in that phase of the orbit. This is because only the magnetometer is used to update the quaternion state in that timespan. Hence, the increase in the mentioned mean error values is caused as the satellite enters the eclipse zone. Although the deviation is evident it still does not go beyond ±20 deg which matches the value found in the nominal case of the dynamics and kinematics-based system model. Therefore, it is important to avoid this single point failure by carefully placing the Fibre Optic gyroscope in the ADCS. The recommendation in the current scenario is to rely on the MEMS gyroscope. This similar to recommendation presented in the previous scenario. However, the effects are a bit severe, as the gyroscope dictates the propagation step. As shown in section 7.4 the MEMS gyroscope still produces similar RMS error as the nominal case of the previous system model. Therefore, proving that using this redundant sensor is acceptable.

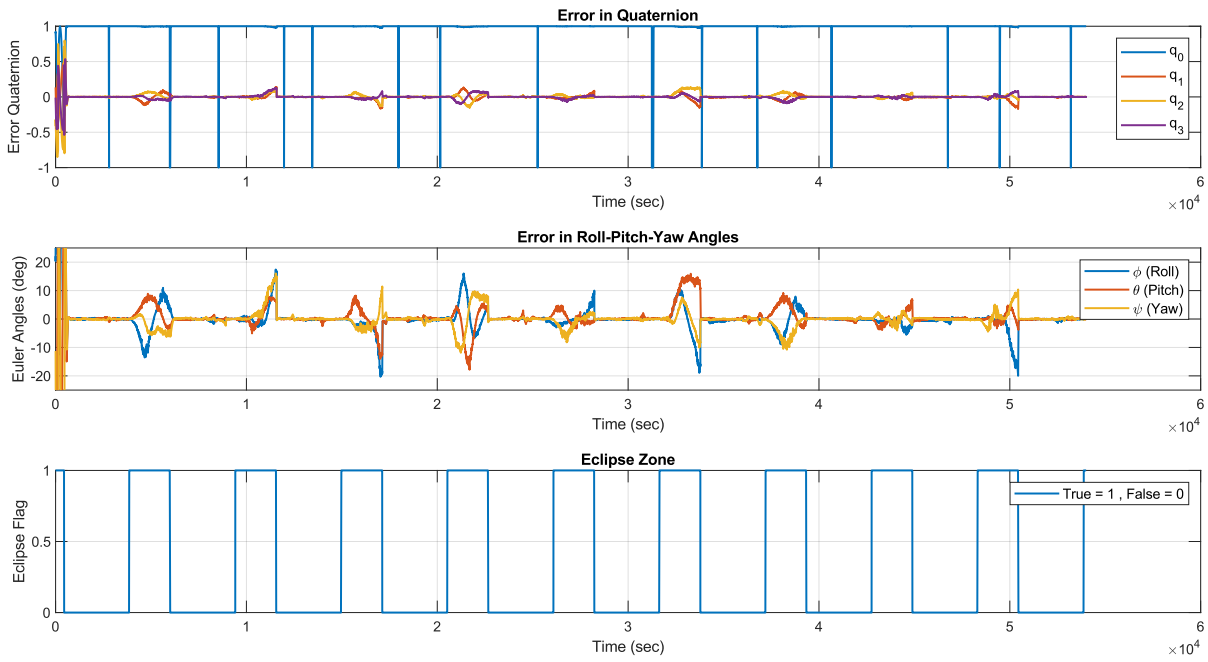


Figure 7.17: Shows the error plots when FOG sensors are misaligned for the kinematics-based system model.

The misconfiguration in the CSS is replicated where all the +X face CSS are programmed to be in an incorrect +Z direction. The effects caused by this error have a larger impact than the first system model as the CSS suite is reduced to only 3 faces. This adds to the drawbacks found for the state update step caused by the lack of sensors. Figure 7.18 shows the error plots for this situation and displays the growth in attitude error throughout the entire simulation. The mean error in terms of Roll, Pitch and Yaw are 58.1693 deg, 27.3400 deg and 56.9789 deg. The error in Roll and Yaw axis is high as the updates keep correcting for the +Z face whenever the sun illuminates the +X face of the satellite. In the previous system model, the sensors were strictly used for state correction and were able to mitigate this issue. Therefore, the misconfiguration of the CSS takes a heavy toll as error values when using the kinematics based MEKF. A recommendation for this situation would be to perform an offline analysis. Measurements would be downlinked and examined to

determine the faults. Upon the conclusion of which CSS output overlaps, this failure can be corrected by up-linking a correction matrix. This would lead to a more accurate attitude estimate using this mitigation technique.

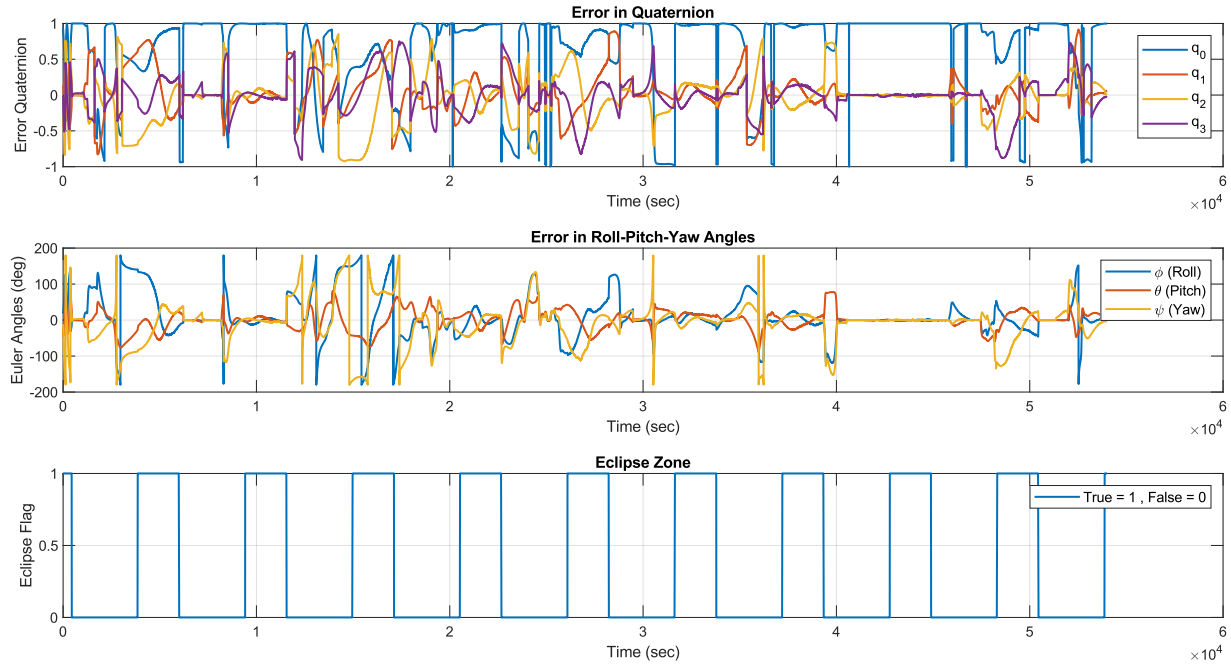


Figure 7.18: Shows the error plots when CSS sensors are misconfigured for the kinematics-based system model.

Various Update Frequencies

The final test was done to compare the results for low and high operational frequencies which are defined at 1 Hz and 100 Hz respectively. The results are listed in Table 7.9 where the comparison in Euler angles are displayed. It is observed that the error slowly grows as the operational frequency are increased and improves if it is decreased. This is because this filter is very sensor dependent, as the input for state propagation is reliant on the accuracy of the Fibre Optic gyroscope. Moreover, the faster the operational frequency the more often bias increases the inaccuracy of the system model. Hence, the slower operational frequency produces lower error values. The error

difference in the lowest operational frequency and in the nominal mode display the values don't vary in large amounts. This hints that the operating at 10 Hz is feasible much like the test conducted for the previous system model. Moreover, this also meets the mentioned ESSENCE mission requirement. When Tables 7.6 and 7.9 are compared, the values in nominal mode for both system model don't show a noteworthy difference. However, the error plots in Figure 7.12 validate the kinematics-based system model is preferred, because the error when the satellite is in eclipse inaccuracies are significantly lower. In addition, the kinematics-based model is more computationally efficient. The execution time for the dynamics and kinematics-based system model is 8990.1 seconds. However, the kinematics-based system model has an execution time of 4385.9 seconds. Thus, making it 52% more computationally efficient than the previous system model. Therefore, the kinematics-based MEKF is selected as the primary attitude determination algorithm for the ESSENCE CubeSat mission.

Table 7.9: Shows the error values in the Euler angles for all three operational frequencies.

	Error in Euler Angles		
Operational Frequencies	**Roll (ϕ)**	**Pitch (θ)**	**Yaw (ψ)**
1 Hz	9.5565 deg	4.3359 deg	10.7800 deg
10 Hz (Nominal)	9.5556 deg	4.6762 deg	11.1560 deg
100 Hz	12.6834 deg	5.9892 deg	14.8582 deg

CHAPTER 8 HARDWARE IN THE LOOP EXPERIMENT

In this chapter, the both system models for the MEKF in Chapter 7 is applied in a hardware-based experiment. An inertial measurement unit (IMU) embedded in a cell phone is utilized for this experiment. In addition, various contents such as the derivation of local reference frames and calibration process required to conduct this experiment is discussed in a thorough manner. Lastly, a comparison of both system models used in the MEKF is done to prove the notion stated in a simulated environment.

8.1 Experiment: Introduction and Setup

This hardware in the loop experiment is focused on producing attitude profiles using an IMU found in cellphones. The applicability of this HIL experiment demonstrates the ability of the MEKF to adapt to any system to determine attitude. Upon the success of this experiment, it would provide assurance that the use of MEKF of onboard attitude determination is an acceptable solution. The online attitude estimation method is not restricted to just satellites; hence it is acceptable to be used for this experiment. The device used for the HIL analysis, does not experience the constraints found for a LEO satellite. However, the process to determine the orientation of the cell phone is the same as the one defined in Table 7.1.

The IMU in an android phone was used which consisted of a gyroscope, magnetometer, and accelerometer. The accelerometer is often ignored because of the sensor's high sensitivity nature. It measures the smallest of amount of acceleration experienced in the system. Due to the lack of sensors found on cellphones, both system models discussed in this chapter make use of the entire this tri-sensory suite. Table 8.1 shows the error values associated with each of the mentioned

sensors. Although the environmental effects in LEO such as the disturbance torques are not present, hardware-based noise sources are present. The outputs obtained from these sensors go through an integrated low pass filter. This benefits such a filter, as it smoothens out any high frequency noise. However, this entails low frequency noise parameters such as bias are still present and become more prominent while determining the orientation of the system. Unlike the simulated tests, the output frequency of the MEMS sensors were not scaled down, as this was a real-time HIL experiment. Hence, further proving that the success of this experiment would confirm the adaptation of the sensor fusion approach described in Chapter 7.

Table 8.1: Shows the error values of the sensors found in the android phone's IMU [99] [100].

IMU Sensor	Parameter	Value
STMicro LSM6DSM (Gyroscope)	RMS error	75×10^{-3} deg/sec
	Bias Instability	15 deg/hr
Memsic MMC5603NJ (Magnetometer)	RMS error	0.2 μT
STMicro LSM6DSM (Accelerometer)	RMS error	0.0176 m/sec^2

The objectives defined for this experiment as stated as follows:

Experiment Objectives:

1. Prove the robustness of the MEKF attitude estimation algorithm

2. Enhance the ZOH technique to incorporate IMU sensors
3. Analyze the two system models used in the MEKF by comparing the error in Euler angles

This experiment was conducted in an uncontrolled testing environment due to the inability to access the laboratory during the thesis timespan. Ideally the experiment should've been conducted in a clean room setting where external noise sources would've been minimized or nearly eliminated. Because of the mentioned circumstances, the manipulation in the cell phone's orientation is not done autonomously either. The movements were done by human interaction the analysis was done of both system models used to form the MEKF. The interactions between the phone and the computer to extract sensor outputs is done by a serial connection. Figure 8.1 shows the setup implemented for this HIL experiment. Moreover, the components used to conduct this experiment are labelled in Figure 8.1.

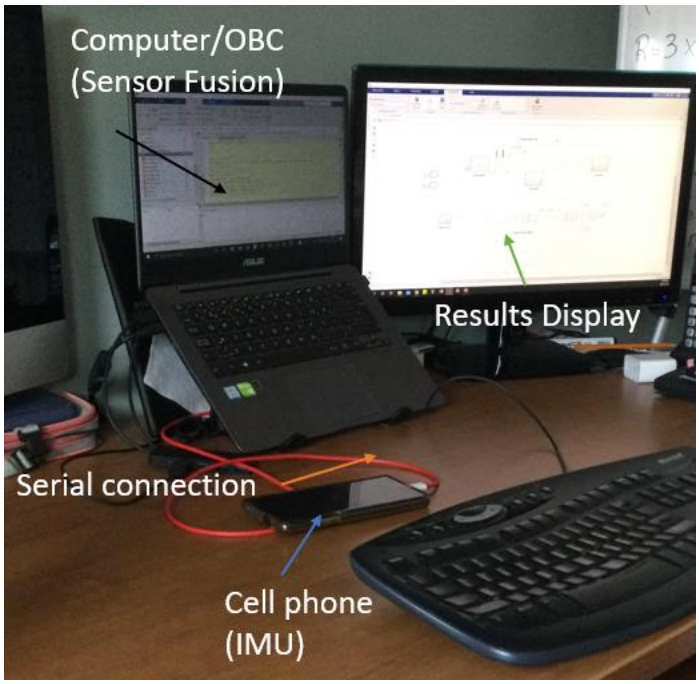


Figure 8.1: Hardware in the Loop experimental setup.

Three key assumptions were made for this experiment and are listed as follows:

1. IMU is at the geometric center of the phone
2. Center of Mass is at the center of the phone
3. No intentional translational motion occurs when conducting rotational movements

Assumptions related to the IMU placement, and the center of mass impacts both forms of MEKFs. This is because both implement the MEMS sensors that would measure about the axis exclusive to the IMU. Hence, causing an additional error when interpreting the measurements. However, these error values are bearable as the cellphone isn't very big to cause noteworthy issues. Moreover, impacts related to unintentional occurrence of translational motion is unavoidable. However, attempts were made to mitigate this by making rotational movements about a pivot point for each axis. Due, to the limitations in the cell phone's documentation, and the experimental environment, the mentioned three assumptions were made for this experiment.

In Chapter 7, it was found that the kinematics-based system model used for the MEKF is the better system model for the MEKF algorithm. Therefore, the following hypothesis is made for the HIL experiment:

The kinematics-based system model produces controllable error values when compared to the dynamics and kinematics-based system model.

8.2 Reference Frames and Vector Definition for HIL Experiment

To begin the HIL experiment, it is required to define two coordinate reference frames to describe the orientation of the cell phone. For satellites, the inertial frame of reference is defined as the J2000 ECI frame. However, the cell phone is not rotating the earth and so the J2000 ECI frame can't be used. Thus, a local tangent reference frame is formed. In addition, reference vectors for sensors such as magnetometer and accelerometer are required been defined in this local tangent

frame as well. This is simply because the outputs of those sensors are in terms of vectors, and a direct attitude quantity such as angles or angular rates.

8.2.1 Reference Frame Definition

The derivation begins with realizing ECEF frame as the base frame. This frame is fixed to the planet and rotates about Earth's rotation axis at the same rate of Earth's rotational rate. Many local frames utilize the ECEF frame to build various local frames. One of such frames is the previously mentioned NED frame. This frame is exclusive to 3 quantities longitude, latitude, and altitude. Every point on Earth has its own NED frame because the geodetic quantities vary with location. Table 8.2 shows the parameter used to form the local NED frame used for this experiment and is visualized in Figure 8.2. The sphere shown on the globe is exaggerated to illustrate it is a topocentric coordinate system and is used for various orientation centric solutions.

Table 8.2: Geodetic parameters used for HIL experiment.

Parameters	Values
Latitude (λ)	43.77735^{o} N
Longitude ($_{\varphi}$)	-79.5019^{o} E
Altitude (Mean Sea Level)	136.0 m

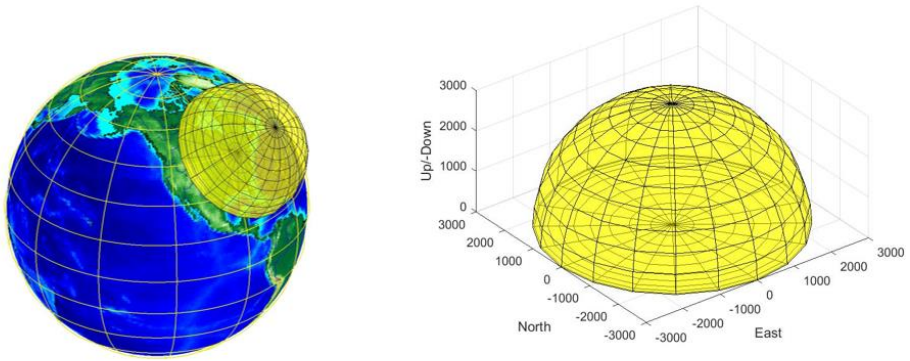


Figure 8.2: Displays the NED shown tangent to Earth and the topocentric view of the frame.

Therefore, using these concepts the local tangent frame and the cell phone's body fixed frame is defined in Figures 8.3 and 8.4. The local tangent frame is defined using Table 8.2 and is as follows:

- I_X (North): Vector Coincident to the Local North based on the parameters shown in Table 8.2
- I_Y (East): Vector perpendicular to the Local North
- I_Z (Down): Vector pointing towards center of the Earth, and completes the TRIAD

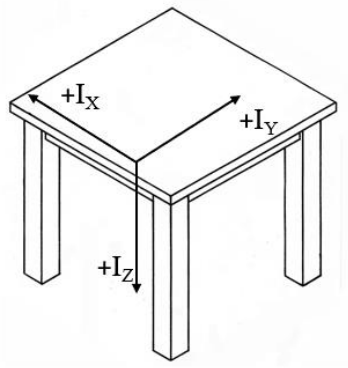


Figure 8.3: Illustration of the local tangent frame used in the HIL experiment.

The IMU used for the experiment is assumed to be placed in the center of the android device. Thus, the body fixed frame definition is described as follows:

- B_X: Completes the TRIAD
- B_Y: Vector along the length of the cell phone
- B_Z: Vector perpendicular to the screen of the cell phone

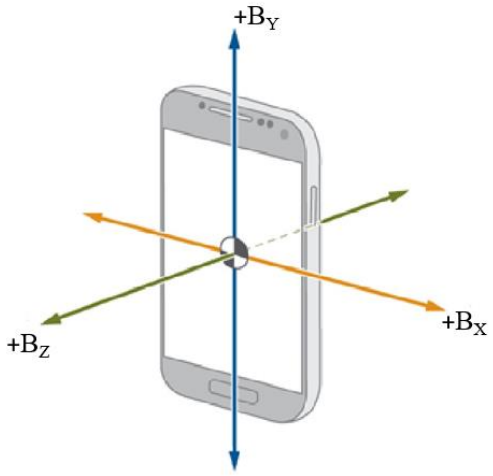


Figure 8.4: Illustration of the cell phone's body fixed frame used in the HIL experiment.

8.2.2 Local Tangent Frame Magnetic field and Acceleration Vectors

Magnetic Field Vector

The derivation of the magnetic field vector process employed in the simulated tests is reused but requires further manipulation. The IGRF-13 model mentioned in Chapter 5 produces the quantities in the ECEF. However, for this experiment it is required to transform it in terms of the NED frame. Using the rotational matrix shown in (8.1) [101], the magnetic field vector expressed in ECEF is expressed in the NED frame using the geodetic parameters in Table 8.2.

$$R_{ECEF}^{NED} = \begin{bmatrix} -\sin(\lambda)\cos(\varphi) & -\sin(\lambda)\sin(\varphi) & \cos(\lambda) \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ -\cos(\lambda)\cos(\varphi) & -\cos(\lambda)\sin(\varphi) & -\sin(\lambda) \end{bmatrix} \tag{8.1}$$

where

- λ: Represents the geodetic latitude shown in Table 8.2
- $_{\varphi}$: Represents the geodetic longitude shown in Table 8.2

The Earth's magnetic field remains constant because the phone is held at the same geographic location. In addition to the IGRF-13 model, the World Magnetic Model can also be used to achieve the same result [102] [103]. However, both IGRF-13 model and WMM produce similar results as shown in Figure 8.5 and Table 8.3. The values remain constant for the duration of this experiment but slightly differ from each other. This is because the Gaussian coefficients used in the WMM model is not the same as the IGRF-13 model. Thus, even if they used the same equation to produce the magnetic field vector, the Gaussian coefficients create this minor difference. Table 8.3 shows the difference between the models, and the difference is negligible. The IGRF-13 was selected for the HIL experiment to remain consistent with the tests conducted in the simulated environment in Chapter 7.

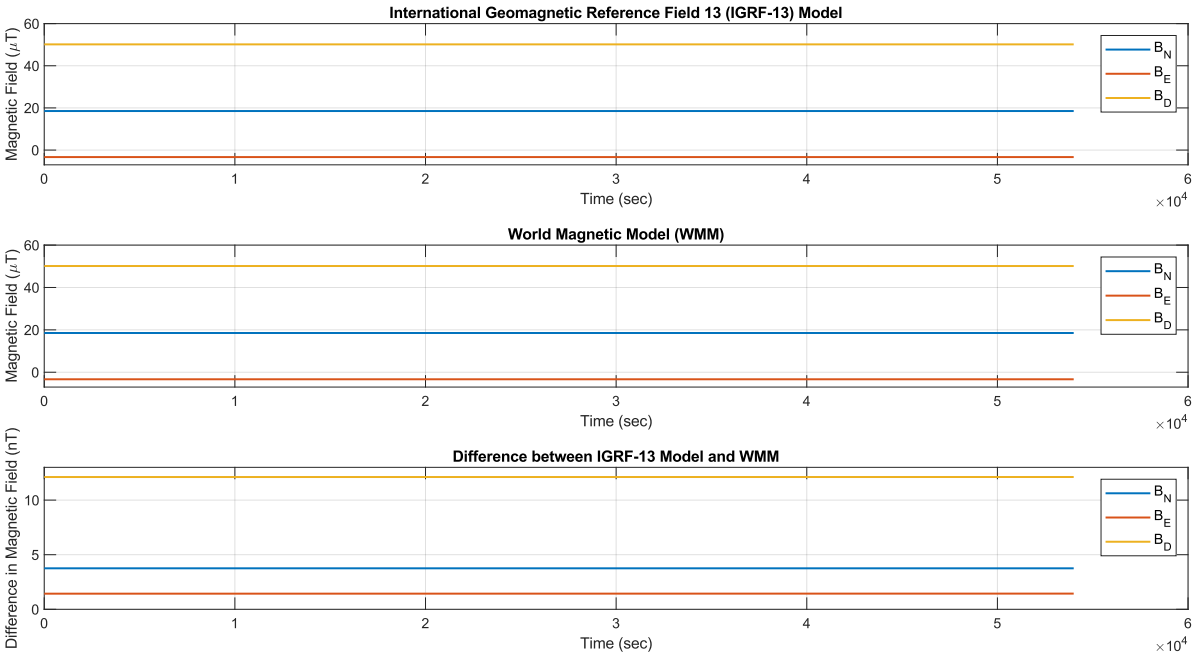


Figure 8.5: Shows the comparison between the IGRF-13 model and WMM, with the difference between the 2 models.

Table 8.3: Show the magnetic field results of the IGRF-13 Model, WMM and the difference between the two models.

Magnetic Field Model	**North**	**East**	**Down**
IGRF-13	18.520 μT	-3.325 μT	50.140 μT
WMM	18.517 μT	-3.326 μT	50.128 μT
Difference	3.753×10^{-3} μT	1.432×10^{-3} μT	12.113×10^{-3} μT

Acceleration Vector

In addition to the magnetic field reference vector, it is required to derive an acceleration vector in NED frame. The accelerometers measure any acceleration that is experienced by the system. However, this is experiment considers rotational motion only. Thus, an assumption is made where no linear accelerations take place. When the motion is restricted to the rotational motion, the only natural force is an attraction force. Hence, only force of gravity is the force acting on the system. Using the mentioned assumption and the movement restrictions, the derivation of the acceleration vector is shown in (8.2). When the system is in motion, it is though as it is free falling. Thus, using the Newton's second law, the normal force is in the negative down direction, and the force of gravity is in the positive. Thus, leading to an acceleration vector shown in (8.3) expressed in the NED frame and is confirmed by [102] [103].

$$\begin{aligned} F_{net} &= 0 = F_N + F_G \\ ma &= mg \\ a &= g \end{aligned} \tag{8.2}$$

$$a = \begin{bmatrix} 0 & 0 & 9.81 \end{bmatrix} \text{ m/sec}^2 \tag{8.3}$$

8.3 Sensor Calibration

As shown in Figure 8.1, the HIL experiment is not conducted in a controlled environment. Thus, it is important to consider the noise sources which can be accounted for prior to initiating the MEKF process. These sources affect the magnetometer the most because such MEMS devices are highly sensitivity in terms of measurements. It not only picks up the Earth's magnetic field but also recognizes any local magnetic field in the vicinity. Local magnetic fields are a product of anything that either produces its own magnetic field or is attracted to any magnetic source. Thus, factors which cause these calibration issues are group in terms of Hard Iron and Soft Iron effects.

Disturbances caused by Hard Iron effects are from anything that generates an electromagnetic wave. Examples that cause the hard iron effects are computers, power outlets, active wires, etc. The Soft Iron effects take place due to any exposed material and gets attracted to the existing magnetic fields. This can be generalized by assuming it is any material that ferromagnetic properties. Examples of such objects include screws, pins, coins, etc. This effect is harder to compensate when compared to the Hard Iron effects, as the possibilities are countless.

The mentioned issues are a product of the experimental environment and are compensated through a sensor calibration technique. This simple technique accounts for both Soft and Hard Iron effects and treats them as offsets to the true magnetic field. MATLAB has an excellent function called "magcal" [104] for such a task. Prior to using this function, the cell phone is rotated in all directions to gather the measured magnetic field vector values at various points while in motion. This allows the user to see how the calibration effects cause a deviation between the Earth's magnetic field and the summation of the local and Earth's magnetic fields. Thus, the function takes the measured magnetometer outputs to produce a Hard Iron Calibration bias (b) and a Soft Iron Distortion Matrix (A). These calibration parameters are shown in Table 8.4. Using these values, the calibration of the magnetometer is conducted using (8.4) [104]. Furthermore, the variation in the calibrated and uncalibrated magnetic field vectors are shown in Figure 8.6. By observing this figure, it is a testament to how the Soft and Hard Iron effects cause incorrect magnetometer readings.

Table 8.4: Shows the Hard Iron bias and the Soft Iron Distortion matrix using "*magcal*".

Parameters	Values
Hard Iron Bias (b)	$\begin{bmatrix} -2.2597 & -0.4829 & -1.426 \end{bmatrix} \mu T$
Soft Iron Distortion Matrix (A)	$\begin{bmatrix} 1.0990 & -0.0005 & 0.0358 \\ -0.0005 & 0.9437 & -0.0054 \\ 0.0358 & -0.0054 & 0.9653 \end{bmatrix}$

$$B_{cal} = \left(B_{uncal} - b \right) \times A \tag{8.4}$$

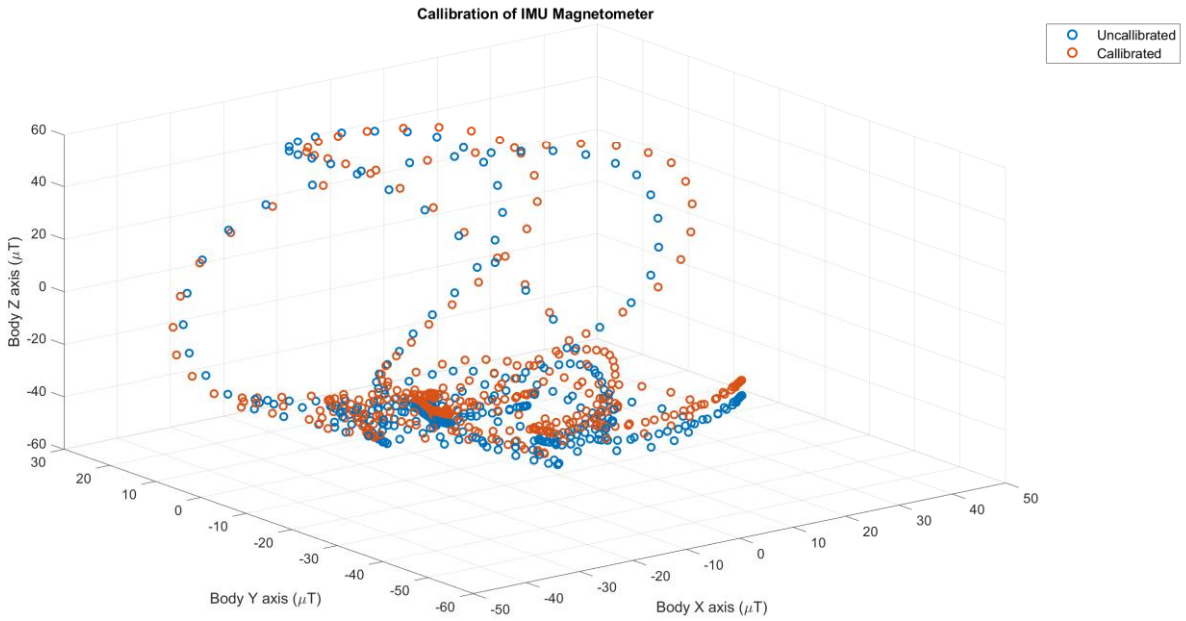


Figure 8.6: 3D plot displaying magnetometer calibration vs uncalibrated values.

In addition to the calibration issues, the definition of Earth's magnetic and rotational poles affect the measured quantities as well. The magnetic poles are approximately 11.5 deg from the Earth's

rotational axis, but it differs from which hemisphere the experiment is performed. The angle which differentiates the two poles is called the declination angle. For the location mentioned in Table 8.1, the declination is 10^{o} 11' W. The magnetometer in an IMU functions can be perceived as a compass. Thus, when the phone points the "North", it always points towards the magnetic north, rather than the true north. This deviation of the pole results in a continuous error in both system model as the NED frame references and is built assuming the north direction is known.

8.4 MEKF Results and Analysis

To meet the objectives defined for the HIL experiment variation in both system model of the MEKF are explored and analyzed. Simulink is used for this experiment as it has support packages for androids. Hence, it is incorporated to extract IMU sensor outputs real time. Figure 8.7 show the Simulink blocks that correspond to the 3 IMU sensors.

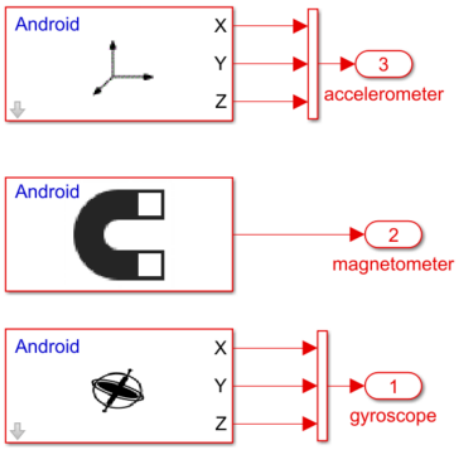


Figure 8.7: The sensor extraction Simulink blocks for accelerometer, magnetometer, and gyroscope in the cell phone's body fixed frame.

Because this is not a simulated experiment, it is necessary to define a ground truth model. A

common estimation approach called the attitude heading reference system (AHRS) is employed for this task. Various applications such as airplanes, UAV, rover, etc. make use of AHRS. This orientation estimator is prebuilt and configured in MATLAB/Simulink. The IMU sensors are inputted in this filter and it outputs quaternions using the sensors directly. Moreover, the frames used match the defined frames used in the HIL experiment. Figure 8.8 show the schematic of a general AHRS model. As seen the outputs form the three sensors are fed to the AHRS. The output of the ground truth model are then expressed in Euler angles and are later transformed to quaternions for the state comparison.

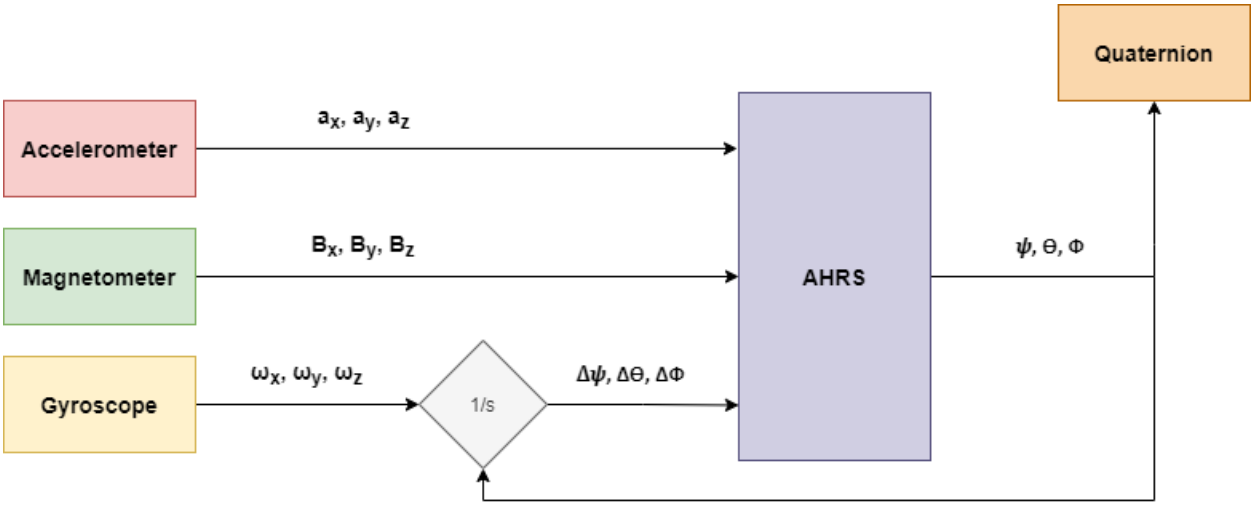


Figure 8.8: Schematics used for the AHRS attitude estimator in the HIL experiment.

8.4.1 Dynamics and Kinematics System Model: Results and Analysis

The system model is initialized by defining MEKF quantities such as states vector and covariance matrix. A few changes are made where quantities such as the process noise matrix (Q) and the moment of inertia matrix (J) are recalculated. The android device used for the experiment was a OnePlus 7 Pro. Table 8.5 shows the dimensions, mass, and the moment of inertia matrix for this system.

Table 8.5: Android device dimensions, mass, and moment of inertia matrix.

Parameters	Values
Dimension (L x W x H)	75 mm x160 mm x8.8 mm
Mass (m)	206 g
Moment of Inertia Matrix (J)	$\begin{bmatrix} 0.441 & 0 & 0 \\ 0 & 0.1002 & 0 \\ 0 & 0 & 0.5417 \end{bmatrix} \times 10^{-3}\ \text{kgm}^2$

The process noise matrix is reconfigured to match the needs of this system. The process noise relevant to the quaternion and angular velocity state vectors is found through trial and error. The rotational motion was done through a physical interaction with the phone; hence the process noise was not easy to model. However, the initialization of the state and covariance quantities don't require any changes. Table 8.6 shows the initial parameters and the process noise matrix defined for this system model. In addition, Figure 8.9 shows the implementation in Simulink which shows the MEKF, AHRS and the sensor inputs using the ZOH technique.

Table 8.6: Initial state vectors, covariance and the process noise matrix for the Dynamics and Kinematics based MEKF.

Parameters	Values
Initial Quaternion Estimate (q_{est})	$\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$
Initial Angular Velocity Estimate (ω_{est})	$\begin{bmatrix}0.1 & 0.1 & 0.1\end{bmatrix}$ rad/sec
Initial Bias Estimate (β_{est})	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}$ rad/sec
Initial Covariance (P_{est})	$\begin{bmatrix} 3\left(\frac{\pi}{4}\right)^2 I_3 & 0_3 & 0_3 \\ 0_3 & \left(1\times10^{-3}\right)^2 I_3 & 0_3 \\ 0_3 & 0_3 & \left(\frac{15\pi}{3600\times180}\right)^2 I_3 \end{bmatrix}$
Filter Timestep (Δt) , Operational Frequency (Fs)	0.1 sec , 10 Hz
Process noise matrix (Q)	$\begin{bmatrix}\sigma_d^2 I_3 & 0_3 \\ 0_3 & \sigma_u^2 I_3\end{bmatrix}$ $\sigma_d \approx 5x10^{-3}, \sigma_u \approx 4.4x10^{-3}$

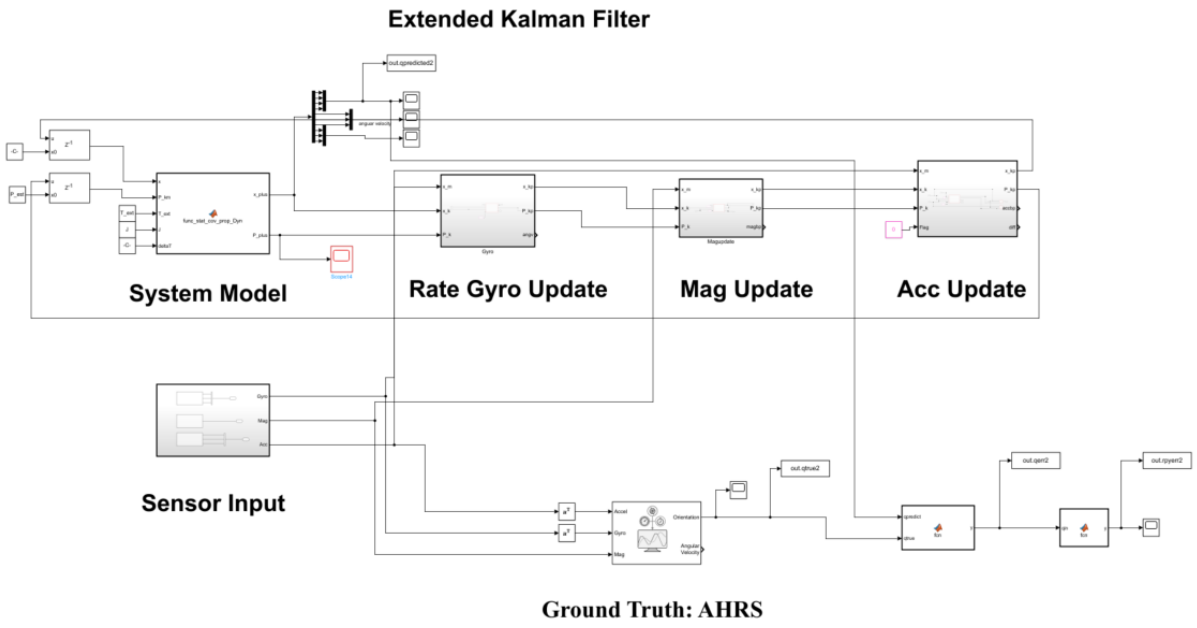


Figure 8.9: Simulink model of the Dynamics and Kinematics based MEKF used for the HIL experiment.

The HIL experiment was ran for ~5 mins, where the cellphone is rotated approximately ±90 deg in the Yaw, ±45 deg in Roll, and ±45 deg in Pitch axes, corresponding to the Z, X and Y axes of the NED frame. The movement is restricted because it is a wired connection between the phone and computer as shown in Figure 8.1. For attitude analysis, only the quaternion state is compared to the AHRS, as that is the desired quantity to describe the phone's orientation. Figure 8.10 shows the quaternion generated from the AHRS and the MEKF. It can be seen the filter mimics the filter relatively well for the duration of the experiment. In Figure 8.11, the plots displaying the error quaternion and the error in terms of Euler angles support this notion. The convergence time for the Dynamics and Kinematics based MEKF is ~ 42.3 seconds, as that is the time for the vector components of the error quaternion coverage to a near 0 values. This symbolizes that the predicted quaternion gets closer to the ground truth as time progresses. As seen the filter demonstrates large

error values specifically in the Yaw axis. The RMS error in terms of Roll, Pitch and Yaw angles are 3.3757 deg, 3.1743 deg and 15.2507 deg. In addition, the error grows rapidly when sudden movements occur. This results in an error growth up to ~11.27 deg, ~10.34 deg, ~27.56 deg in Roll, Pitch and Yaw respectively. These high error values can be explained as it is assumed the IMU is placed at the phone's center of mass. Because this cannot be confirmed, it causes the values to deviate from the ground truth. Another source of error is the declination issue mentioned in the previous section. This shows an additional increase in noise specifically in the Yaw axis. This affect is summated with the effects caused by sudden movements making the growth very prominent.

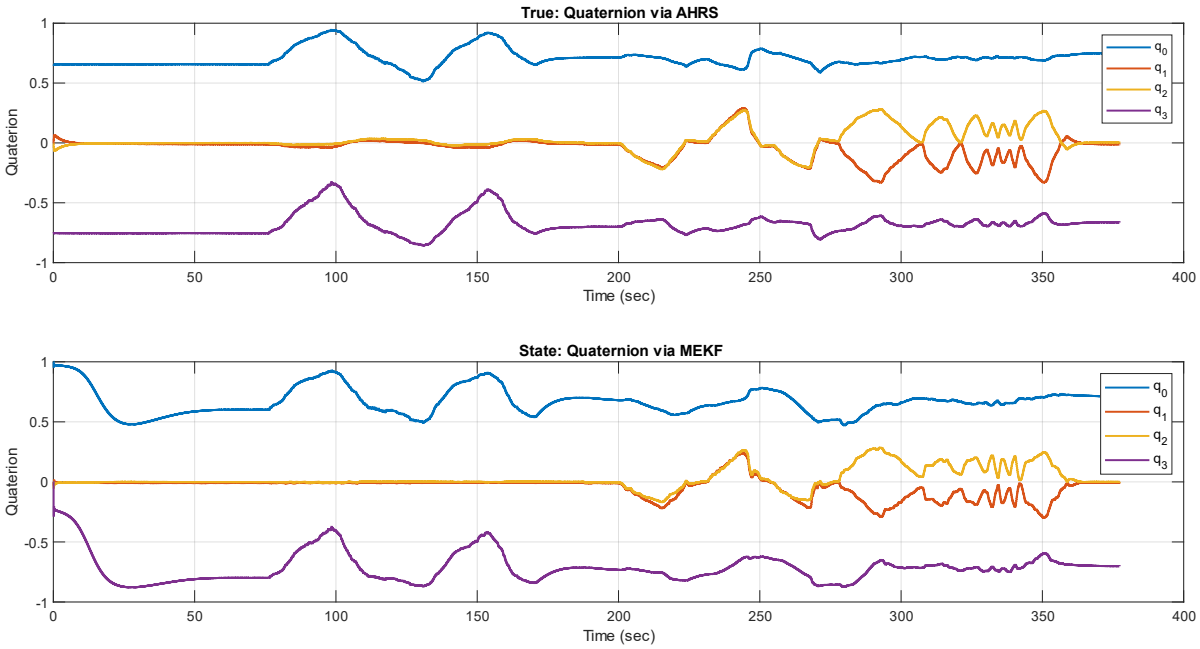


Figure 8.10: HIL experiment true vs predicted quaternion plots using Dynamics and Kinematics based system model.

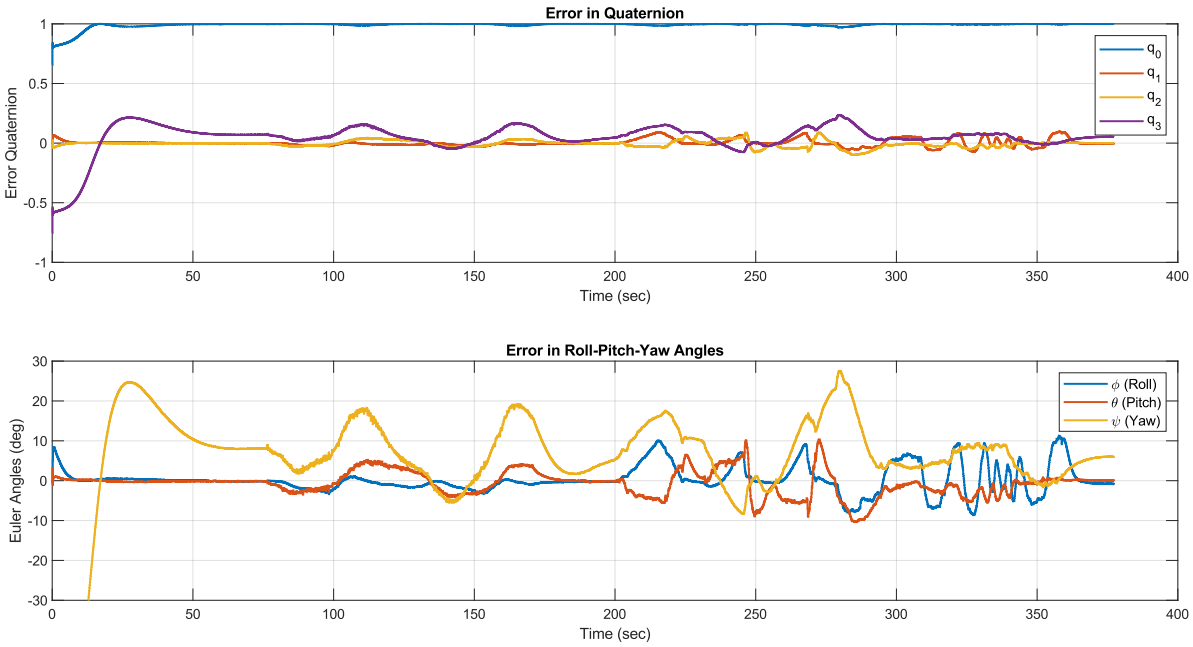


Figure 8.11: HIL experiment plots consisting of error quaternion and error expressed in Euler angles for the Dynamics and Kinematics-based system model.

8.4.2 Kinematics System Model: Results and Analysis

The kinematics-based system model shown in Chapter 7 was concluded to be the better choice for attitude estimation. Thus, for the HIL experiment it was hypothesized that this would be the same results. This model is sensor centric because the input to the propagation step is dependent on the MEMS gyroscope. A low pass filter is present in the cell phone which interacts with the raw measured values. Thus, the bias associated with the gyroscope is very noticeable, as it a low frequency noise source. Table 8.7 Shows the initialized state and covariance values used for the kinematics-based system model. The process noise matrix (Q) remains the same, as there weren't any changes in the experimental settings. Figure 8.12 shows the kinematics based MEKF is implemented in Simulink. A comparison between the true and predicted quaternions is shown in Figure 8.13. The plots communicate a quicker convergence time when compared to the previous system model. The kinematics model leads to a convergence time of 26.7 seconds. This is

approximately 16 seconds quicker than the dynamics and kinematics centric MEKF. Using a sensor dominant system model allows the propagation to be done in an efficient and optimal manner, as it is able to utilize the MEMS gyroscope well. This hints that this system model is able to predict the bias state with an acceptable accuracy value. This is critical as the quaternion kinematics take the bias and subtracts it from the measured angular velocity of the phone, and gradually eliminating low frequency noise found in the system. The RMS error in terms of Roll, Pitch and Yaw are 3.1194 deg, 2.6481 deg and 12.7590 deg. It is clear that there is an improvement in the error values when using the kinematics based MEKF. Moreover, the maximum error values seen in this filter when rotations take place are significantly less than the previous system model. These values are found to be ~11.27 deg ,~7.03 deg ,~20.41 deg in Roll, Pitch and Yaw respectively. This is outlined in Figure 8.14 which display the error plots. It can be observed that the plots are smoother when rotations were made in the same manner as the initial system model. Hence, supporting the notion that the kinematics-based system model is more stable than the dynamics and kinematics.

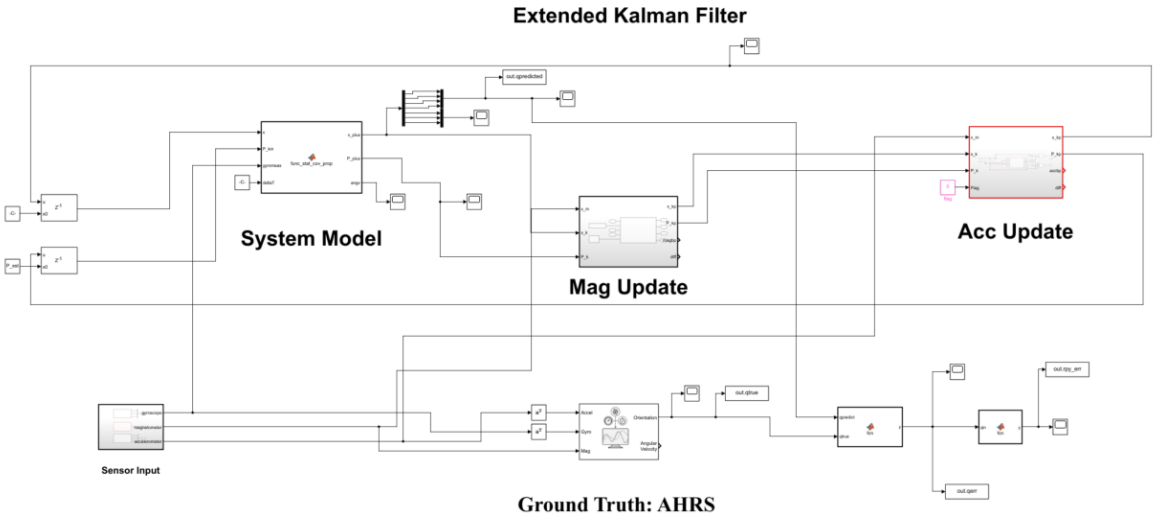


Figure 8.12: Simulink model of the Kinematics based MEKF used for the HIL experiment.

Table 8.7: Initial state vectors, covariance and the process noise matrix for the Dynamics and Kinematics based MEKF.

Parameters	Values
Initial Quaternion Estimate (q_{est})	$\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$
Initial Bias Estimate (β_{est})	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}$rad/sec
Initial Covariance (P_{est})	$\begin{bmatrix} 3\left(\frac{\pi}{4}\right)^2 I_3 & 0_3 & 0_3 \\ 0_3 & \left(1\times10^{-3}\right)^2 I_3 & 0_3 \\ 0_3 & 0_3 & \left(\frac{15\pi}{3600\times180}\right)^2 I_3 \end{bmatrix}$
Filter Timestep (Δt) , Operational Frequency (Fs)	0.1 sec , 10 Hz

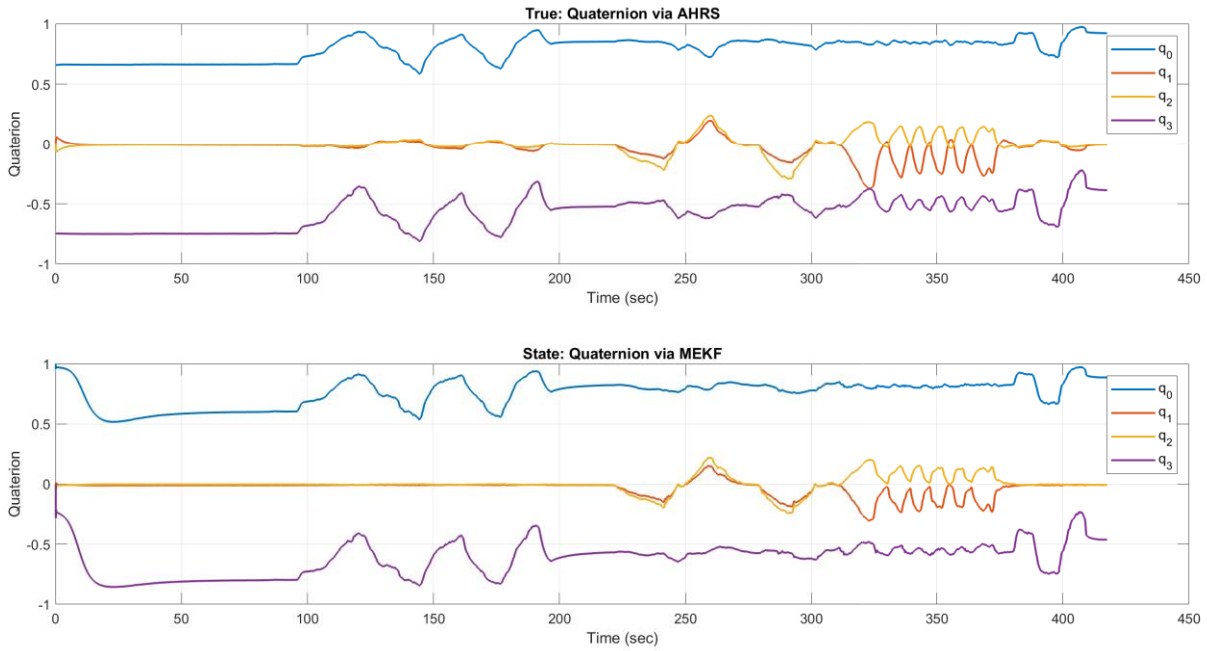


Figure 8.13: HIL experiment true vs predicted quaternion plots using kinematics-based system model.

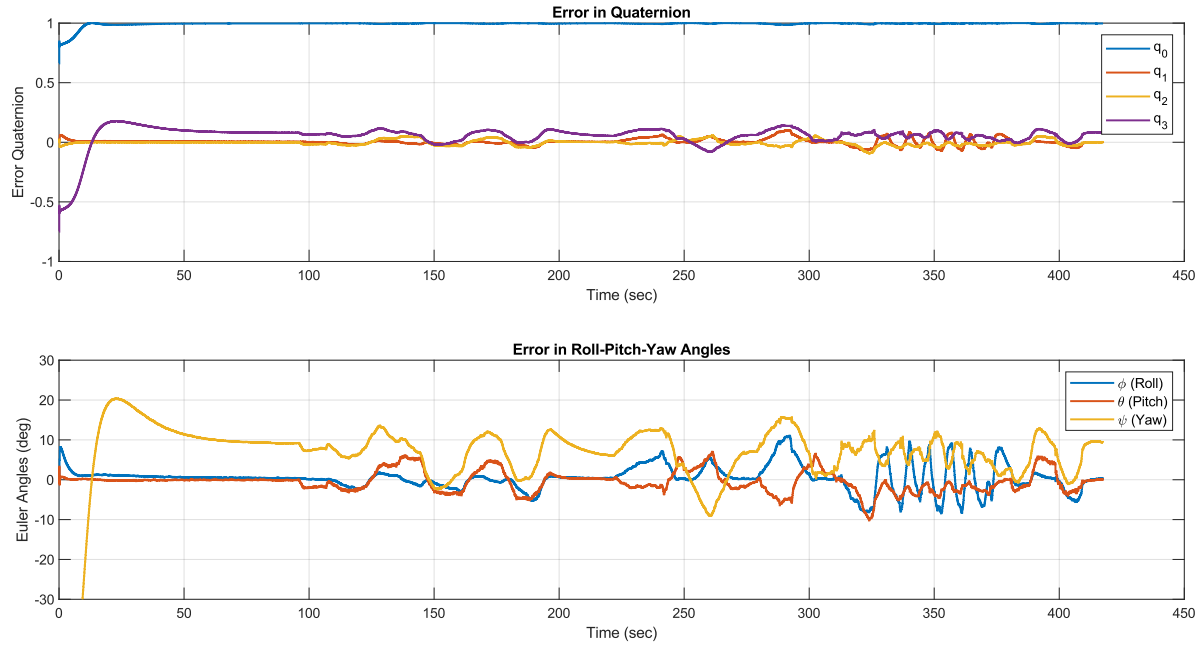


Figure 8.14: HIL experiment plots consisting of error quaternion and error expressed in Euler angles for the kinematics-based system model.

In conclusion, the objectives defined for this HIL experiment were fulfilled. Regardless of the system model used in the HIL experiment, the results show that it is able to adapt to the environment. This is a testament to the robustness of the MEKF and the system models used for orientation estimation. Moreover, such tests provide preliminary analysis to gauge accuracy levels of any satellite's attitude determination system. This is because even if the situations and environments are different the orientation problem remains the same. Secondly, the sensors incorporated for this experiment made use of the ZOH technique to ensure the sensors are extracted at rate defined for nominal conditions. Lasty, a comparison between the two models which concluded the better model used for ADCS. Therefore, the HIL experiment validates the conclusion made through simulated testing that the kinematics-based model, and the hypotheses made for this experiment.

CHAPTER 9 CONCLUSION AND FUTURE WORK

This chapter provides conclusions to demonstrate the objectives defined in this thesis are met, the contribution provided by this thesis and the future work that can be driven off the findings presented through this research.

9.1 Thesis Conclusions

The thesis went over several key concepts used to develop a functional attitude determination system for nanosatellites. Four main entities were developed and analyzed. These are orbit determination, static and online attitude determination, and hardware in the loop experiment. The sensors utilized in the thesis correspond to the ones used on the ESSENCE CubeSat mission. This was because the thesis aimed to aid amateur mission by providing an acceptable attitude determination standard.

9.1.1 Orbit Determination/Propagation Conclusions

The SGP model and the GPS based Kalman Filter were first developed and analyzed. This analysis was critical, as the environmental models which produced sun position and magnetic field vectors were dependent on the results of such techniques. These results included frame transformation matrices and satellite's position and velocity vectors in multiple coordinate frames. It was found that the SGP propagator produced error values that were concerning. This was expected as it's a pure mathematical model in which small errors accumulate and causes a larger effect over time. Hence making this a secondary option for orbit determination. A drastic improvement in error was observed when the GPS receiver based Kalman Filter was developed. This recursive approach enhanced the measured position and velocity vectors in a smooth fashion to produce accurate

position and velocity estimates. When compared to the SGP model, the GPS based Kalman filter required fewer calculations for frame transformation and mathematical relations. This is one of the benefits in using a Kalman Filters, as a reliant sensor such as the GPS receiver significantly improves the predicted values as time progresses. Therefore, making this approach acceptable for operations that are done for long periods of time. In addition, two environmental models were built to produce two useful vector quantities for attitude determination. These quantities are sun position and Earth's magnetic field vector. All the mentioned results were compared with STK and were accepted by the means of comparison. Therefore, the GPS based Kalman filter is concluded to be the superior choice and will be implemented on the ESSENCE CubeSat for orbit determination. Hence, the first objective of this thesis was met, as comparisons were made to provide the optimal orbit determination model for nanosatellites.

9.1.2 Static Attitude Determination Conclusions

Three static attitude determination approaches were developed and analyzed to meet the second objective of the thesis. All three approaches were able to generate attitude estimates using various unique optimization techniques. The commonality between all three static estimators was that they are easy to implement. Hence, making them useful for missions which are not concerned with accurate attitude determination. However, the drawbacks associated them were alarming for mission that have a larger purpose. The Optimized TRIAD method was found to be computationally heavy and had a risk of singularity. The Davenport's Q-method mitigated the singularity issue, but the computational time was the same as the previous method. Lastly, the QUEST method was investigated. This approach was concluded to be the better choice among the three, as it avoided the issues with the previous two methods in an efficient manner. However, a common setback with all three methods still existed. The uncontrollable error values observed for

all static estimators when in eclipse was alarming. Hence, none of the discussed estimation methods are suitable to be the primary attitude determination algorithm. Therefore, the second objective of this thesis was met, as the accuracy levels of all static estimators were analyzed thoroughly.

9.1.3 Online Attitude Determination Conclusions

To complete the thesis objectives three to five, MEKF with two system models were developed and analyzed. The MEKF is a recursive solution which gradually corrects the propagated system models by using attitude sensors. ZOH and FBUA methods were also developed to determine the best approach to handle the asynchronous nature of the mentioned attitude sensors in Chapter 3. Upon comparison, ZOH technique was chosen to be the better choice, as error values were worse in the FBUA method. Next, the two system models were analyzed for MEKF. One made use of the dynamics and kinematics, and the other used only kinematics. Under nominal conditions, it was found that the kinematics-based model had a faster convergence and execution time, and had lower error values when in vulnerable orbit phases such as eclipse. Hence proving the notion indicated in Chapter 2, that the kinematics based MEKF is better suited to be proposed as the attitude determination standard for nanosatellites. In addition, the thesis conducted a system analysis for both filters under various failure conditions to demonstrate how each system model behaves. The dynamics and kinematics-based model was overall performing better, but this is simply because more sensors were used in that system model than the kinematics based MEKF. Hence, the difference was expected to be present if the failures mentioned in the Chapter 7 were to occur. Therefore, this shows the completion of objectives three to five as all the required analysis was conducted and appropriate conclusions are drawn from them in a simulated environment. Moreover, the comparison of all attitude determination algorithms provided a

suitable algorithm that helped answer the research question raised in this thesis.

9.1.4 Hardware in the Loop Experiment Conclusions

To achieve the final thesis objective a hardware in the loop experiment was conducted to validate the results for MEKF system models. An environment was created where the goal was to determine the orientation of a cell phone. This was selected because it had the presence of a tri-sensory unit called an IMU. This main objective was to prove the kinematics based MEKF is the better choice between the two system models and, validate MEKF algorithm's robustness. Once the comparison was done between both system models, the mean error behaved the same way as the simulated environment. However, the variation of error in the dynamics and kinematics model was highly noticeable. The maximum values found in the Pitch and Yaw axis were much larger than the ones seen in the kinematics based system model. Furthermore, the error plots of both system model indicate the deviation from an approximate zero convergence occurs at a lower rate in the kinematics centric MEKF. Thus, the notion presented in the simulated environment was verified, and the final thesis objective was met.

Therefore, all the objectives defined for the thesis gave an answer to the proposed research question. A software algorithm which uses a kinematics based MEKF, is proposed as an attitude determination standard for nanosatellites. The benefits discussed in Chapters 7 and 8 support this solution as it is least likely to contribute to any causes of CubeSat failures discussed in Chapter 1.

9.2 Thesis Contributions

The thesis seeks to provide an appropriate attitude determination standardization for amateur missions. Making the ESSENCE CubeSat mission the center of the analysis across all domains was proven to be beneficial. Several points of failures were exposed regarding every solution

mention for attitude and orbit determination. In doing so, several contributions were made to the scientific community.

The first contribution was directed towards first time satellite developers as the thesis provided a detailed insight on the sensors used on the ESSENCE CubeSat. The most common and low-cost sensors intended for attitude and orbit determination were discussed. This allowed amateur satellite designers understand how attitude sensors behave and their outputs are. There are a lot more sensors such as star trackers and horizon sensors that can be used for CubeSat missions. However, they are lot more expensive and not practical for amateurs due to their complexity.

The second contribution catered towards an insight on methods which provide the means to conduct an onboard orbit determination. Prior to conducting an analysis, the steps are outlined in a detailed manner which can be easily implemented on amateur CubeSat missions. Moreover, analysis conducted between the two approaches gives options to amateurs in case GPS receivers aren't available for their satellite mission.

Third contribution is directed towards a thorough breakdown of two forms of attitude determination methods. The system analysis done in the thesis gave an insight how various types of failure affect the performance of the selected attitude estimator. Hence, providing a guideline to utilize the appropriate sensor model suitable for any amateur CubeSat mission's accuracy requirements.

The final contribution focused on the testing of attitude determination algorithms. There are multiple ways to test an attitude determination algorithm. However, a cost-effective approach that allows amateurs to understand the drawbacks of the estimator is highly crucial. The use of a cell phone's IMU to conduct attitude testing was conducted, and useful conclusions were formulated

that can be useful during early mission development stages. The use of test rigs are very common but are not cost efficient. Although the test was not done in an enclosed setting, it demonstrated the robustness of MEKF based algorithms. Moreover, solidifying the proposed software solution that can be used as an attitude determination standard for nanosatellites.

9.3 Future Work

ADCS is one of the most unique subsystems that can produce multiple means to progress the research in this domain. Once such progression is regarding testing approaches. This can be further analyzed to the performance of the attitude determination algorithm that doesn't hold the same constraints as the HIL experiment in this thesis. The common testing techniques used in the past included Helmholtz cage, test beds and air bearing tables. However, these were primarily meant to test control rather than attitude determination [105]. The use of a simple platform which has the capability of 3 rotational movements would be better suited for the task at hand. As mentioned, the HIL experiment required manipulation of attitude via human interactions. This leads to several problems as the sensors present in the system don't behave appropriately due to this effect. The benefits of various testing methods discussed in [105] are worth exploring as they avoid the mentioned issues in the current HIL experimental setup.

Second, a low-cost sensor was introduced by NASA Ames Research Center that made use of a cell phone's regular camera for position and attitude determination [106]. There were some hints of success seen where cellphone's camera was interpreted as a star tracker, and horizon sensor. However, due to the limitations of phone cameras at the time, it posed significant limitations and issues such as sensor blooming, and low ISO sensitivity values [106]. However, in current phone camera's are high resolution and high ISO cameras that are worth testing as star trackers to further improve attitude estimation. As mentioned, star trackers are one of the most accurate sensors.

Therefore, this approach can hold potential benefits for small missions with heavy cost restrictions.

Attitude determination was the focus of this thesis, but attitude control is equally important. Thus, the standard proposed can be further expanded to generate one for the entire ADCS. This would lead to an ease in implementation of the ADCS for amateur CubeSat developers. The standard proposed in the thesis gave a through insight on the attitude determination process. Therefore, it is worth doing the same for attitude control as earth observation missions require the capabilities to conduct pointing modes.

REFERENCES

[1] C. S. Agency, "What is a CubeSat," [Online]. Available: https://www.asc-csa.gc.ca/eng/satellites/cubesat/what-is-a-cubesat.asp. [Accessed May 2021].

[2] M. Langer, "Reliability of CubeSats – Statistical Data, Developers' Beliefs and the Way Forward," in *30th Annual AIAA/USU Conference on Small Satellites*, 2016.

[3] E. Kulu, "Nanosats Database," [Online]. Available: https://www.nanosats.eu/. [Accessed May 2021].

[4] Elprocus, "Know about Under and Overvoltage Protection Circuit with Working," [Online]. Available: https://www.elprocus.com/under-and-overvoltage-protection-circuit/. [Accessed May 2021].

[5] "NanoRacks CubeSat Deployer (NRCSD) Interface Definition Document (IDD)," 04 June 2018. [Online]. Available: https://nanoracks.com/wp-content/uploads/Nanoracks-CubeSat-Deployer-NRCSD-IDD.pdf. [Accessed May 2021].

[6] M. Swartwout, "The First One Hundred CubeSats: A Statistical Look," *Journal Of Small Satellites*, vol. 2, no. 2, pp. 213-233, 2013.

[7] L. Kepko, L. Santos, C. Clagett, B. Azimi, D. Chai, A. Cudmore, S. Starin and James Marshall, "Dellingr: Reliability lessons learned from on-orbit," in *32nd Annual AIAA/USU Conference on Small Satellites*, 2018.

[8] N. Saeed, A. Elzanaty, H. Almorad, H. Dahrouj, T. Y. Al-Naffouri and M.-S. Alouini, "CubeSat Communications: Recent Advances and Future Challenges," *Institution of Electrical and Electronic Engineers*, vol. 3, no. 3, pp. 1-24, 2020.

[9] T. Villela, C. A. Costa, A. M. Brandão, F. T. Bueno and R. Leonardi, "Towards the Thousandth CubeSat: A Statistical Overview," *International Journal of Aerospace Engineering*, vol. 2019, no. 5063145, pp. 1-13, 2019.

[10] "Document Tree," European Cooperation for Space Standardization (ECSS), [Online]. Available: https://ecss.nl/standards/ecss-document-tree-and-status/. [Accessed May 2021].

[11] "NASA Technical Standards System: CCSDS Standards," National Aeronatutics and Space Administration (NASA), 30 September 2015. [Online]. Available: https://standards.nasa.gov/ccsds-standards. [Accessed May 2021].

[12] L. D. G. Matignon, "THE KESSLER SYNDROME," Space Legal Issues, 27 March 2019. [Online]. Available: https://www.spacelegalissues.com/space-law-the-kessler-

syndrome/. [Accessed May 2021].

[13] "PC/104," Wikimedia Foundation Inc., 2010. [Online]. Available: https://en-academic.com/dic.nsf/enwiki/861848#Form_factors. [Accessed May 2021].

[14] M. Southworth, "PC/104 and UASs: The beauty of the system," *PC104 and Small form Factors The Journal of Modular Design,* pp. 1-6, 2014.

[15] A. Winfield and O. Holland, "The application of wireless local area network technology to the control of mobile robots," *Microprocessors and Microsystems,* vol. 23, no. 10, pp. 597-607, 2000.

[16] A. Parness, N. Abcouwer, C. Fuller, N. Wiltsie, J. Nash and B. Kennedy, "LEMUR 3: A Limbed Climbing Robot for Extreme Terrain Mobility in Space," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[17] "CubeSat Concept and the Provision of Deployer Services," European Space Agency (ESA), [Online]. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/cubesat-concept. [Accessed May 2021].

[18] "Platform Software," GOMSPACE, [Online]. Available: https://gomspace.com/shop/subsystems/command-and-data-handling/platform-software.aspx. [Accessed April 2021].

[19] "Flight Software Developement Kit," Bright Ascension, [Online]. Available: https://brightascension.com/products/flight-software-development-kit/. [Accessed April 2021].

[20] H. D. Black, "Early development of Transit, the Navy navigation satellite system," *Journal of Guidance, Control, and Dynamics,* vol. 13, no. 4, pp. 577-585, 1990.

[21] "Transit (satellite)," Wikimedia Foundation Inc., 9 July 2021. [Online]. Available: https://en.wikipedia.org/wiki/Transit_(satellite). [Accessed August 2021].

[22] M. Taraba, C. Rayburn, A. Tsuda and C. ". MacGillivray, "Boeing's CubeSat TestBed1 Attitude Determination Design and On-Orbit Experience," in *23rd Annual AIAA/USU Conference on Small Satellites*, 2009.

[23] "CSTB1 (CubeSat TestBed 1)," European Space Agency (ESA), [Online]. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/cstb1. [Accessed May 2021].

[24] N. G. Guecha-Ahumada and E. G. Lightsey, "Design of the Attitude Estimation Algorithms for the GT-1 CubeSat," Georgia Institute of Technology, 2020.

[25] M. Shuster and S. Oh, "Three-axis attitude determination from vector observations,"

Journal of Guidance, Control, and Dynamics, vol. 4, no. 1, pp. 70-77, 1981.

[26] K. M. d. Santos, A. L. d. Silva and F. d. S. N. Coelho, "Attitude Determination For A Brazilian CubeSat Mission Using The Kalman Filter," in *24th ABCM International Congress of Mechanical Engineering*, 2017.

[27] F. L. Markley and J. L. Crassidis, Fundamentals of Spacecraft Attitude Determination and Control, Springer: New York, 2014.

[28] S. Ahmed and E. C. Kerrigan, "Robust Static Attitude Determination via Robust Optimization," *The International Federation of Automatic Control (IFAC) Proceedings Volumes,* vol. 44, no. 1, pp. 5807-5812, 2011.

[29] M. D. Shuster, "The Quest for Better Attitudes," *The Journal of the Astronautical Sciences,* vol. 54, no. 3 & 4, p. 657–683, 2006.

[30] M. Wade, "HEAO," The Encyclopedia Astronautica, [Online]. Available: http://www.astronautix.com/h/heao.html. [Accessed June 2021].

[31] V. Sazdovski, P. M. G. Silson and A. Tsourdos, "Attitude Determination from Single Camera Vector Observations," in *Institute of Electrical and Electronics Engineers (IEEE) Intelligent Systems Conference*, London, 2010.

[32] Y. Cheng and M. D. Shuster, "An Improvement to the Implementation of the QUEST Algorithm," *Journal of Guidance, Control, and Dynamics,* vol. 37, no. 1, pp. 301-305, 2014.

[33] A. H. d. Ruiter, C. Damaren and J. R. Forbes, Spacecraft Dynamics and Control: An Introduction, Wiley, 2013.

[34] "MagSat," European Space Agency (ESA), [Online]. Available: https://earth.esa.int/web/eoportal/satellite-missions/m/magsat. [Accessed June 2021].

[35] R. Langel, G. Ousley, J. Berbert, J. Murphy and M. Settle, "The MAGSAT mission," *Geophysical Research Letters,* vol. 9, no. 4, p. 243–245, 1982.

[36] "CAT-2: Attitude Determination and Control System for a GNSS-R Earth Observation 6U CubeSat Mission," *European Journal of Remote Sensing,* vol. 49, no. 1, pp. 759-776, 2016.

[37] A. Scholz, W. Ley, B. Dachwald, J. Miau and J. Juang, "Flight results of the COMPASS-1 picosatellite mission," *Acta Astronautica,* vol. 67, no. 9 & 10, p. 1289–1298, 2010.

[38] A. O. Erlank and W. H. Steyn, "Arcminute Attitude Estimation for CubeSats with a Novel Nano Star Tracker," *The International Federation of Automatic Control (IFAC)*

Proceedings Volumes, vol. 47, no. 3, pp. 9679-9684, 2014.

[39] F. L. Markley, "Equivalence of Two Solutions of Wahba's Problem," *The Journal of the Astronautical Sciences,* vol. 60, no. 3, p. 303–312, 2013.

[40] J. D. Searcy, "Magnetometer-only attitude determination with application to the M-SAT mission," Masters Thesis, Missouri University of Science and Technology, 2011.

[41] F. L. Markley, "Statistical Attitude Determination," Encyclopedia of Aerospace Engineering, 2010.

[42] K. Krogh and E. Schreder, "Attitude Determination for AAU CubeSat," 6 June 2002. [Online]. Available: https://www.yumpu.com/en/document/view/4458650/attitude-determination-report-pdf-studentspace-spaceaaudk. [Accessed October 2020].

[43] L. Baroni, "Kalman filter for attitude determination of a CubeSat using low-cost sensors," *Computational & Applied Mathematics,* vol. 37, no. 1, pp. 72-83, 2017.

[44] E. P. Babcock, "CubeSat Attitude Determination via Kalman Filtering of Magnetometer and Solar Cell Data," in *25th Annual AIAA/USU Conference on Small Satellites,* 2011.

[45] D. Bhatia, "Attitude Determination and Control System Design of Sub-Arcsecond Pointing Spacecraft," *Journal of Guidance, Control, and Dynamics,* vol. 44, no. 2, pp. 295-314, 2020.

[46] R. Roubache, M. Benyettou, A. M. Simohammed, A. Boudjemai and A. Bellar, "Extended Kalman filter for attitude determination of an elliptical orbit satellites," in *6th International Conference on Recent Advances in Space Technologies (RAST)*, 2013.

[47] R. Burton, S. Rock, J. Springmann and J. Cutler, "Online Attitude Determination of a Passively Magnetically Stabilized Spacecraft," *Acta Astronaut,* vol. 133, no. 1, pp. 269-281, 2017.

[48] M. Alger, *Development of an Attitude Determination filter using Single-axis Sun Sensors and a Magnetometer,* Toronto: Ryerson University, 2017.

[49] T. Wright, "Development Of Magnetometer-Based Orbit And Attitude Determination For Nanosatellites," Masters Thesis, York University, Toronto, 2014.

[50] H. Ma, Z. Lu, X. Zhang, W. Liao and K. Briess, "High-Accuracy and Low-Cost Attitude Measurement Unit of the CubeSat," *International Journal of Aerospace Engineering,* vol. 2020, no. 1, pp. 1-13, 2020.

[51] O. Awad and R. H. Bishop, "A MULTIPLICATIVE EXTENDED KALMAN FILTER FOR LOW EARTH ORBIT ATTITUDE ESTIMATION ABOARD A 0.5U SMALLSAT," in *43rd Annual AAS Guidance, Navigation, and Control Conference*,

Breckenridge, 2020.

[52] F. Qin, L. Chang, S. Jiang and F. Zha, "A Sequential Multiplicative Extended Kalman Filter for Attitude Estimation Using Vector Observations," *Sensors,* vol. 18, no. 5, pp. 1-10, 2018.

[53] B. Liu, Z. Chen, X. Liu and F. Yang, "An Efficient Nonlinear Filter for Spacecraft Attitude Estimation," *International Journal of Aerospace Engineering,* vol. 2014, no. 1, pp. 1-11, 2014.

[54] B. Lobo-Fernandes, "Design and Analysis of an Attitude Determination and Control," Instituto Superior T´ecnico, Universidade de Lisboa, Lisboa, 2019.

[55] I. Gautam, "Quaternion Based Attitude Estimation Technique Involving The Extended Kalman Filter," Masters Thesis, The University of Akron, Akron, 2019.

[56] J. Zhou, X. Li, R. Liu and Y. Liu, "A scheme of satellite multi-sensor fault-tolerant attitude estimation," *Sage Journals,* vol. 38, no. 9, p. 1053–1063, 2016.

[57] B. Gao, G. Hu, S. Gao, Y. Zhong and C. Gu, "Multi-Sensor Optimal Data Fusion Based on the Adaptive Fading Unscented Kalman Filter," *Sensors,* vol. 18, no. 2, pp. 1-22, 2018.

[58] V. G. Goecks, A. Probe, R. M. Woollands, J. Hurtado and J. Junkins, "Low Cost Spacecraft Attitude Determination for Cubesat Type Missions," in *AAS Guidance and Control Conference,* Breckenridge, 2016.

[59] D. Barbosa, A. Lopes and R. E. Araújo, "Sensor fusion algorithm based on Extended Kalman Filter for estimation of ground vehicle dynamics," in *42nd Annual Conference of the IEEE Industrial Electronics Society,* 2016.

[60] Y. Yu, J. Liu, Z. Xiong and R. Li, "Research on multi-sensor information fusion system application for UAVS with long range and high altitude," in *IEEE/ION Position, Location and Navigation Symposium,* 2010.

[61] F. Edrisi and V. J. Majd, "Attitude estimation of an accelerated rigid body with sensor fusion based-on switching extended Kalman filter," in *2015 AI & Robotics (IRANOPEN),* 2015.

[62] F. Raiti, "Efficient and Robust Attitude Determination and Control System Design for the MIST CubeSat," Ph.D. Dissertation, KTH Royal Institute of Technology, 2018.

[63] J. Ma, X. Wang, F. Blaabjerg, L. Harnefors and W. Song, "Accuracy Analysis of the Zero-Order Hold Model for Digital Pulse Width Modulation," *IEEE Transactions on Power Electronics,* vol. 33, no. 12, pp. 10826 - 10834, 2018.

[64] C. Yang, S. Gao and Z. Xue, "Improving the Closed-Loop Tracking Performance Using the First-Order Hold Sensing Technique with Experiments," 4 January 2018. [Online]. Available: https://arxiv.org/abs/1801.01263. [Accessed June 2021].

[65] M. D. Ortigueira and J. T. Machado, "A Review of Sample and Hold Systems and Design of a," *New Fractional Algorithm,* vol. 10, no. 20, pp. 1-14, 2020.

[66] M. Macdonald and V. Badescu, The International Handbook of Space Technology, Springer & Praxis Publishing, 2014.

[67] D. A. Vallado, Fundamentals of Astrodynamics and Applications, 4th ed., Microcosm Press, 2013.

[68] "ADCS For Beginners," Wordpress, [Online]. Available: https://adcsforbeginners.wordpress.com/. [Accessed June 2021].

[69] J. H. Seago and D. A. Vallado, "Coordinate frames of the U.S. Space Object Catalogs," in *Astrodynamics Specialist Conference*, 2000.

[70] A. K. Mallik, "Precession of the Earth's Axis," *Himalayan Physics,* vol. 1, no. 1, pp. 83-84, 2011.

[71] "PDS Geosciences Node," 7 August 2009. [Online]. Available: https://pds-geosciences.wustl.edu/grail/grail-l-rss-2-edr-v1/grail_0201/document/lib_10_grail_coord_trans_rev1.pdf. [Accessed July 2021].

[72] "Spherical coordinate system," Wikimedia Foundation Inc., [Online]. Available: https://en.wikipedia.org/wiki/Spherical_coordinate_system. [Accessed May 2021].

[73] KERA, "Nanomind A3200," 29 January 2021. [Online]. Available: https://gomspace.com/UserFiles/Subsystems/datasheet/gs-ds-nanomind-a3200_1006901-117.pdf. [Accessed July 2021].

[74] "MPU-3300 Datasheet by TDK InvenSense," 23 July 2012. [Online]. Available: https://www.digikey.com.au/htmldatasheets/production/1278147/0/0/1/mpu-3300-datasheet.html. [Accessed June 2021].

[75] A. Giese, "Ongoing problems with special and general relativity, and solutions," *Journal of Physics Conference Series,* vol. 1251, pp. 012017-012025, 2019.

[76] Fizoptika, "Series VG091," August 2020. [Online]. Available: https://fizoptika.com/docs/fiber_optic_gyro_specification_vg091a.pdf. [Accessed July 2021].

[77] Texas Instruments Incorporated, "ADS1115," 8 January 2018. [Online]. Available: https://www.ti.com/product/ADS1115#tech-docs. [Accessed June 2021].

[78] KLK, "NanoPower P110," 22 October 2018. [Online]. Available: https://gomspace.com/UserFiles/Subsystems/datasheet/gs-ds-nanopower-p110-210.pdf. [Accessed May 2021].

[79] RAH, "NanoSense FSS," 23 June 2020. [Online]. Available: https://gomspace.com/UserFiles/Subsystems/datasheet/gs-ds-nanosense-fss-30.pdf. [Accessed July 2021].

[80] MB, "NanoSense FSS," 8 March 2018. [Online]. Available: https://gomspace.com/UserFiles/Subsystems/manual/gs-man-nanosense-fss-21.pdf. [Accessed July 2021].

[81] Honeywell International Inc., "Triple Axis Magnetometer Breakout - HMC5843," 2009. [Online]. Available: https://www.sparkfun.com/products/retired/9371. [Accessed July 2021].

[82] J. S. Subirana, J. J. Zornoza and M. Hernández-Pajares, "Reference Frames in GNSS," European Space Agency (ESA), 2011. [Online]. Available: https://gssc.esa.int/navipedia/index.php/Reference_Frames_in_GNSS#:~:text=7%20References-,GPS%20reference%20frame%20WGS%2D84,vector%20referencing%20%5Bfootnotes%201%5D. [Accessed July 2021].

[83] A. Mohan, "Calculating Position from Raw GPS Data," TELESENS, 17 July 2017. [Online]. Available: https://www.telesens.co/2017/07/17/calculating-position-from-raw-gps-data/. [Accessed July 2021].

[84] NavtechGPS, "NovAtel OEM615 Series Receivers," May 2014. [Online]. Available: https://2w6vmg3m8cv83pn80b2dfi9f-wpengine.netdna-ssl.com/wp-content/uploads/assets/1/7/OEM615_DS1.PDF. [Accessed July 2021].

[85] NavtechGPS, "NovAtel OEM615 Series Receivers," October 2016. [Online]. Available: https://hexagondownloads.blob.core.windows.net/public/Novatel/assets/Documents/Manuals/om-20000128/om-20000128.pdf. [Accessed July 2021].

[86] "Earth orientation data," International Earth Rotation and Reference Systems Service (IERS), 2013. [Online]. Available: https://www.iers.org/IERS/EN/Home/home_node.html. [Accessed June 2021].

[87] "NORAD Two-Line Element Sets," CelesTrak, [Online]. Available: http://celestrak.com/NORAD/elements/. [Accessed July 2016].

[88] CelesTrak, "Documentation," [Online]. Available: https://www.celestrak.com/NORAD/documentation/spacetrk.pdf. [Accessed October 2020].

[89] "Longitude of the ascending node," Wikimedia Foundation, Inc., [Online]. Available: https://en.wikipedia.org/wiki/Longitude_of_the_ascending_node. [Accessed June 2021].

[90] "The International Celestial Reference System (ICRS)," International Earth Rotation and Reference Systems Service (IERS), 2013. [Online]. Available: https://www.iers.org/IERS/EN/Science/ICRS/ICRS.html. [Accessed July 2021].

[91] D. D. McCarthy, "IERS Standards (1989) (IERS Technical Note 3)," 1989. [Online]. Available: https://www.iers.org/SharedDocs/Publikationen/EN/IERS/Publications/tn/TechnNote03/tn03_04.pdf?__blob=publicationFile&v=1. [Accessed July 2021].

[92] "What is Polar Motion?," Naval Oceanography Portal, [Online]. Available: https://www.usno.navy.mil/USNO/earth-orientation/eo-info/faq/what-is-pm. [Accessed July 2021].

[93] H. D. Curtis, Orbital Mechanics for Engineering Students Third Edition, Waltham: Elsevier Canada, 2014.

[94] C. Stolle, I. Michaelis, C. Xiong, M. Rother, T. Usbeck, Y. Yamazaki, J. Rauberg and K. Styp-Rekowski, "Observing Earth's magnetic environment with the GRACE-FO mission," *Earth, Planets and Space,* vol. 73, no. 1, pp. 1-21, 2021.

[95] "International Geomagnetic Reference Field," International Association of Geomagnetism and Aeronomy (IAGA), 19 December 2019. [Online]. Available: https://www.ngdc.noaa.gov/IAGA/vmod/igrf.html. [Accessed July 2021].

[96] C. M. Roithmayr, "Contributions of Spherical Harmonics to Magnetic and Gravitational Fields," 1 March 2004. [Online]. Available: https://ntrs.nasa.gov/citations/20040047194. [Accessed October 2020].

[97] E. Thébault, C. Finlay, C. Beggan, P. Alken, J. Aubert, O. Barrois, F. Bertrand, T. Bondar, A. Boness, L. Brocco and e. al., "International Geomagnetic Reference Field: the 12th," *Earth Planets and Space,* vol. 67, pp. 1-19, 2015.

[98] X. Zhu, M. Ma, D. Cheng and Z. Zhou, "An Optimized Triad Algorithm for Attitude Determination," *Artificial Satellites,* vol. 52, no. 3, pp. 41-47, 2017.

[99] STMicroelectronics, "LSM6DSM," September 2017. [Online]. Available: https://www.st.com/resource/en/datasheet/lsm6dsm.pdf. [Accessed June 2021].

[100] MEMSIC , "MMC5603NJ," 7 December 2018. [Online]. Available: https://www.mouser.ca/datasheet/2/821/Memsic_09102019_MMC5603NJ_Datasheet_Rev.B-1635324.pdf. [Accessed June 2021].

[101] J. S. Subirana, J. J. Zornoza and M. Hernández-Pajares, "Transformations between ECEF and ENU coordinates," European Space Agency (ESA), 2011. [Online]. Available: https://gssc.esa.int/navipedia/index.php/Transformations_between_ECEF_and_ENU_coordinates. [Accessed August 2021].

[102] M. M. Carrasco and A. L. d. Silva, "Attitude Determination for Low Cost IMU and Processor Board using the Methods of TRIAD, Kalman Filter and Allan Variance," *REVISTA BRASILEIRA DE INICIAÇÃO CIENTÍFICA*, vol. 3, pp. 26-41, 2016.

[103] T. Michel, P. Genevès, H. Fourati and N. Layaïda, "On Attitude Estimation with Smartphones," in *IEEE International Conference on Pervasive Computing and Communications*, Kona, 2017.

[104] "magcal," MathWorks, 2019. [Online]. Available: https://www.mathworks.com/help/nav/ref/magcal.html. [Accessed August 2021].

[105] D. Modenini, A. Bahu, G. Curzi and A. Togni, "A Dynamic Testbed for Nanosatellites Attitude Verification," *Aerospace*, vol. 7, no. 3, pp. 1-19, 2020.

[106] R. Shimmin, "Using a smartphone camera for nanosatellite attitude determination," in *Advanced Maui Optical and Space Surveillance Technologies*, 2014.