DESIGNING COMPUTATIONALLY LIGHT ALGORITHMS FOR CONCURRENT REAL-TIME ATTITUDE ESTIMATION AND SENSOR CALIBRATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DOĞUKAN BENLI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

MAY 2023

Approval of the thesis:

DESIGNING COMPUTATIONALLY LIGHT ALGORITHMS FOR CONCURRENT REAL-TIME ATTITUDE ESTIMATION AND SENSOR CALIBRATION

submitted by **DOĞUKAN BENLI** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Serkan Özgen
Head of Department, **Aerospace Engineering** _____

Assoc. Prof. Dr. Halil Ersin Söken
Supervisor, **Aerospace Engineering** _____

Examining Committee Members:

Prof. Dr. Ozan Tekinalp
Aerospace Engineering, METU _____

Assoc. Prof. Dr. Halil Ersin Söken
Aerospace Engineering, METU _____

Assist. Prof. Dr. Ali Türker Kutay
Aerospace Engineering, METU _____

Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering, METU _____

Assist. Prof. Dr. Yakup Özkazanç
Electrical and Electronics Engineering, Hacettepe University _____

Date:23.05.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Doğukan Benli

Signature :

ABSTRACT

DESIGNING COMPUTATIONALLY LIGHT ALGORITHMS FOR CONCURRENT REAL-TIME ATTITUDE ESTIMATION AND SENSOR CALIBRATION

Benli, Doğukan
M.S., Department of Aerospace Engineering
Supervisor: Assoc. Prof. Dr. Halil Ersin Söken

May 2023, 70 pages

Computer processors' computational power and efficiency continue to improve each year. This improvement enables the usage of less power for computing or more complex models for embedded systems, especially in the aerospace industry. However, implementing newer and more advanced technologies to the systems in the field or production can be challenging. Therefore, focusing on more capable or efficient algorithms for these systems is still crucial. This research aims to develop and evaluate methods that decrease the computational load of onboard attitude estimation algorithms that use the Kalman Filter (KF) as the core algorithm. Computationally efficient algorithms are essential for small satellites as they are limited in hardware and power consumption. A real-time runnable accurate attitude estimation algorithm that also estimates the additional parameters, such as the sensor errors, can be beneficial for these resource-limited satellites.

This thesis focuses on reducing the number of KF updates without compromising the performance of the KF and demonstrates its performance on the small satellite attitude estimation problem. This task requires manipulating the measurements to get

a pseudo-measurement of slower frequency. To this end, “integrated measurements” are suggested to replace the original measurements.

Keywords: attitude estimation, computational cost, integrated measurements, Multiplicative Extended Kalman Filter, sensor calibration

ÖZ

HESAPLAMA YÜKÜ HAFİF EŞ VE GERÇEK ZAMANLI YÖNELİM KESTİRİM VE SENSÖR KALİBRASYON ALGORİTMALARI TASARIMI

Benli, Doğukan
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Halil Ersin Söken

Mayıs 2023 , 70 sayfa

Her geçen yıl işlemcilerin gücü ve verimi artmakta ve bu sayede, gömülü sistemlerin daha az güç kullanması veya daha karmaşık modeller içermesi mümkün olmaktadır. Ancak, yeni ve gelişmiş teknolojilerin sahadaki veya üretimdeki sistemlere aktarılması zorlu veya zaman alan bir süreçtir. Bu sebeple, kabiliyeti daha yüksek veya daha verimli algoritmaların geliştirilmesi hala büyük önem taşımaktadır. Bu tezdeki araştırma, sistem üzerinde çalışan Kalman Filtreli yönelim kestirimi algoritmalarının hesaplama yükünü azaltan yöntem geliştirme ve bu yöntemleri değerlendirmeyi hedeflemektedir. Hesaplama yükü özellikle bu konuda kısıtlı kaynaklara sahip olan küçük uydular gibi sistemler için önemlidir. Gerçek zamanlı yönelim kestirimi gerçekleştirirken, sensör hataları gibi ek parametreleri de kestiren algoritmalar bu sistemler için faydalıdır.

Bu tez kestirim performansından taviz vermeksizin KF güncellemelerinin sıklığını azaltma çalışmasını tarifleyip, bu çalışmanın performansını küçük uydu yönelim kestirimi problemi üzerinde göstermektedir. Bu amaçla, ölçüm sıklığını düşüren "integrallenmiş ölçümler" yöntemi önerilmektedir.

Anahtar Kelimeler: yönelim kestirimi, hesaplama yükü, integrallenmiş ölçümler, Doğrusal olmayan Çarpımsal Kalman Filtresi, sensör kalibrasyonu

Dedicated to my family

ACKNOWLEDGMENTS

Firstly, I wish to express my gratitude to my supervisor Dr. Halil Ersin Söken for his mentorship, patience, and guidance throughout my graduate studies. I would also like to extend my gratitude to the mentors of my early engineering career Dr. Burak Seymen, Dr. Yiğiter Yüksel, and Eren Kahraman. I am also thankful to my colleagues for teaching me a lot of different aspects of engineering. A special mention to Dr. Berk Osunluk for his friendship, mentorship, and constantly encouraging and nagging me to finish my Master's.

I would like to thank my family; my mother for her love and support, my father for his example and wisdom, and my brother for always challenging me and his friendship. I am also indebted to my other family, who has accepted me as their son rather than just as a son-in-law. Thank you all for all the efforts you have put in for me.

Lastly, I owe a great deal of thanks to my wife, *Merve*. Thanks for always supporting me, pushing me forward, being a great friend, and loving me. This thesis would not be possible without you at all.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

DCM	Direction Cosine Matrix
ECEF	Earth-Centered Earth-Fixed Coordinate System
ECI	Earth-Centered Inertial Coordinate System
EKF	Extended Kalman Filter
FPGA	Field Programmable Gate Array
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filter
MC	Monte-Carlo
MEKF	Multiplicative Extended Kalman Filter
PDF	Probability Density Function
RTN	Radial Transverse Normal
SLAM	Simultaneous Localization and Mapping
TRIAD	Triaxial Attitude Determination
UKF	Unscented Kalman Filter
WGS84	World Geodetic System 1984
WMM	World Magnetic Model

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Position and attitude are considered vital information for the operation of various systems such as tanks and cannons on the ground, ships in the vast seas, planes and helicopters in the skies, and satellites and space vehicles in space. Inertial Navigation Systems (INS) are utilized with other external aiding in systems where this information is critical. This aiding usually comes in the form of Global Navigation Satellite Systems. GNSS data can identify a vehicle's position, velocity, and attitude. However, these signals may not always be available or may be spoofed to give incorrect results.

On the other hand, Inertial Navigation Systems utilize Inertial Measurement Units (IMU), which sense inertial motion in the form of acceleration and rotation. This type of information does not rely on any outside signal. Thus, it is uninterrupted and can not be spoofed. Their downfall is the multiple integrals required to achieve position, velocity, and attitude, which builds up significant errors in time. Hence, an INS is usually combined with aiding measurements to get the best of both worlds; uninterrupted and unspoofable data with bounded error margins.

To combine these independent systems, a data fusion algorithm is required. The most common approach to this is a solution to the estimation problem, Kalman Filter (KF). KF is an unbiased optimal estimator (Kalman, 1960). It utilizes known statistical properties of unknown system states along with observations to estimate those unknowns. It also keeps track of correlations between the system states, updating statistical knowledge for future potential use.

In small and light systems such as satellites, a similar approach is commonly used to estimate the attitude (Garcia et al., 2019; Qiu & Guo, 2021). If available, a triad of gyroscopes is used instead of a full IMU. Measurements provided by the gyroscopes are integrated to keep track of the body's attitude, which builds up attitude errors due to the erroneous nature of the gyroscopes. To keep the errors bounded, the Earth's magnetic field and the Sun's direction information, which are obtained as magnetometer and Sun sensor measurements, can be used for aiding. A lightweight KF can be used for this data fusion.

The KF requires several matrix operations, which scale with the number of parameters needed to be estimated. If the KF is used only to estimate the attitude, computational load is very light at the cost of degraded estimations in the presence of sensor errors (Vinther et al., 2011). Sensors can develop additional errors during a mission, leading to critical problems (Yuan et al., 2021). Errors on the sensors must be accounted for in order to get accurate attitude information from a magnetometer (Soken & Sakai, 2020). As the number of parameters increases, so does the computational cost. Thus, the computational load must be carefully considered while implementing the estimation algorithm, specifically for attitude estimation for small satellites with a limited computational resource (Pham et al., 2015).

In the author's experience working with medium and high-grade INSs, computational load for such systems is also critical. To lighten this load, methods that decrease the computational load for estimation using the KF are investigated. A different approach to the existing literature is suggested, and its estimation performance and computational load are compared to a commonly used KF.

This thesis focuses on the attitude estimation problem on a satellite with gyroscopes, magnetometers, and sun sensors to demonstrate its findings. The research is performed with a KF that can perform attitude estimation and sensor calibration concurrently. The KF and attitude integral architectures are combined in a fashion fit for real-time applications.

1.2 Literature Survey

A common approach to reducing the computational cost of KF is to reduce the computational cost of related operations. These operations constitute matrix addition, multiplication, and inversion.

Small satellites are a prime candidate for KF computational cost reduction research. They are constrained in terms of mass, power, and computational power.

Asundi et al., 2011 describes the attitude estimation algorithm used on a picosatellite, SwampSat, where computational resources are "at a premium". It involves the usage of Murrell's version of the EKF, which performs observation as a series of 3×1 vector observations, limiting the matrix inverted for the gain calculation to a size of 3×3. In Asundi et al., 2021, performance of the Murrell's version of the EKF is evaluated for attitude estimation using a non-redundant set of sensor configurations.

Chia et al., 2016 presents a low complexity KF to reduce gyroscope noises for a nano-satellite. The low complexity, in terms of computational power, is achieved by expressing state transition and observation matrices as sparse matrices and utilizing matrix decomposition, which simplifies matrix operations. Pham et al., 2015 proposes the usage of a gain-scheduled EKF to reduce the computational load of the attitude determination process. The Kalman gains are analytically determined beforehand using the sensor specifications for different attitude modes of the satellite. This allows skipping the gain computation, which involves costly inverse matrix operations.

Vinther et al., 2011 demonstrates the usage of Unscented KF for satellite attitude estimation with gyroscopes, magnetometers, and sun sensors. The UKF utilizes multiple sigma points to represent the distribution of the states. These sigma points are individually propagated, increasing the computational cost of the KF. However, due to a better representation of the statistics, it is expected to have lower errors. This makes a reduction in its computational cost even more important for systems with low computational capabilities. Biswas et al., 2018 investigates the Single Propagation UKF and Extrapolated Single Propagation UKF for such a scenario, comparing estimation performance and computational load. Both methods propagate only a single state vector,

the *a posteriori* mean. The other points are constructed using the first-order Taylor series approximation for the former, and multi-dimensional Richardson extrapolation for the latter.

On the other hand, Bonato et al., 2009; J. Guivant and Nebot, 2001; J. E. Guivant, 2002 tackles SLAM (Simultaneous Localization and Mapping) problems, which typically have a very high number of states to represent points on the map. In J. Guivant and Nebot, 2001; J. E. Guivant, 2002, the map is divided into local sub-maps, including only a subset of the entire state vector. Due to this, computations made within this area are cheaper. The whole state vector is required only when information transfer between a sub-map and the global map is to occur. In Bonato et al., 2009, a specialized FPGA architecture is developed which combines the strength of FPGAs with the parallel nature of the matrix operations.

Fatehi and Huang, 2017 describes the usage of *Integrated measurements* in industrial processes where a sample material is gradually collected and then sent to a laboratory for analysis. The laboratory measures the integral of the desired states, called *Integrated measurements*. A modified KF is proposed that uses these slow-rate *Integrated measurements* as its observation to improve the final state observation. Balaguer and Ibeas, 2021 mentions that the *Integrated measurements* contain process noise that increases the variance on the measurement, and presents a guide to computing the optimal averaging/integration window that minimizes it. In Movassagh et al., 2023, a sliding window smoothing algorithm is proposed to enhance the accuracy of the fast state estimations.

Last but not least, in Raitoharju and Piché, 2019, a tutorial for KF complexity reduction is provided. Different approaches are discussed, such as using partial linearity and dividing the state.

1.3 Contribution

This thesis proposes the manipulation of observations to reduce the computational load of the KF by reducing its operational frequency. This differs from the existing literature, where the reduction is achieved for each operation cycle of the KF indi-

vidually. The proposed method is expected to reduce the computational load without compromising the estimation quality. It is also expected to be compatible with the literature, providing further benefits to computational load.

The proposed method, referred to as *Integrated Measurements*, is implemented to a real-time attitude estimation and concurrent sensor calibration algorithm and simulated for a Low Earth Orbit satellite to demonstrate its impact on estimation performance and computational load. It differs from the previously mentioned method with the same name in that, the proposed method manipulates observations and states by taking their integral to reduce computational load, rather than deriving a method to observe states by their integrals for smoother estimations. An early version of this approach was presented in Benli and Söken, 2022

1.4 The Outline of the Thesis

Chapter 1 contains an introduction to the problem at hand, the motivation for this research, and the available literature on the subject.

In Chapter 2, preliminary information concerning the attitude estimation problem within the small satellite context is shared. It lists the commonly used coordinate systems, describes the coordinate transformation concept from the ground up with alternative methods, presents the common coordinate transformations, and summarizes the attitude update algorithm and model. It also contains the trajectory and data generation process for a simulation aimed to test the proposed method against a baseline.

Chapter 3 describes the attitude estimation problem, and methods utilized for it. It also briefly introduces the Kalman Filter, building up to the variation used in this thesis, MEKF.

In Chapter 4, integrated measurements approach for attitude filtering is introduced and detailed. The chapter contains the required operations and modifications for its application.

Chapter 5 demonstrates the estimation and computational performance of the proposed approach. It contains estimation results for an example simulation and Monte-

Carlo simulations within the described trajectory range.

Finally, Chapter 6 summarizes the results and concludes the research. Also, possible future research is touched upon.

CHAPTER 2

ATTITUDE, MODELLING, AND SIMULATION ENVIRONMENT

In order to develop, test, and evaluate the proposed algorithm, an appropriate environment must be developed. This chapter describes related coordinate systems, data generation from ground truth, sensor error models, and system error models used throughout this study.

2.1 Coordinate Systems

A coordinate system is a set of basis vectors that form a vector space. For the purposes of this study, a coordinate system refers to an orthonormal set of basis vectors that form the 3-dimensional vector space, that is, the classical Euclidean Coordinates.

Coordinate systems are commonly used to represent vectors in space. However, the orientation of a body can be defined with respect to a reference coordinate system via the linear transformation between them. The coordinate systems used in this thesis are discussed in this section.

2.1.1 Earth Centered Inertial Coordinate System (ECI)

ECI is a non-rotating coordinate system whose origin is fixed to the Earth, and it is commonly utilized as the inertial reference for close-earth proximity problems. There are two common versions of ECI; ecliptic and equatorial ECI. This thesis refers to the ecliptic ECI unless explicitly stated otherwise.

The unit vectors for the ECI, shown in Figure 2.1, are as follows:

- X-axis lies within the ecliptic plane of the Earth and is in the direction of the line connecting the Sun and the Earth at the vernal equinox.
- Z-axis is normal to the ecliptic plane and in the direction of the angular momentum of the Earth's solar orbit.
- Y-axis completes the orthonormal set.

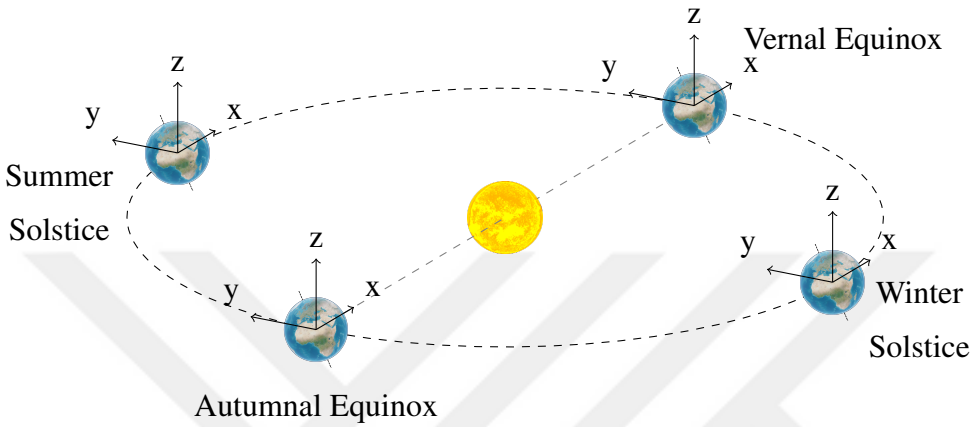

Figure 2.1: (Ecliptic) ECI

2.1.2 Earth Centered Earth Fixed Coordinate System (ECEF)

ECEF is a non-inertial coordinate system whose origin and axes transform with the Earth. This thesis uses the WGS84 definition for the axes.

The unit vectors for the ECEF, shown in Figure 2.2, are as follows:

- X-axis lies within the equatorial plane and is in the direction of the line connecting the geodetic center of the Earth and the prime meridian.
- Z-axis is normal to the equatorial plane and is the direction of the Earth's North Pole.
- Y-axis completes the orthonormal set.

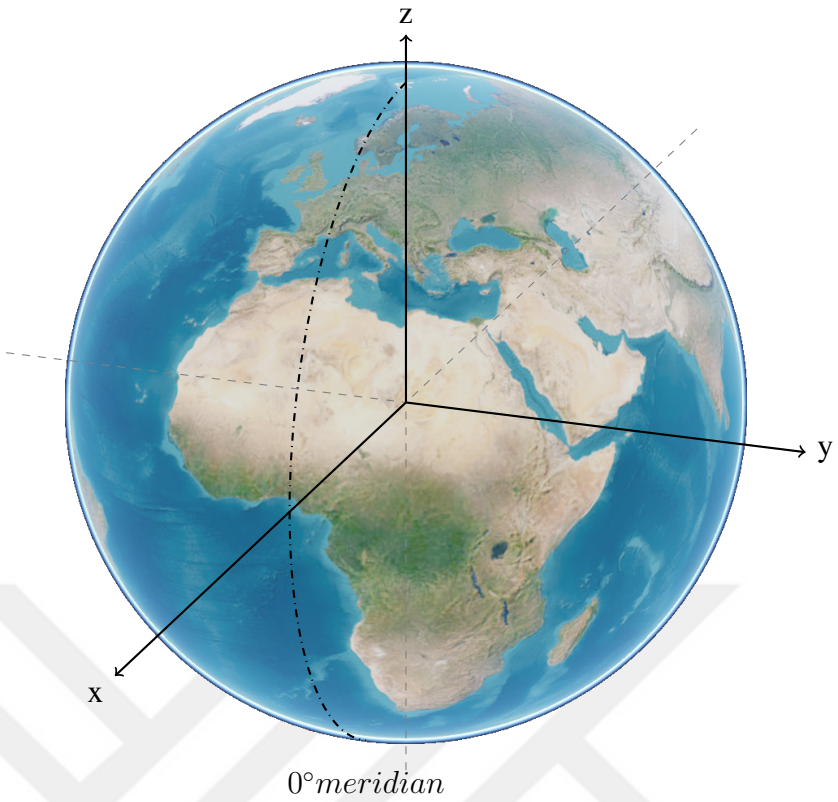

Figure 2.2: ECEF

2.1.3 Perifocal Orbital Coordinate System

Perifocal Orbital Coordinate System is defined by the orbit, and its origin lies at the periapsis point of the orbit and is inertially fixed for a constant orbit. Its primary function is to reduce the 3-dimensional orbit into a 2-dimensional orbit in the orbital plane.

The unit vectors for the Perifocal Orbital Coordinate System , shown in Figure 2.3, are as follows:

- X-axis lies within the orbital plane and is in the direction of the line connecting the focal point of the orbit to the periapsis.
- Z-axis is normal to the orbital plane and is in the direction of the orbit's angular momentum.

- Y-axis completes the orthonormal set.

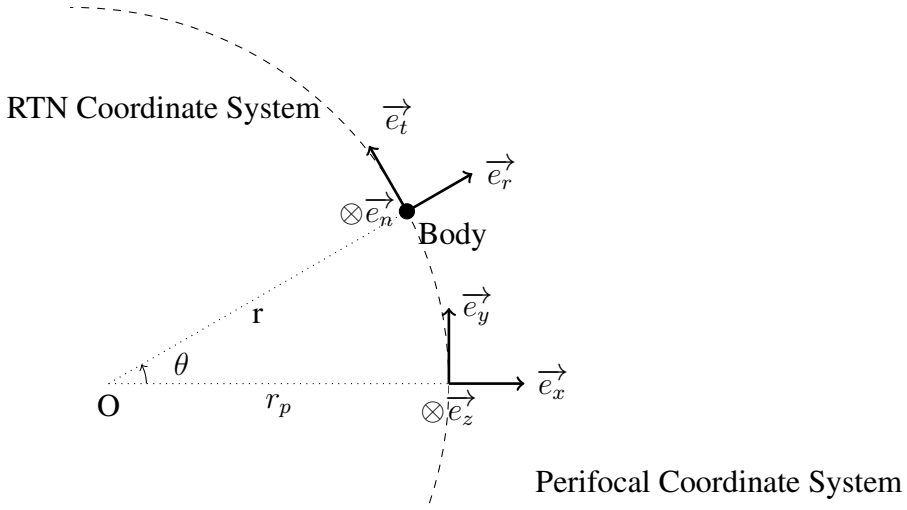

Figure 2.3: RTN and Perifocal Coordinate System

2.1.4 Radial, Transverse, and Normal Orbital Coordinate System (RTN)

RTN is a non-inertial orbital coordinate system defined by the orbit and the position vector. Its origin is fixed to the position along the orbit, and it rotates at the same rate as the position vector.

The unit vectors for the RTN, shown in Figure 2.3, are as follows:

- X-axis lies within the orbital plane and is in the Zenith direction, the direction of the position vector.
- Z-axis is normal to the orbital plane and is in the direction of the orbit's angular momentum.
- Y-axis completes the orthonormal set.

2.1.5 Body Coordinate System

The body coordinate system is a pre-defined coordinate system attached to the body of interest. It translates and rotates with the body, keeping a fixed orientation to the

rigid body. It is usually used to define the body's orientation with respect to a known reference coordinate system.

The body coordinate system is usually defined using the symmetries within the body's shape. Although there are commonly used conventions, it is not strictly defined.

- Earth-pointing satellites generally have their Z-axis (camera axis) Nadir pointing, X-axis along the orbit trajectory, and Y-axis completing the set.
- Aircraft utilize their longitudinal symmetry plane to define the X-Z Plane, with the X-axis in the direction of the nose.
- Smartphones use their vertical symmetry line as X-axis and screen normal as Z-axis.
- Conventional land vehicles use their forward-vertical symmetry plane as their X-Z Plane, with X-axis in the forward direction.

2.2 Orientation Representations

Coordinate transformation is a change in basis from the unit vectors of an origin coordinate system to that of a destination coordinate system. Many representations exist for this transformation, such as Direction Cosine Matrix(DCM), Quaternions, Rotation Vector, and Euler Angles. Since a coordinate transformation ultimately defines the orientation between two coordinate systems, the attitude of a body can be represented by the coordinate transformation between a reference coordinate system and the body coordinate system.

2.2.1 DCM

DCM is the matrix representation of the coordinate transformation, and it contains a projection vector for each of the origin coordinate system unit vectors to the destination coordinate system. It should also be noted that DCM is equivalent to the Rotation Matrix that rotates the coordinate system; as such, it shares all the properties of the Rotation Matrix.

Let $\boldsymbol{i_O}, \boldsymbol{j_O}, \boldsymbol{k_O}$ be the unit vectors for the origin(O) orthonormal coordinate system and $\boldsymbol{i_D}, \boldsymbol{j_D}, \boldsymbol{k_D}$ be the unit vectors for the destination(D) orthonormal coordinate system.

The transformation matrix that transforms the basis from O to D, which C_O^D denotes, is obtained by transforming the unit vectors of O to D.

A vector, $\boldsymbol{V}$, can be written in a desired orthonormal basis by utilizing the orthogonal projection method.

$$\boldsymbol{V_D} = \begin{bmatrix} \boldsymbol{i_D} & \boldsymbol{j_D} & \boldsymbol{k_D} \end{bmatrix} \begin{bmatrix} \langle \boldsymbol{V}, \boldsymbol{i_D} \rangle \\ \langle \boldsymbol{V}, \boldsymbol{j_D} \rangle \\ \langle \boldsymbol{V}, \boldsymbol{k_D} \rangle \end{bmatrix} \tag{2.1}$$

where $\langle , \rangle$ is the inner product operator.

Projecting the basis vectors of O onto the D and concatenating results in the transformation matrix (Friedberg et al., 2014). Elements of a DCM are cosines of the angles between the combination of directions of the two coordinate systems.

$$C_O^D = \begin{bmatrix} \langle \boldsymbol{i_O}, \boldsymbol{i_D} \rangle & \langle \boldsymbol{j_O}, \boldsymbol{i_D} \rangle & \langle \boldsymbol{k_O}, \boldsymbol{i_D} \rangle \\ \langle \boldsymbol{i_O}, \boldsymbol{j_D} \rangle & \langle \boldsymbol{j_O}, \boldsymbol{j_D} \rangle & \langle \boldsymbol{k_O}, \boldsymbol{j_D} \rangle \\ \langle \boldsymbol{i_O}, \boldsymbol{k_D} \rangle & \langle \boldsymbol{j_O}, \boldsymbol{k_D} \rangle & \langle \boldsymbol{k_O}, \boldsymbol{k_D} \rangle \end{bmatrix} \tag{2.2}$$

One important property of a DCM is that its inverse is equal to its transpose. It is also equal to the coordinate transformation in the reverse direction.

$$(C_O^D)^{-1} = (C_O^D)^T = C_D^O \tag{2.3}$$

The main advantage of using a DCM lies in the fact that it is the direct representation of the coordinate transformation operation. As such, rotating a vector does not require additional operations or definitions. This operation takes 6 additions and 9 multiplications. It spans all 3-dimensional rotation space and has no algebraic singularities. Consecutive transformations can be chained together to get a combined coordinate transformation.

$$C_A^C = C_B^C C_A^B \tag{2.4}$$

Its disadvantage comes from its size. It contains 9 variables in its full form, and the transformation combining requires the multiplication of two $3 * 3$ matrices. The combined transformation is computed by 18 addition and 27 multiplication operations (Stančin & Tomažič, 2020).

It should also be noted that a DCM may lose its orthonormal property due to attitude updates. There are methods to restore this property without guaranteeing convergence to the desired DCM.

This thesis utilizes DCMs whenever a coordinate transformation is performed on a vector.

The following special definitions are used to represent the DCM of an elemental rotation:

$$C_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & sin(\theta) \\ 0 & -sin(\theta) & cos(\theta) \end{bmatrix} \tag{2.5}$$

$$C_2(\theta) = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix} \tag{2.6}$$

$$C_3(\theta) = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

Functions given in Equation 2.5, 2.6, and 2.7 represent elemental rotations around x, y, and z axes, respectively. They are such that each transforms a vector into a coordinate system that is rotated in the axis they represent.

2.2.2 Quaternion

Quaternions skew field is 4 parameter extension of the complex numbers field (Buchmann, 2011). In the field of complex numbers, multiplication is a linear transformation operation that scales and rotates its target. Its magnitude determines scaling, while the ratio of real and imaginary components determines the rotation angle. Quaternions carry these same properties, which can be utilized to achieve 3-dimensional rotations.

A quaternion is comprised of a real number and three imaginary numbers and is usually represented as:

$$q = w + xi + yj + zk \tag{2.8}$$

where i, j, k are imaginary numbers with non-commutative multiplication and are defined by the equalities $i^2 = j^2 = k^2 = ijk = -1$, and $w, x, y, z \in \mathbb{R}$.

A quaternion can be expressed in ordered basis $\{1, i, j, k\}$ as a 4-dimensional vector. For the purposes of this thesis, quaternions are used as a representation of coordinate transformation. Multiplication of 2 quaternions combines the two rotations they represent into a single quaternion, combining the series of transformations.

$$q_O^D = q_R^D \otimes q_O^R \tag{2.9}$$

where the notation q_O^D represents coordinate transformation from origin coordinate system O to destination coordinate system D, and R is a coordinate system whose orientation is known wrt O and D.

It is important to note that quaternion multiplication is not commutative. The symbol $\otimes$ is used to emphasize quaternion multiplication, where the operands are either a quaternion or a vector. In the case of the latter, the vector is rewritten as $\boldsymbol{v} = \begin{bmatrix} 0 & \boldsymbol{v}^T \end{bmatrix}^T$.

Quaternion multiplication operation is a linear transformation whose transformation matrix can be written as:

$$q_R^D \otimes q_O^R = Q(q_R^D)q_O^R = Q'(q_O^R)q_R^D \tag{2.10}$$

where $Q(q) = \begin{bmatrix} w & -x & -y & -z \\ x & w & -z & y \\ y & z & w & -x \\ z & -y & x & w \end{bmatrix}$ and $Q'(q) = \begin{bmatrix} w & -x & -y & -z \\ x & w & z & -y \\ y & -z & w & x \\ z & y & -x & w \end{bmatrix}$.

Coordinate transformation by a quaternion is performed by two quaternion multiplication of the vector.

$$\boldsymbol{v}_D = q_O^D \otimes \boldsymbol{v}_O \otimes (q_O^D)^{-1} \tag{2.11}$$

The inverse quaternion is the reverse coordinate transformation, the conjugate quaternion.

$$(q_O^D)^{-1} = q_D^O = \overline{q}_O^D \tag{2.12}$$

Using Equation 2.10, and the inverse quaternion property, coordinate transformation can be written as:

$$\boldsymbol{v}_D = Q(q_O^D)Q'(q_D^O)\boldsymbol{v}_O \tag{2.13}$$

Transformation matrices can be combined to yield a single transformation matrix that converts vector $\boldsymbol{v}$ from O to D.

$$Q(q)Q'(q) = \begin{bmatrix} q^T q & 0 & 0 & 0 \\ 0 & w^2 + x^2 - y^2 - z^2 & 2xy - 2wz & 2wy + 2xz \\ 0 & 2wz + 2xy & w^2 - x^2 + y^2 - z^2 & 2yz - 2wx \\ 0 & 2xz - 2wy & 2wx + 2yz & w^2 - x^2 - y^2 + z^2 \end{bmatrix}$$

Column and row corresponding to the scalar quaternion component (first row and column) of the matrix are ineffective and can be disposed of, as the scalar component of a vector is always 0. Rewriting the remaining 3x3 matrix yields the DCM correspondent of the quaternion.

$$\boldsymbol{v}_D = C_O^D \boldsymbol{v_O} \tag{2.14}$$

where

$$C_O^D = \begin{bmatrix} 1-2(y^2+z^2) & 2(xy-wz) & 2(wy+xz) \\ 2(wz+xy) & 1-2(x^2+z^2) & 2(yz-wx) \\ 2(xz-wy) & 2(wx+yz) & 1-2(x^2+y^2) \end{bmatrix} \tag{2.15}$$

and $q_O^D = w + xi + yj + zk$ is a unit quaternion.

This shows that the quaternion coordinate transformation operation involves extra computations (Stančin & Tomažič, 2020) equivalent to constructing its DCM correspondent.

Hence, this thesis performs quaternion to DCM conversion given in Equation 2.15 whenever a coordinate transformation is required, as subsequent uses of the same transformation can be done with this DCM to save on computations. Quaternions are used to compose a series of coordinate transformations into a single transformation, as they are lighter both in terms of memory and computations (Stančin & Tomažič, 2020).

While not a necessity, a unit quaternion is usually preferred for attitude representation, as it does not scale the vector it transforms. The quaternion may lose its unit magnitude through attitude updates or corrections and may need to be re-normalized.

2.2.3 Rotation Vector

A Rotation Vector defines an axis and magnitude for a rotation. A vector in $\mathbb{R}^3$ can be rotated by the rotation vector. Applying this to the unit vectors of a coordinate system would create another coordinate system whose orientation with respect to the original is defined by the rotation vector. Conversely, as per Euler's Rotation Theorem, a rotation vector can be defined using the orientation of a coordinate system with respect to another, given that the coordinate systems are orthogonal transformations of each other.

Although a rotation vector can define attitude, it is usually not preferred. In order to

apply the rotation, its linear transformation matrix needs to be computed, which is its DCM correspondent.

Let $\boldsymbol{\phi} = \phi \boldsymbol{u}_{\boldsymbol{\phi}}$ be the rotation vector that rotates unit vectors of the origin (O) coordinate system onto the unit vectors of destination (D) coordinate system, where ϕ is the magnitude and $\boldsymbol{u}_{\boldsymbol{\phi}}$ is the unit direction vector.

Then, the linear transformation matrix from O to D can be written as (Savage, 2007):

$$C_O^D = I - sin(\phi)[\boldsymbol{u}_{\boldsymbol{\phi}}\times] + (1 - cos(\phi))[\boldsymbol{u}_{\boldsymbol{\phi}}\times]^2 \tag{2.16}$$

Note that this is the inverse of the rotation matrix created by the Rodrigues' rotation formula.

Another reason rotation vectors are not used to define attitude is that they are not easily combined. Two successive coordinate transformations described by two rotation vectors are not easily combined into a single rotation vector, making them unfeasible for attitude computations.

Since they are not well suited for either coordinate transformation or attitude computation, they are usually not used for these purposes. They best serve to convert simultaneous orthogonal rotations into a single rotation as shown by Stančin and Tomažič, 2011; that is, to combine gyroscope measurements. The reader can refer to Savage, 2007 for the specifics of this conversion.

The rotation vector can be converted into a quaternion (Savage, 2007):

$$q_D^O = \begin{bmatrix} cos(\frac{\phi}{2}) \\ \boldsymbol{u}_{\boldsymbol{\phi}} sin(\frac{\phi}{2}) \end{bmatrix} \tag{2.17}$$

Note that the quaternion coordinate transformation is in the opposite direction of Equation 2.17.

This thesis uses rotation vectors to convert gyroscope measurements into attitude change quaternions.

2.2.4 Euler Angles

Euler angles describe the coordinate system transformation as three elemental rotations. There are many possible conventions for using the Euler angles. This thesis uses the so-called *3-2-1* (*ZYX*) convention, where the successive rotations are around *Z*, *Y* and *X* axes. These are usually referred as the *yaw* (ψ), *pitch* (θ) and *roll* (ϕ) angles respectively. Euler angles representation is popular due to their physical significance, which are the angles of a classical 3-gimballed platform. This makes them easier to visualize compared to the other representations. However, Euler angle rate equations become indeterminate when the pitch is at $90\,\mathrm{deg}$, which limits their use (Titterton & Weston, 2004).

According to Titterton and Weston, 2004, a quaternion can be constructed from Euler angles using:

$$q_O^D = a + bi + cj + dk \tag{2.18}$$

where

$$a = cos(\frac{\phi}{2})cos(\frac{\theta}{2})cos(\frac{\psi}{2}) + sin(\frac{\phi}{2})sin(\frac{\theta}{2})sin(\frac{\psi}{2})$$

$$b = -sin(\frac{\phi}{2})cos(\frac{\theta}{2})cos(\frac{\psi}{2}) + cos(\frac{\phi}{2})sin(\frac{\theta}{2})sin(\frac{\psi}{2})$$

$$c = -cos(\frac{\phi}{2})sin(\frac{\theta}{2})cos(\frac{\psi}{2}) - sin(\frac{\phi}{2})cos(\frac{\theta}{2})sin(\frac{\psi}{2})$$

$$d = -cos(\frac{\phi}{2})cos(\frac{\theta}{2})sin(\frac{\psi}{2}) + sin(\frac{\phi}{2})sin(\frac{\theta}{2})cos(\frac{\psi}{2})$$

Since Euler angles represent a series of elemental rotations, they can be converted into a DCM by chaining elemental rotation DCMs.

$$C_O^D = C_1(\phi)C_2(\theta)C_3(\psi) \tag{2.19}$$

where $C_1(\cdot)$, $C_2(\cdot)$, and $C_3(\cdot)$ are defined in Equation 2.5, 2.6, and 2.7 respectively.

2.3 Common Coordinate Transformations

Coordinate transformations are used as orientation representations, consequently, attitude representations. Although any of the transformations given in § 2.2 can be used to describe the orientation of a coordinate system with respect to a reference coordinate system, only the DCM versions will be given in the following sections for brevity.

2.3.1 ECI to ECEF

This thesis utilizes a simplified relation between ECI and ECEF where the Earth's rotation is considered to be constant at its nominal rate for a sidereal day.

$$C_i^e = \begin{bmatrix} cos(w_{ie}^e t) & sin(w_{ie}^e t)cos(\gamma) & sin(w_{ie}^e t)sin(\gamma) \\ -sin(w_{ie}^e t) & cos(w_{ie}^e t)cos(\gamma) & cos(w_{ie}^e t)sin(\gamma) \\ 0 & -sin(\gamma) & cos(\gamma) \end{bmatrix} \tag{2.20}$$

where t is time since last Vernal Equinox, w_{ie}^e is the angular rate of the Earth wrt inertial frame written in ECEF, and γ is the Earth's equatorial tilt with respect to its ecliptic plane.

2.3.2 ECI to Perifocal Coordinate System

Perifocal Coordinate System serves as a good stepping stone between ECI and RTN. It needs to be computed only once for a non-perturbed orbit, as it does not depend on the only time-varying orbit parameter which is the *True Anomaly*.

$$C_i^p = C_3(\omega)C_1(i)C_3(\Omega) \tag{2.21}$$

where ω, i, Ω are *Argument of Periapsis*, *Inclination*, and *Right Ascension of the Ascending Node* respectively. Definitions of $C_1(\cdot)$ and $C_3(\cdot)$ are given in Equation 2.5 and 2.7 respectively.

2.3.3 ECI to RTN

RTN Coordinate system can be obtained by a single rotation from the Perifocal Coordinate system.

$$C_p^r = C_3(\theta) \tag{2.22}$$

where θ is the *True Anomaly*, C_p^r is the DCM from Perifocal to RTN, and $C_3(\cdot)$ is given in Equation 2.7.

Then, the DCM from ECI to RTN is:

$$C_i^r = C_p^r C_i^p \tag{2.23}$$

where C_p^r is given in Equation 2.22, and C_i^p is given in Equation 2.21.

2.3.4 Navigation to Body

The navigation coordinate system is the coordinate system chosen as the reference. For ground systems, this is usually a Local Level Coordinate System. For the purposes of this thesis, either ECI or RTN is treated as the Navigation coordinate system. Given the Euler angles from the Navigation to Body, DCM is:

$$C_N^B = C_1(\phi)C_2(\theta)C_3(\psi) \tag{2.24}$$

as given in Equation 2.19.

2.4 Attitude Update and System Model

Attitude update is performed by applying the rotation measured by onboard gyroscopes to the last known attitude of the system in a cumulative manner. This results in error build-up in attitude due to gyroscope sensor errors. In order to mitigate this, attitude error can be estimated using an estimator.

For the purposes of this thesis, Multiplicative Extended Kalman Filter (Markley, 2003) is used. MEKF works with a system model linearized around the last known

attitude state. Thus, system model components linearized around the latest estimate is discussed in this section.

2.4.1 Attitude Update

Let q_b^i be the quaternion that represents the coordinate transformation from the body coordinate system to ECI. A triad of gyroscopes measures the angular displacement in the body coordinate system. Let the rotation vector corresponding to this angular measurement be $\boldsymbol{\phi}_{b_k}^{b_{k+1}}$.

Using Equation 2.17, quaternion that describes the angular displacement, $q_{b_{k+1}}^{b_k}$ can be calculated.

Then, angular displacement can be combined with the prior attitude quaternion to get the updated attitude quaternion.

$$q_{b_{k+1}}^i = q_{b_k}^i \otimes q_{b_{k+1}}^{b_k} \tag{2.25}$$

2.4.2 Attitude Error Model

Dynamics for the states that represent the attitude error will be derived in this section. Let $q_{\hat{b}}^i$ be the latest attitude, where $\hat{}$ denotes the parameters that contain errors. Errors can be separated from the attitude:

$$q_{\hat{b}}^i = q_b^i \otimes q_{\hat{b}}^b \tag{2.26}$$

where $\hat{b}$ is the computed body coordinate system which contains the attitude errors and b is the truth body coordinate system.

According to Savage, 2007, the time rate of change of the attitude quaternion can be written as:

$$\dot{q}_{\hat{b}}^i = \frac{1}{2} q_{\hat{b}}^i \otimes \hat{\boldsymbol{\omega}}_{ib}^b \tag{2.27}$$

with $\hat{\boldsymbol{\omega}}_{ib}^{b} = \boldsymbol{\omega}_{ib}^{b} + \delta\boldsymbol{\omega}$ where $\boldsymbol{\omega}$ is the gyroscope measurements and $\delta\boldsymbol{\omega}$ is the combined gyroscope errors.

Separating truth and error terms:

$$\dot{q}_{b}^{i} \otimes q_{\hat{b}}^{b} + q_{b}^{i} \otimes \dot{q}_{\hat{b}}^{b} = \frac{1}{2} q_{b}^{i} \otimes q_{\hat{b}}^{b} \otimes (\boldsymbol{\omega}_{ib}^{b} + \delta\boldsymbol{\omega}) \tag{2.28}$$

Re-arranging the equation to leave the time rate of change of the attitude error on one side:

$$q_{b}^{i} \otimes \dot{q}_{\hat{b}}^{b} = \frac{1}{2} q_{\hat{b}}^{i} \otimes \delta\boldsymbol{\omega} + \frac{1}{2} q_{b}^{i} \otimes (q_{\hat{b}}^{b} \otimes \boldsymbol{\omega}_{ib}^{b} - \boldsymbol{\omega}_{ib}^{b} \otimes q_{\hat{b}}^{b}) \tag{2.29}$$

Rewriting using Equation 2.10:

$$q_{b}^{i} \otimes \dot{q}_{\hat{b}}^{b} = \frac{1}{2} q_{\hat{b}}^{i} \otimes \delta\boldsymbol{\omega} + \frac{1}{2} q_{b}^{i} \otimes (Q'(\boldsymbol{\omega}_{ib}^{b}) - Q(\boldsymbol{\omega}_{ib}^{b})) q_{\hat{b}}^{b} \tag{2.30}$$

Multiplying both sides from left with q_{i}^{b} to cancel out rotations.

$$\dot{q}_{\hat{b}}^{b} = \frac{1}{2} q_{\hat{b}}^{b} \otimes \delta\boldsymbol{\omega} + \frac{1}{2} (Q'(\boldsymbol{\omega}_{ib}^{b}) - Q(\boldsymbol{\omega}_{ib}^{b})) q_{\hat{b}}^{b} \tag{2.31}$$

Note that:

$$Q'(\boldsymbol{\omega}_{ib}^{b}) - Q(\boldsymbol{\omega}_{ib}^{b}) = -2 \begin{bmatrix} 0 & \mathbf{0}^{T} \\ \mathbf{0} & \boldsymbol{\omega}_{ib}^{b} \times \end{bmatrix} \tag{2.32}$$

where $\mathbf{0}$ is the zero vector and $\times$ is the skew-symmetric operator.

Let attitude error be the rotation vector $\delta\boldsymbol{\phi}$. For a small attitude error:

$$q_{\hat{b}}^{b} \approx \begin{bmatrix} 1 \\ \delta\boldsymbol{\phi}/2 \end{bmatrix} \tag{2.33}$$

Then its time rate of change is:

$$\dot{q}_b^b \approx \begin{bmatrix} 0 \\ \delta\dot{\boldsymbol{\phi}}/2 \end{bmatrix} \tag{2.34}$$

Substituting Equation 2.32, 2.33 and 2.34 into the Equation 2.31:

$$\begin{bmatrix} 0 \\ \delta\dot{\boldsymbol{\phi}}/2 \end{bmatrix} \approx \frac{1}{2}\left(I + \begin{bmatrix} 0 & -\delta\boldsymbol{\phi}^T \\ \delta\boldsymbol{\phi} & \delta\boldsymbol{\phi}\times \end{bmatrix} \right) \begin{bmatrix} 0 \\ \delta\boldsymbol{\omega} \end{bmatrix} - \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \boldsymbol{\omega}_{ib}^b\times \end{bmatrix} \begin{bmatrix} 1 \\ \delta\boldsymbol{\phi}/2 \end{bmatrix} \tag{2.35}$$

Ignoring the second order terms formed by multiplication of components of $\delta\boldsymbol{\phi}$ and $\delta\boldsymbol{\omega}$, and dropping the first row of the Equation 2.35 as it contains no information:

$$\delta\dot{\boldsymbol{\phi}} \approx \delta\boldsymbol{\omega} - \left[\boldsymbol{\omega}_{ib}^b\times\right] \delta\boldsymbol{\phi} \tag{2.36}$$

where $\delta\boldsymbol{\phi}$ is the attitude error rotation vector and $\delta\boldsymbol{\omega}$ is the combined gyroscope error.

Note that in Equation 2.36, the term $\boldsymbol{\omega}_{ib}^b$ is not known and, in practice, is replaced by the gyroscope measurement $\hat{\boldsymbol{\omega}}_{ib}^b$. Its multiplication with $\delta\boldsymbol{\phi}$ results in second-order terms, which were already neglected.

2.5 Trajectory Generation

For the purposes of this study, trajectory generation was divided into position and attitude generation parts. First, the body position was determined throughout the time interval. Then, the nominal attitude was fixed using the Body position vector to get the satellite to face toward either the Earth or the Sun. Finally, deviations were added to attitude to simulate the disturbances on the body.

2.5.1 Position Generation

In order to determine a 3D orbit completely, 6 independent parameters must be known (Curtis, 2008). Hyperbolic orbits were not considered as a part of this research. Thus, from this point onward, equations are given under the assumption of an elliptical or circular orbit. The Classical(Keplerian) Orbital Elements are :

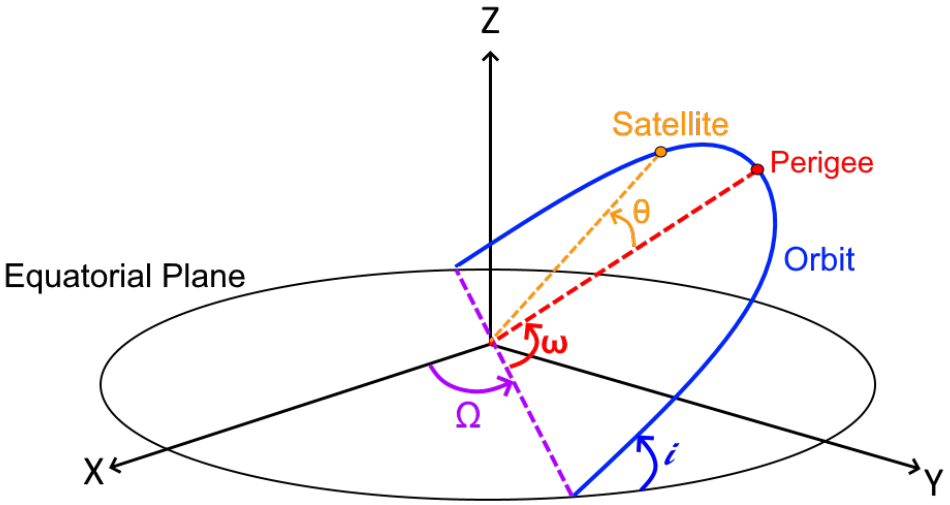

Figure 2.4: Angles defined in the Classical Orbital Elements

- Eccentricity(e)
- Semi-Major Axis(a)
- Inclination(i)
- Right Ascension of the Ascending Node(Ω)
- Argument of Periapsis(ω)
- True Anomaly(θ)

Angles defined in the classical orbital elements can be observed in Figure 2.4. To ensure a LEO orbit, *Eccentricity* and *Semi-Major Axis* can be replaced by *periapsis* (r_p) and *apoapsis* (r_a) radii using Equation 2.37 and Equation 2.38.

$$e = \frac{r_a - r_p}{r_a + r_p} \tag{2.37}$$

$$2a = r_a + r_p \tag{2.38}$$

The 6 chosen orbital elements satisfy the following ranges:

- $100km \leq r_a - R_{Earth} \leq 800km$
- $100km \leq r_p - R_{Earth} \leq r_a - R_{Earth}$
- $-90° < i < 90°$
- $-90° \leq \Omega \leq 90°$
- $-180° \leq \omega \leq 180°$
- $-180° \leq \theta \leq 180°$

where R_{Earth} is the radius of the Earth. For simplicity, it can be taken as the semi-major axis of the WGS84 ellipsoid of the Earth, $R_{Earth} = 6378137.0m$ (WGS84, 2014).

From the given orbital elements, only *True Anomaly* changes with time. Before it is computed, *Mean Anomaly (M_e)* at that time must be calculated using Equation 2.39.

$$M_e = \frac{2\pi}{T}t \tag{2.39}$$

where t is the time since periapsis, and T is the orbit period which can be computed using Equation 2.40.

$$T = \frac{2\pi}{\sqrt{\mu}}a^{\frac{3}{2}} \tag{2.40}$$

Then, the Equation 2.41 can be solved for the *Eccentric Anomaly (E)*.

$$M_e = E - esin(E) \tag{2.41}$$

Finally, Equation 2.42 can be solved for *True Anomaly (θ)*. Range of inverse tangent function is limited to $(-\frac{\pi}{2}, \frac{\pi}{2})$. The domain of the tangent used in the equation is limited to the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ which matches with the range of the inverse tangent function. Thus, the equation can be solved with the tan^{-1} without any problems.

$$tan(\frac{E}{2}) = \sqrt{\frac{1-e}{1+e}} tan(\frac{\theta}{2}) \tag{2.42}$$

Through the use of equations 2.39, 2.41 and 2.42 for all time points along the trajectory, all points on the trajectory become known. Using the 6 orbital elements, the 3-dimensional position can be calculated.

To calculate the 3-dimensional position, one can start with the orbit radius and compute the position vector using coordinate transformations.

Orbit radius at a given *True Anomaly* (θ) can be computed using Equation 2.43.

$$r = a\frac{1-e^2}{1+ecos(\theta)} \tag{2.43}$$

The position of the body in the RTN is $\boldsymbol{r}^r = \begin{bmatrix} r & 0 & 0 \end{bmatrix}^T$ where r is the orbit radius.

Then, using the DCMs defined in Equation 2.23, the position in ECI can be computed as:

$$\boldsymbol{r}^i = C_r^i \boldsymbol{r}^r \tag{2.44}$$

Equation 2.20 can be used for each time instance to convert the position to the ECEF.

$$\boldsymbol{r}^e = C_i^e \boldsymbol{r}^i \tag{2.45}$$

2.5.2 Attitude Generation

For the attitude generation, nadir-pointing and sun-pointing configurations were considered. The satellite's attitude was initialized such that its pointing axis aligns with the initial position vector $\boldsymbol{r}^e$. Then, the satellite can be rotated at each time instance to satisfy this alignment. However, a single vector is insufficient to define the full rotation between each time instance, and another constraint is required. The smallest rotation needed to satisfy the pointing criteria can be used for simplicity. Savage, 2007 derives the DCM via the representations of the same vector with the minimum rotation criteria.

For the satellite to be at a *Sun pointing* attitude at time k, the following equation must be satisfied:

$$\boldsymbol{u}_k^{b_k} = C_i^{b_k} \boldsymbol{u}_k^i \tag{2.46}$$

where $\boldsymbol{u}_k^b$ is the unit pointing vector of the satellite in its body coordinate system, $\boldsymbol{u}_k^i$ is the Sun's direction from the body in ECI, and $C_i^{b_k}$ is the body attitude with respect to ECI, at time k.

Since the unit pointing vector of the satellite is constant, $\boldsymbol{u}_{k+1}^{b_{k+1}} = \boldsymbol{u}_k^{b_k}$ is true. Then, Equation 2.46 can be used to write equal vectors between adjacent sampling times k and $k+1$.

$$C_i^{b_k} \boldsymbol{u}_k^i = C_i^{b_{k+1}} \boldsymbol{u}_{k+1}^i \tag{2.47}$$

Using property of the Equation 2.4, $C_i^{b_{k+1}}$ can be rewritten as $C_{b_k}^{b_{k+1}} C_i^{b_k}$ where $C_{b_k}^{b_{k+1}}$ contains the rotation of the body.

$$C_i^{b_k} \boldsymbol{u}_k^i = C_{b_k}^{b_{k+1}} C_i^{b_k} \boldsymbol{u}_{k+1}^i \tag{2.48}$$

Considering $\boldsymbol{u}_k^{b_k} = C_i^{b_k} \boldsymbol{u}_k^i$, Equation 2.48 can be re-written:

$$\boldsymbol{u}_k^{b_k} = C_{b_k}^{b_{k+1}} \boldsymbol{u}_{k+1}^{b_k} \tag{2.49}$$

In Equation 2.49, $C_{b_k}^{b_{k+1}}$ is defined as the transformation of $\boldsymbol{u}_{k+1}^{b_k}$ into $\boldsymbol{u}_k^{b_k}$. However, this DCM is not unique as this definition is under-constrained. Imposing a minimum rotation constraint along with Equation 2.49, one can solve for $C_{b_k}^{b_{k+1}}$.

According to Savage, 2007, such minimum rotation occurs within the plane formed by the two vectors. So, the direction of rotation vector $\boldsymbol{\phi}$ that represents $C_{b_k}^{b_{k+1}}$:

$$\frac{\boldsymbol{\phi}}{\phi} = \frac{\boldsymbol{u}_k^{b_k} \times \boldsymbol{u}_{k+1}^{b_k}}{||\boldsymbol{u}_k^{b_k} \times \boldsymbol{u}_{k+1}^{b_k}||} \tag{2.50}$$

where $||\cdot||$ is the L^2-Norm operator.

Since $\boldsymbol{u}_k^{b_k}$ and $\boldsymbol{u}_{k+1}^{b_k}$ are unit vectors, $||\boldsymbol{u}_k^{b_k} \times \boldsymbol{u}_{k+1}^{b_k}|| = sin(\phi)$ and $\boldsymbol{u}_k^{b_k} \cdot \boldsymbol{u}_{k+1}^{b_k} = cos(\phi)$ where $\cdot$ is the dot product operator. Then,

$$\phi = tan^{-1}(\frac{sin(\phi)}{cos(\phi)}) \tag{2.51}$$

Finally, plugging all back into the Equation 2.50, the rotation vector corresponding to the minimum required rotation to keep the satellite pointing in the desired direction is:

$$\boldsymbol{\phi} = \frac{\boldsymbol{u}_k^{b_k} \times \boldsymbol{u}_{k+1}^{b_k}}{||\boldsymbol{u}_k^{b_k} \times \boldsymbol{u}_{k+1}^{b_k}||} tan^{-1}(\frac{||\boldsymbol{u}_k^{b_k} \times \boldsymbol{u}_{k+1}^{b_k}||}{\boldsymbol{u}_k^{b_k} \cdot \boldsymbol{u}_{k+1}^{b_k}}) \tag{2.52}$$

$\boldsymbol{\phi}$ in Equation 2.52 can be used with Equation 2.16 to find $C_{b_k}^{b_{k+1}}$. Performing this operation for every consequent time generates the truth attitude by vector pointing. This method is used with the sun's direction to create *sun pointing* portions of the attitude.

Nadir pointing portion is strictly defined by the angle with respect to the RTN. Choosing $-X$ as the Earth pointing direction of the body coordinate system, Euler angles from RTN to the body can be set to $\mathbf{0}$, that is:

$$C_{r_k}^{b_k} = I \tag{2.53}$$

where k is the time instance, and I is the identity matrix.

Then, Equation 2.23 can be used to generate attitude at time k:

$$C_i^{b_k} = C_i^{r_k} \tag{2.54}$$

Finally, attitude deviations can be added on top.

The figures below show an example of an attitude generated using methods described in this section.

Figure 2.5 contains attitude with respect to the ECI, described in terms of Euler angles. Sections not shaded with gray contain the attitude for the *Sun pointing* attitude mode. The pitch and yaw angles are fixed such that the +X axis of the body faces toward the Sun. Deviations in this attitude can be seen clearly in the close-up.

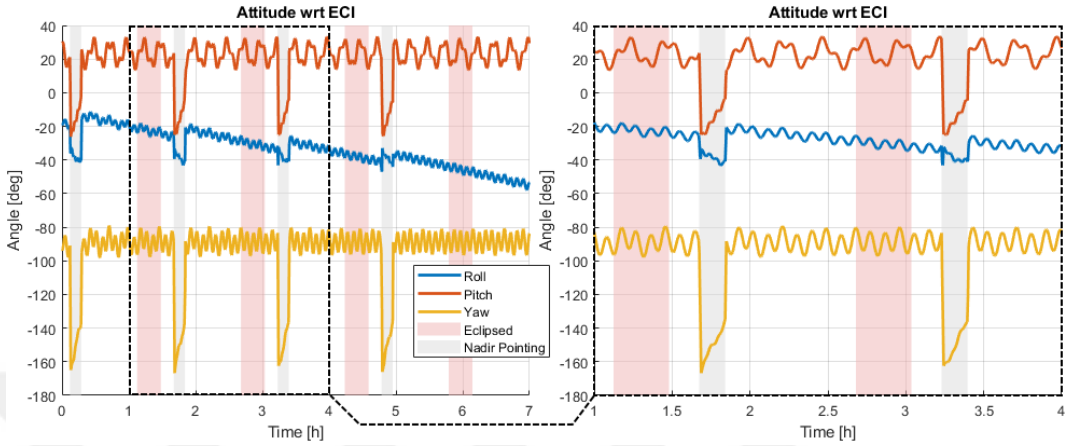

Figure 2.5: Example Attitude with respect to ECI

Figure 2.6 contains the attitude with respect to the RTN. Gray shaded sections contain the *Nadir pointing* attitude mode. Euler angles are all set to 0, which fixes the body coordinate system to the RTN. This results in -X axis facing towards the Earth. For this attitude mode, smaller deviations are created compared to the *Sun pointing* mode. These deviations may be observed in the close-up.

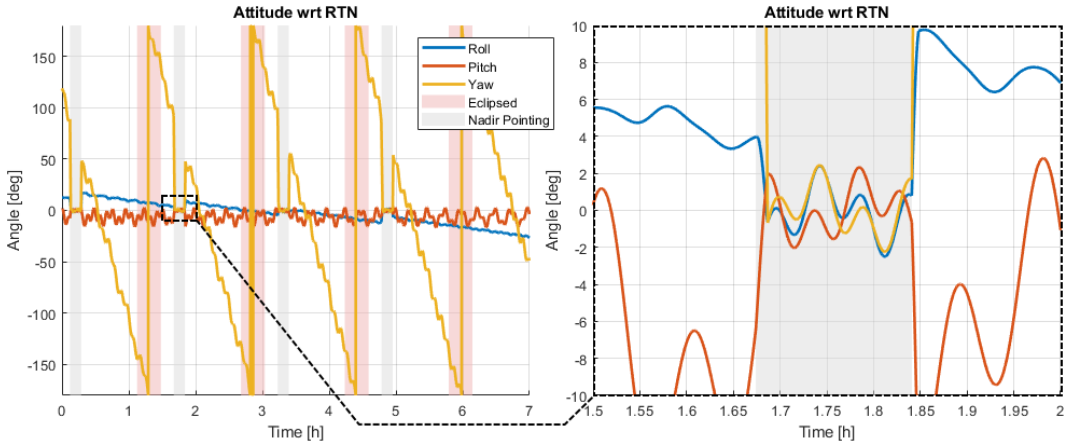

Figure 2.6: Example Attitude with respect to RTN

Sensor data generation for the onboard (a triad of) magnetometer, sun sensor, and

gyroscope on the described trajectory can be considered separately.

2.6 Truth Data Generation

This section explains methods for error-free sensor data generation. Generated trajectory, a reference magnetic model, and the Sun position were used to generate gyroscope, magnetometer, and sun sensor data. These data represent error-free measurements along the trajectory.

2.6.1 Gyroscope

Gyroscopes either measure an instantaneous angular rate or angle change within its measurement interval. For the purposes of this thesis, the two were considered identical, except for the time interval term. Also, the gyroscope measurements were deemed coning corrected.

Given the attitude at each time instance, the angle change required for the transition can be calculated by computing the delta attitude.

$$q_{b_k}^{b_{k+1}} = q_i^{b_{k+1}} \otimes q_{b_k}^i \tag{2.55}$$

where $q_i^{b_k}$ is the attitude at time k.

Rotation vector $\boldsymbol{\phi}$ corresponding to the $q_{b_k}^{b_{k+1}}$ can be calculated as follows:

Let $q_{b_k}^{b_{k+1}} = w + xi + yj + zk$ where i, j and k are the complex numbers. The magnitude of the rotation vector is:

$$\phi = 2tan^{-1}(\frac{\sqrt{x^2+y^2+z^2}}{w}) \tag{2.56}$$

Then, the rotation vector is:

$$\boldsymbol{\phi} = \frac{\phi}{sin(\frac{\phi}{2})} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.57}$$

For simplicity, a constant angular rate can be assumed between k and $k+1$. Then, the error-free mean angular rate from time k to $k+1$ is:

$$\boldsymbol{\omega}_k = \boldsymbol{\phi}_k / \Delta t \tag{2.58}$$

where Δt is the gyroscope measurement interval.

2.6.2 Magnetometer

Magnetometers in satellite missions are used to measure the Earth's magnetic field. This vector measurement and its known reference value can be utilized to obtain information about the satellite's orientation. Having generated trajectory data in the form of position and attitude with respect to the ECEF, one can generate the expected magnetic field using a model for it. World Magnetic Model (WMM) (Chulliat et al., 2020) is a well-known model that can be used for this purpose. It is renewed every 5 years with the most up-to-date measurements. The WMM is a spherical harmonical model of the Earth's magnetic field at its valid date and its rate of change. These coefficients for the magnetic field and their time derivatives can be used to find the magnetic field at a given location.

In § 2.5, satellite position at each time instance was created in ECEF. Using the transformation algorithm provided in Savage, 2007, their (geodetic) latitude, longitude, and altitude equivalents can be found. Chulliat et al., 2020 describes the calculation of the magnetic field starting from the geodetic coordinates. The conversion from the geodetic coordinates (λ, ϕ, h) to the geocentric coordinates (λ, ϕ', r) is given in Equation 2.59, where λ is the latitude, ϕ is the longitude in its respective coordinate system, h is the geodetic altitude, and r is the geocentric radius.

$$\begin{aligned} p &= (R_c + h)cos(\phi) \\ z &= (R_c(1-e^2) + h)sin(\phi) \\ r &= \sqrt{p^2 + z^2} \\ \phi' &= sin^{-1}(\frac{z}{r}) \end{aligned} \tag{2.59}$$

where R_c is the radius of curvature of the prime vertical at the latitude ϕ, and e is the Earth's eccentricity.

The radius of curvature, R_c, can be computed by the following set of equations:

$$R_c = \frac{A}{\sqrt{1 - e^2 sin^2(\phi)}} \tag{2.60}$$

where A is the semi-major axis of the WGS84 ellipsoid.

The model contains *main field coefficients* $g_n^m(t_0)$ and $h_n^m(t_0)$ and their change rates, *secular variation coefficients*, $\dot{g}_n^m(t_0)$ and $\dot{h}_n^m(t_0)$, where $t_0 = 2020.0$ is the reference epoch of the model in decimal years. Gauss coefficients for the desired time t, in decimal years, computed as follows:

$$\begin{aligned} g_n^m(t) &= g_n^m(t_0) + (t - t_0)\dot{g}_n^m(t_0) \\ h_n^m(t) &= h_n^m(t_0) + (t - t_0)\dot{h}_n^m(t_0) \end{aligned} \tag{2.61}$$

Finally, the magnetic field components in ECEF can be calculated as:

$$\begin{aligned} X &= -\sum_{n=1}^{12} \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} (g_n^m(t)cos(m\lambda) + h_n^m(t)sin(m\lambda)) \frac{d\breve{P}_n^m(sin(\phi'))}{d\phi'} \\ Y &= \frac{1}{cos(\phi')} \sum_{n=1}^{12} \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} m(g_n^m(t)sin(m\lambda) - h_n^m(t)cos(m\lambda)) \breve{P}_n^m(sin(\phi')) \\ Z &= -\sum_{n=1}^{12} (n+1) \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} (g_n^m(t)cos(m\lambda) + h_n^m(t)sin(m\lambda)) \breve{P}_n^m(sin(\phi')) \end{aligned} \tag{2.62}$$

where a is the geomagnetic reference radius, and $\breve{P}_n^m(\cdot)$ are the Schmidt seminormalized associated Legendre functions. Details regarding these functions can be found in Chulliat et al., 2020.

The components in Equation 2.62 can be combined to find the magnetic field vector in ECEF:

$$\boldsymbol{B}^e = \begin{bmatrix} X & Y & Z \end{bmatrix}^T \tag{2.63}$$

Then, the reference magnetic field vector in ECI can be found by applying the relevant coordinate transformation, whose inverse is defined in Equation 2.20:

$$\boldsymbol{B}^i = q_e^i \otimes \boldsymbol{B}^e \tag{2.64}$$

Using the attitude generated in § 2.5, the error-free magnetometer measurements in the body coordinate system can be calculated.

$$\boldsymbol{B}^b = q_i^b \otimes \boldsymbol{B}^i \tag{2.65}$$

2.6.3 Sun Sensor

In order to calculate the direction of the Sun from the satellite, the position of the satellite and the Sun in ECI can be used. Satellite position is known from the trajectory generation in § 2.5. To calculate the position of the Sun in ECI, a reverse approach can be taken.

Given a simulation start date, it can be converted into a Julian date. Julian day at 0 h Universal Time(UT) can be computed as (Curtis, 2008):

$$J_0 = 367y - INT\left(\frac{7\left(y + INT(\frac{m+9}{12})\right)}{4}\right) + INT(\frac{275m}{9}) + d + 1721013.5 \tag{2.66}$$

where $INT(\cdot)$ rounds towards zero, y, m, and d are integers for the year, month, and day respectively lying in the following ranges:

$$\begin{aligned} 1901 \leq y \leq 2099 \\ 1 \leq m \leq 12 \\ 1 \leq d \leq 31 \end{aligned} \tag{2.67}$$

Then, any other time of the day can be calculated using:

$$JD = J_0 + UT/24 \tag{2.68}$$

Then, the elapsed days can be calculated by the difference from J2000:

$$\Delta JD = JD - J2000 \tag{2.69}$$

where $J2000 = 2451545.0$

This can be converted into centuries by division with 365.25. Then, all orbital elements in Table 2.1(Curtis, 2008) can be propagated in time to the desired date using their change rates. Using the orbital equations provided in § 2.5, the position of the Earth with respect to the Sun in ECI, $\boldsymbol{r}^i_{e\backslash s}$, can be calculated.

Table 2.1: Elements for Earth's orbit around the Sun

Element	J2000 Value	Change per century
Semi-Major Axis [m]	149597887.5	-7.5
Eccentricity	0.0167	-0.00004
Inclination [°]	0.00005	-0.01304
RAAN [°]	-11.26	-5.063
Longitude of Perihelion [°]	102.95	0.333
Mean Longitude [°]	100.46	35999.37

Then, the position of the Sun with respect to the satellite in ECI is:

$$\boldsymbol{r}^i_{s\backslash b} = -\boldsymbol{r}^i_{b\backslash e} - \boldsymbol{r}^i_{e\backslash s} \tag{2.70}$$

where $\boldsymbol{r}^i_{b\backslash e}$ is the position of the body with respect to the Earth in ECI and $\boldsymbol{r}^i_{s\backslash b}$ is the position of the Sun with respect to the satellite body in ECI.

$\boldsymbol{r}^i_{s\backslash b}$ in Equation 2.70 can be normalized to find the direction of the Sun from the satellite, $\boldsymbol{u}^i_{s\backslash b}$. This becomes the sun sensor measurement reference, which can be converted into the body coordinate system using the generated attitude to find the error-free sun sensor measurement.

$$\boldsymbol{r}^b_{s\backslash b} = q^b_i \otimes \boldsymbol{r}^i_{s\backslash b} \tag{2.71}$$

Additionally, whether the Earth blocks the view of the Sun from the satellite must be checked. This can be done by testing the vector $\boldsymbol{r}^i_{s\backslash b}$ for intersection with a sphere representing the Earth. This intersection check can be performed by finding the closest point on the vector $\boldsymbol{r}^i_{s\backslash b}$ to the center of the Earth and comparing its distance with

the radius of the Earth. The intersection threshold can be exaggerated to disregard partially eclipsed conditions as eclipsed.

Alternatively, the intersection can be checked in terms of angles.

Let α be the angle whose opposite edge is Earth's radius in the plane perpendicular to the body position vector and the neighbor edge is the Earth's position vector from the body, and β be the angle between the Sun vector and the Earth vector from the body given in Figure 2.7. Then, $\alpha \geq \beta$ must be true for the Earth to block the view of the Sun. α can be increased by a flat value for additional clearance.

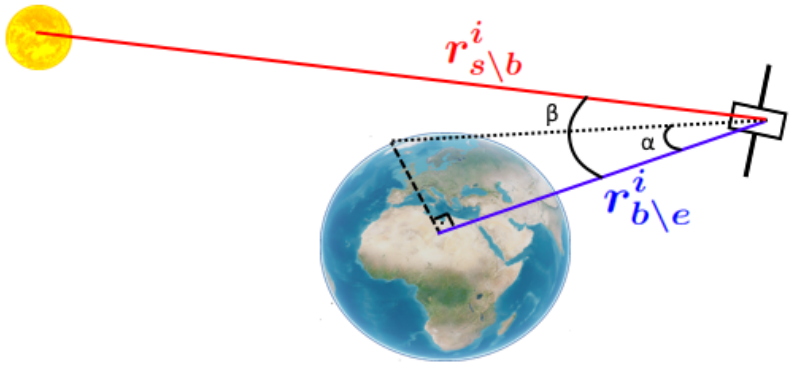

Figure 2.7: Eclipse test by angles

2.7 Sensor Errors

For a realistic simulation, generated error-free sensor data must be corrupted by errors. This section explains the types and generation methods for these errors for each sensor type.

Sensor measurements can generally be considered as the sum of the truth value they are measuring and a combined error. This combined error may have various sources with different underlying mechanisms. These errors are usually simplified into groups according to the needs of the system at hand. Error types considered in this thesis can be found in this section.

$$\Delta y(t) = \sum_i \delta y_i(t) \tag{2.72}$$

where Δy_t is the combined error at time t and $\delta y_{t,i}$ is error caused by error type i at time t.

2.7.1 Gyroscope Errors

For the purposes of this thesis, a gyroscope that would fall into the tactical grade MEMS gyroscope category was considered. Table 2.2 contains the specifications for the generated gyroscope errors.

Table 2.2: Characteristics for the Modeled Gyroscope

Bias Repeatability Error	1°/h
Bias Instability Error	10°/h
Instability Time Constant	2 h
Noise	$0.1°/\sqrt{\text{h}}$
Output Frequency	1 Hz

$$\Delta y_{gyro} = \delta y_{b,rep} + \delta y_{b,ins}(t) + w \tag{2.73}$$

In Equation 2.73, $\delta y_{b,rep}$ is the gyroscope *Bias Repeatability Error*, which is a constant offset independent of the truth measurement. While constant for a single run, it takes different values from run to run. $\delta y_{b,rep}$ is considered to be a zero mean normally distributed random value across simulations, with a standard deviation given in Table 2.2.

On the other hand, $\delta y_{b,ins}(t)$ is the gyroscope *Bias Instability Error*, which is represented with a random-walk trough cumulative sum of zero-mean normally distributed random values. The standard deviation for this sum accumulates at a rate of $10°/h$ in $2h$.

Lastly, w in Equation 2.73 is the gyroscope noise, which is a zero-mean normally distributed random value with its standard deviation given in Table 2.2.

2.7.2 Magnetometer Errors

A magnetometer whose specifications are given in Table 2.3 was considered.

Table 2.3: Characteristics for the Modeled Magnetometer

Bias Error	4000 nT
Orthogonality Error	50 mrad
Scale Factor Error	0.1
Noise	200 nT/$\sqrt{}$Hz
Output Frequency	1 Hz

$$\hat{\boldsymbol{y}}_{mag} = (I + M)^{-1}\boldsymbol{y}_{mag} + \delta\boldsymbol{y}_{bias} + \boldsymbol{w} \tag{2.74}$$

In Equation 2.74, $\delta\boldsymbol{y}_{bias}$ is the constant magnetometer bias which only varies across simulations. It is zero-mean normally distributed across the simulations, with its standard deviation specified in Table 2.3.

The term M is the magnetometer's combined scale factor and misalignment error, and $\boldsymbol{y}$ is the error-free measurement. Scale factor error indicates a scaling error from the sensor's input to its output. Misalignment error occurs when the sensor's sensitive axis does not align with the intended sensing axis. For a triad of sensors, this may happen due to a rotation of the triad block as a whole or individual rotations of the sensing axes. Small misalignment errors can be treated as multiplicative errors that scale with the truth measurements perpendicular to the intended sensing axis. For the purposes of this thesis, only orthogonality error was considered in terms of misalignment, which is shown for the XY plane in Figure 2.8.

$$M = \begin{bmatrix} SF_x & M_{xy} & M_{xz} \\ M_{xy} & SF_y & M_{yz} \\ M_{xz} & M_{yz} & SF_z \end{bmatrix} \tag{2.75}$$

where M_{ab} is the orthogonality error between a and b axes and SF_a is the scale factor error in axis a, with $a = x, y, z$ and $b = x, y, z$.

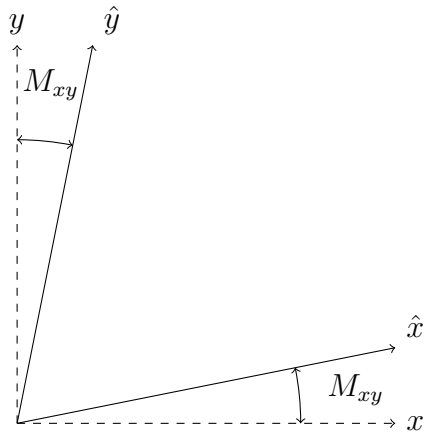

Figure 2.8: XY Orthogonality Error Diagram

Lastly, $\boldsymbol{w}$ in Equation 2.74 is the magnetometer noise vector, whose components are zero-mean normally distributed random values with given standard deviation in Table 2.3.

2.7.3 Sun Sensor Errors

Sun sensor was considered to provide a unit vector in the direction of the Sun. The only significant error considered for the sensor was the noise. Also, sun sensors' inability to provide measurements when eclipsed is considered. Under the eclipse conditions, the generated data contains a zero vector and is unusable.

$$\hat{\boldsymbol{u}}_{sun} = \frac{\boldsymbol{u}_{sun} + \boldsymbol{w}}{||\boldsymbol{u}_{sun} + \boldsymbol{w}||} \tag{2.76}$$

In Equation 2.76, $\boldsymbol{u}_{sun}$ is the error-free sun direction vector and $\boldsymbol{w}$ is the zero mean normally distributed noise on this vector. The noise corrupted vector is normalized at achieve a direction vector. Standard deviation of the components of w is given in Table 2.4

Table 2.4: Characteristics for the Modeled Sun Sensor

Noise	2 mrad/$\sqrt{\text{Hz}}$
Output Frequency	1 Hz
Measurement cut-off by eclipse	

CHAPTER 3

ATTITUDE ESTIMATION AND CONCURRENT SENSOR CALIBRATION

Attitude determination is the process of estimating the current attitude of a system from a set of observations with respect to a reference coordinate system. This process can be categorized into static methods and filtering approaches. Static methods utilize simultaneous, or close enough, measurements to determine the attitude. On the other hand, filtering approaches accumulate a "memory" of observations and fuse this information with the dynamics (mathematical model) of the system (Markley & Crassidis, 2014).

3.1 Attitude Estimation Algorithms

The focus of this thesis is attitude estimation via Kalman Filter. Thus, only the TRIAD method is mentioned as a static method in this section as a means to find an initial attitude estimate for the MEKF.

3.1.1 The TRIAD Algorithm

The TRIaxial Attitude Determination algorithm is derived from the idea that the coordinate transformation matrix, DCM, can be computed using the vectors it transforms.

Let $\boldsymbol{u}$ be a vector whose representation in the reference coordinate system r is $\boldsymbol{u}^r$ and in body coordinate system b is $\boldsymbol{u}^b$

Then, the relation between the different representations of $\boldsymbol{u}$ is:

$$\boldsymbol{u}^b = C_r^b \boldsymbol{u}^r \tag{3.1}$$

where C_r^b is the DCM from the reference to the body coordinate system.

Let $\boldsymbol{v}$ be another such vector, with the constraint $\boldsymbol{u} \cdot \boldsymbol{v} \neq 0$. Then, a third vector can be constructed:

$$\boldsymbol{w} = \boldsymbol{u} \times \boldsymbol{v} \tag{3.2}$$

Since all three vectors are known in both coordinate systems, Equation 3.1 can be modified to contain all three transformations.

$$\begin{bmatrix} \boldsymbol{u}^b & \boldsymbol{v}^b & \boldsymbol{w}^b \end{bmatrix} = C_r^b \begin{bmatrix} \boldsymbol{u}^r & \boldsymbol{v}^r & \boldsymbol{w}^r \end{bmatrix} \tag{3.3}$$

Since $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$ are linearly independent set of vectors, the matrix constructed by them is invertible.

$$C_r^b = \begin{bmatrix} \boldsymbol{u}^b & \boldsymbol{v}^b & \boldsymbol{w}^b \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^r & \boldsymbol{v}^r & \boldsymbol{w}^r \end{bmatrix}^{-1} \tag{3.4}$$

Equation 3.4 is true for error-free measurements of all three vectors. The solution will not yield an orthonormal matrix if the vectors contain errors, such as noise.

The TRIAD method builds on this idea by finding an orthonormal set of vectors before attempting to solve for the DCM.

$$\begin{aligned} \boldsymbol{x} &= \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|} \\ \boldsymbol{y} &= \frac{\boldsymbol{u} \times \boldsymbol{v}}{\|\boldsymbol{u} \times \boldsymbol{v}\|} \\ \boldsymbol{z} &= \boldsymbol{x} \times \boldsymbol{y} \end{aligned} \tag{3.5}$$

Set $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$ is an orthonormal set whose representation is known in both r and b coordinate systems. Note that this method discards information provided by the projection of $\boldsymbol{u}$ onto $\boldsymbol{v}$. Then, the coordinate transformation can be computed as (Markley & Crassidis, 2014):

$$C_r^b = \boldsymbol{x}^b(\boldsymbol{x}^r)^T + \boldsymbol{y}^b(\boldsymbol{y}^r)^T + \boldsymbol{z}^b(\boldsymbol{z}^r)^T \tag{3.6}$$

The TRIAD method is a very light attitude determination algorithm. However, it does not account for sensor errors; it discards some of the information available and utilizes measurements from a single time instance.

3.1.2 Recursive Attitude Estimation

Recursive attitude estimation methods make use of the most recent estimated attitude and iterate on it to improve the results. Kalman Filter is a prevalent approach that utilizes system dynamics to correlate states, "accumulating" information over time. This information is utilized for the estimation process. It is important to note that there is no expanding memory or memory window, but rather the state correlations kept by the KF represents this information.

3.1.2.1 Kalman Filter

Kalman Filter is a recursive optimal estimator for linear systems under white process noise and measurement noise. It offers a way of balancing information from the system dynamics with the information from the observations. This is done by creating a weight matrix via the uncertainty ratio of the system states and the observations.

Operations of a KF are usually divided into two steps. First is the time propagation (prediction) step, which propagates the states through time using known system dynamics.

In this section, the superscript $^-$ is used to denote *a priori* (before observation), and the superscript $^+$ is used to denote *a posteriori* (after observation) information.

The discrete-time propagation on a linear system with no inputs is performed as follows:

$$\boldsymbol{x}_{k+1}^{-} = \Phi_k^{k+1} \boldsymbol{x}_k \tag{3.7}$$

where $\boldsymbol{x}_k$ is the state vector at time k and Φ_k^{k+1} is the state transition matrix from time k to $k+1$.

Corresponding covariance propagation is performed by the following equation:

$$P_{k+1}^{-} = \Phi_k^{k+1} P_k \left(\Phi_k^{k+1}\right)^T + Q \tag{3.8}$$

where P_k is the covariance matrix at time k and Q is the process noise.

The second step is the measurement (observation) update, where external measurements are fed into the system, and an estimation is performed. This is done by feeding back the difference between the estimated output and external measurement to the system model and its covariance. This feedback factor is the *Kalman Gain*:

$$K = P_k^{-} H^T / (H P_k^{-} H^T + R) \tag{3.9}$$

where H is the measurement (output) matrix, R is the measurement noise and K is the *Kalman Gain*.

The difference between the estimated output and external measurement is called *innovation*.

$$\boldsymbol{z} = \boldsymbol{y} - H\boldsymbol{x}^{-} \tag{3.10}$$

where $\boldsymbol{y}$ is the external measurement corresponding to the measurement matrix H and $\boldsymbol{z}$ is the innovation. Note that R in Equation 3.9 corresponds to the noise on the innovation $\boldsymbol{z}$.

Then the states can be updated with the following:

$$\boldsymbol{x}^{+} = \boldsymbol{x}^{-} + K\boldsymbol{z} \tag{3.11}$$

Corresponding covariance update can be performed by:

$$P^{+} = (I - KH)P^{-}(I - KH)^T + KRK^T \tag{3.12}$$

where I is the identity matrix.

Note that Equation 3.12 is given in the non-simplified form, valid for any feedback gain K.

3.1.2.2 Extended Kalman Filter

Extended Kalman Filter is an extension that enables the usage of KF on a non-linear system via linearization of the system around an estimate state. The state propagation shown in Equation 3.7 is replaced by:

$$\boldsymbol{x}_{k+1}^{-} = f(\boldsymbol{x}_k) \tag{3.13}$$

where $f(\cdot)$ is the non-linear state transition function.

The function $f(\cdot)$ can be linearized around the state vector x_k to compute the state transition matrix Φ_k^{k+1}, which can be used in Equation 3.8 for covariance propagation.

For a non-linear system, the measurement matrix H may be a function of the states. Then, Equation 3.10 can be replaced by:

$$\boldsymbol{z} = \boldsymbol{y} - g(\boldsymbol{x}^{-}) \tag{3.14}$$

where $g(\cdot)$ is the non-linear measurement function.

Like the state transition function, the measurement function can be linearized around the state vector to compute the measurement matrix H. Then, Equation 3.9, 3.11, and 3.12 can be used to apply corrections and update the covariance matrix.

3.1.2.3 Multiplicative Extended Kalman Filter

Multiplicative EKF extracts an unconstrained 3-component attitude error from the 4-component non-singular representation of the attitude (Markley, 2003). This derivation is detailed in § 2.4.2. It is important to note that the attitude-related states within the MEKF are attitude errors rather than the full attitude. Following this example, all states in the system are written in terms of errors, constructing a system model of errors.

A diagram representing the order and interaction of operations of the attitude estimation with MEKF and gyroscopes can be seen in Figure 3.1. Gyroscope measurements are integrated to update changes in attitude. This attitude integral builds up errors due

to gyroscope errors. KF is used to estimate both sensor and growing attitude errors. When a measurement from another sensor is available along with its reference, a *KF Cycle* can be processed for the estimation.

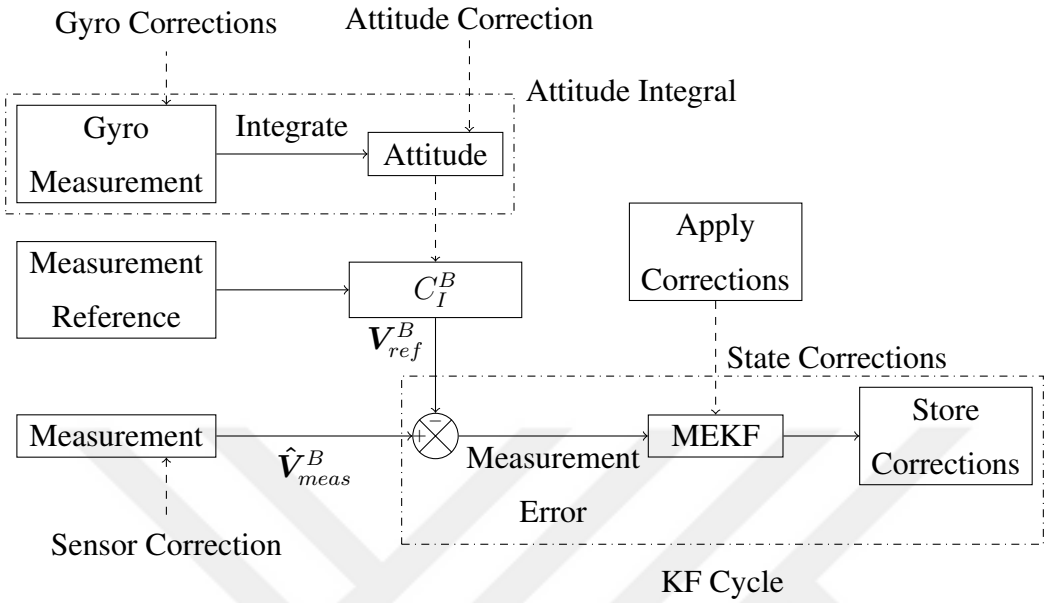

Figure 3.1: Attitude Estimation Diagram using MEKF

Firstly, the KF is propagated to the measurement time using Equation 3.7 and 3.8. Stored corrections from the previous cycle are fed back into the KF system model and sensor corrections, deserializing the KF by adding one cycle delay to the feedback. Then, the measurement reference must be converted to the same coordinate system as the measurements. This is done by utilizing attitude quaternion in its DCM form C_I^B. Measurement error can be calculated via the difference between the transformed measurement reference and the corresponding measurement. This is the observation for the KF, given as $\boldsymbol{y}$ in Equation 3.10. With the observation at hand, the measurement update stage of the KF is performed to estimate the states using Equation 3.9-3.12. Lastly, estimated states are stored for feedback, without applying immediately.

The delayed feedback mechanism is chosen to parallelize *Attitude Integral* and *KF Cycle* without adding extra computations, only at the cost of a little additional memory to store corrections.

CHAPTER 4

INTEGRATED MEASUREMENTS

Commonly, observation for a KF is used as soon as it becomes available. This requires the KF not to run slower than the frequency of the observations. One can reduce this frequency by downsampling the observations at the cost of estimation performance. This thesis suggests manipulation of the observations so that the KF can be slowed down without much effect on the estimation accuracy of the filter.

Figure 4.1 shows the modified MEKF diagram that utilizes the *Integrated Measurements*. Measurement and its coordinate-transformed reference are integrated within a window to produce a slower measurement and reference pair that contains the same amount of information at a reduced frequency. Utilizing the deserialized KF approach, operations regarding the MEKF can be distributed within this window rather than in between two *Attitude Integral* operations. This reduces overall computational load, as the KF operations are performed less frequently and can be distributed over time. At the end of the window, integrals of the measurements are reset, and the next integration window begins. Note that this also means corresponding uncertainties in the MEKF, if any, must also be reset.

Let $\boldsymbol{y_k^{B_k}}$ be the measurement made at time t_k in the body coordinate system B_k. For an interval of N measurement cycles, the integral can be taken in a chosen frame of reference. Choosing B_N as the coordinate system for this purpose, which is the body coordinate system at the end of the integration interval, yields the following:

$$\boldsymbol{Y_N} = \int_{t_0}^{t_N} C_{B_t}^{B_N} \boldsymbol{y}_t^{B_t} \, dt \tag{4.1}$$

where $C_{B_t}^{B_N}$ is the direction cosine matrix (DCM) that relates the body coordinate system at time t to that at time t_N.

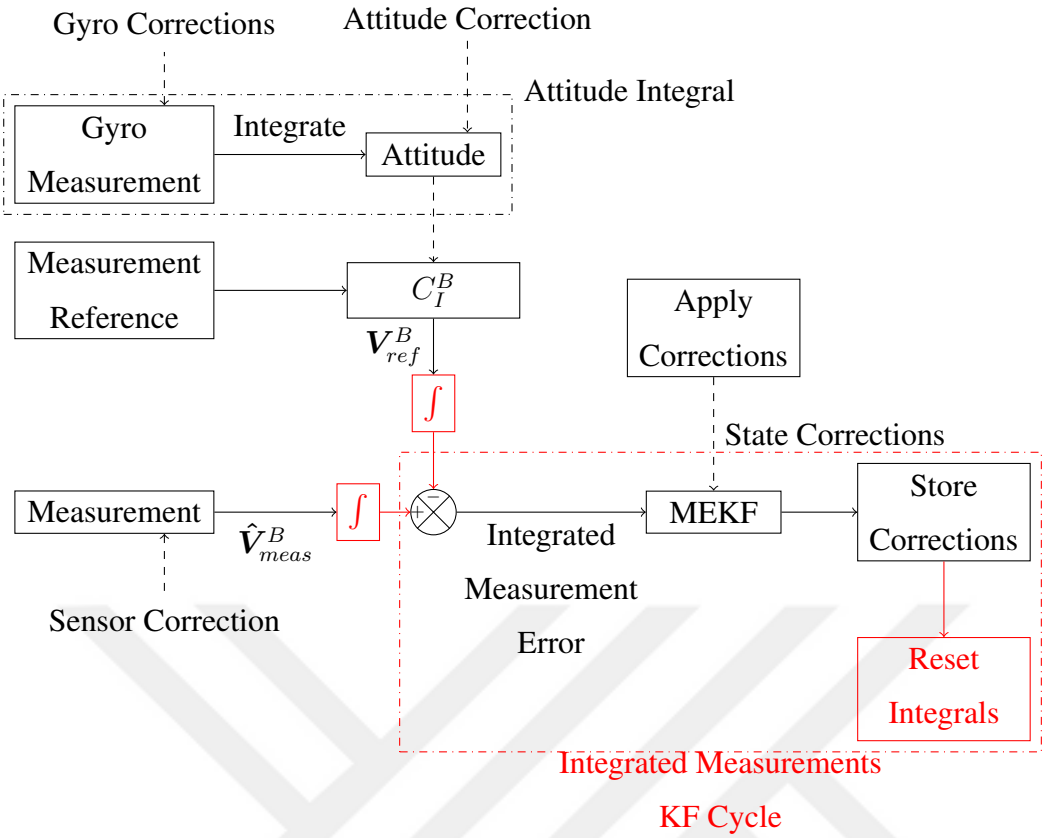

Figure 4.1: Integrated Measurements MEKF Diagram

For discrete (sampled) measurements, the integration can be formulated in discrete form using a midpoint integration technique as follows:

$$\boldsymbol{Y_N} = \Delta t \left(\frac{1}{2} (C_{B_0}^{B_N} \boldsymbol{y}_0^{B_0} + \boldsymbol{y}_N^{B_N}) + \sum_{i=1}^{N-1} C_{B_i}^{B_N} \boldsymbol{y}_i^{B_i} \right) \tag{4.2}$$

Furthermore, since the attitude relation between the beginning and end of the interval is not known until the completion of the interval, the scheme can be converted into a recursive form to enable its computation at every time step rather than a bulk at the end, both saving memory and distributing the computational load evenly.

$$\boldsymbol{Y}_m = C_{m-1}^m \left(\boldsymbol{Y}_{m-1} + 0.5 \boldsymbol{y}_{m-1}^{B_{m-1}} \Delta t \right) + 0.5 \boldsymbol{y}_m^{B_m} \Delta t \tag{4.3}$$

where $\boldsymbol{Y}_0 = \boldsymbol{0}$, Δt is the output interval of the measurement sensor, and $m = 1, 2, ..., N$ is the counter for the steps to reach the integration interval.

This scheme distributes the computation of the discrete integral among the sensor measurement cycles rather than clumping them all at the end. It is essential for a real-

time application for the computations to be evenly distributed in time for a uniform decrease in load.

KF can be designed for this new measurement with little modification. The errors in the measurements $\boldsymbol{y}$ are integrated into $\boldsymbol{Y}$.

$$\hat{\boldsymbol{Y}} = \boldsymbol{Y} + \boldsymbol{\delta Y} \tag{4.4}$$

where $\boldsymbol{\delta Y}$ is the error vector in the integrated measurements.

Expanding the state vector by the integrated measurement error $\boldsymbol{\delta Y}$ can account for this.

$$x_{new} = \begin{bmatrix} x \\ \boldsymbol{\delta Y} \end{bmatrix} \tag{4.5}$$

In a regular observation scheme, innovation for the measurement and reference pair is:

$$\boldsymbol{Z} = \hat{\boldsymbol{V}}^B_{meas} - C^{\hat{B}}_I \boldsymbol{V}^I_{ref} \tag{4.6}$$

Equation 4.6 can be simplified by eliminating truth terms and linearizing it.

$$\boldsymbol{Z} \approx \delta \boldsymbol{V}^B_{meas} + [\boldsymbol{\phi}\times] C^B_I \boldsymbol{V}^I_{ref} \tag{4.7}$$

where $C^{\hat{B}}_B$ is the DCM corresponding to the attitude error, $[\boldsymbol{\phi}]\times$ is the skew-symmetric form of the attitude error vector given in Equation 2.33 and $\boldsymbol{V}^I_{ref}$ is the reference vector in the inertial frame.

For integrated measurements, $\delta \boldsymbol{V}^B_{meas}$ in Equation 4.7 directly corresponds to the new part of the state vector, $\boldsymbol{\delta Y}$. Its contribution to the measurement matrix H becomes identity, and the rest of the contribution comes from the coordinate transformation applied to the measurement reference:

$$\boldsymbol{Z} \approx \delta \boldsymbol{Y}_{meas} + [\boldsymbol{\phi}\times] \boldsymbol{Y}_{ref} \tag{4.8}$$

With this new measurement, both time propagation and measurement update portions of the KF can be slowed down to the frequency of the new measurements. Overall, this results in a reduction of computational load for the system. However, it should be noted that this integration interval has a limit to be considered when increased. The attitude estimation problem can be highly non-linear, assuming a dynamic attitude. The linearized system model in the KF can fail to represent the system properly within the given time interval. This will lead to a discrepancy between the actual errors of the estimation and the standard deviation estimated by the filter, along with the probability distribution of the errors.

The measurement noise R in Equation 3.9 and 3.12 changes with the integration as well. Measurement corrupted by errors and noise can be written as:

$$\hat{V} = V + \delta V + r \tag{4.9}$$

where V is the error-free measurement, δV combined measurement error and r is the white noise with standard deviation σ.

Then, the integrated measurement can be written as:

$$\hat{Y} = Y + \delta Y + \sum_{i=1}^{N} r \tag{4.10}$$

where N is the number of measurements within the integration window.

Since the noise is uncorrelated, it can be treated as the summation of N numbers with the same standard deviation:

$$std\left[\sum_{i=1}^{N} r\right] = \sqrt{N}\sigma \tag{4.11}$$

where $std[\cdot]$ is the standard deviation function.

Measurement noise for the KF measurement update is formed by the Equation 4.11. For a measurement with constant magnitude, the signal-to-noise ratio increases by a factor of $\sqrt{N}$. This is the redeeming factor for the estimation performance, as integrated measurements have fewer observations than the original measurements.

CHAPTER 5

SIMULATION RESULTS

The proposed algorithm was tested against an unmodified approach to compare estimation performance and computational load. To this end, a simulation environment was created. The test case was chosen to be a satellite in a LEO orbit with a triad of gyroscopes, magnetometers, and sun sensors on board.

First, a LEO trajectory was generated for the simulation case. Then, corresponding error-free sensor measurements were generated along this trajectory. Simulated sensor data were created by corrupting the error-free measurements. Finally, the sensor data were run through 2 real-time concurrent attitude estimation and sensor calibration algorithms; a regular MEKF as a baseline and the *Integrated Measurements* approach MEKF. It is important to note that the same sensor data and code structure, except for the modifications performed for the *Integrated Measurements*, were used for both estimation algorithms for a fair comparison.

The estimation results and compute duration for the entire trajectory were collected. Estimated attitude and sensor errors were compared against the ground truth from the trajectory and generated sensor errors. The performance of both algorithms was evaluated and compared in terms of accuracy. Finally, the total compute times were used to gauge the overall change in the computational load.

Simulations were performed over two configurations of sensor error models, using an unmodified MEKF as a baseline and *Integrated Measurements* as the proposed method. The first configuration was a simplified case where magnetometer bias, scale factor, and misalignment errors were ignored in sensor error generation and system modeling. The second configuration included all these terms and represented attitude

estimation with concurrent sensor calibration.

The *Integrated Measurements* approach with an integration window of 10 seconds was compared to the baseline in terms of estimation performance for all states and in terms of total computational load. This section contains these results for a single representative simulation to demonstrate performance in the time domain and the final estimation statistical results for a Monte-Carlo test of 100000 simulations to verify the integrity of statistical assumptions of the estimations. In the sample simulation plots, eclipsed portions are shaded with red and *Nadir pointing* sections are shaded with gray for ease of identification.

5.1 Simple Configuration

Simple configuration is a test for a simplified case where only attitude estimation is required and sufficient. It was meant as grounds for further investigation with the *full configuration*. The results belong to a single sample simulation unless otherwise stated.

In this configuration, the following sensor errors are present and are modeled in the MEKF:

- Gyroscope Bias Repeatability & Instability Error
- Gyroscope Noise Error
- Magnetometer Noise Error
- Sun Sensor Noise Error
- Sun Sensor Eclipse Measurement Cut-off

Figure 5.1 and 5.2 show that both regular KF and integrated measurements KF produce very similar attitude and gyroscope bias estimations for the *simple configuration*. The attitude error goes down to 0.5 mrad and the total gyroscope bias is reduced to $2^\circ/h$, along with their standard deviations when both magnetometer and sun sensor measurements are available.

In this sample case, the satellite is eclipsed approximately every 100 minutes, and sun sensor measurements are lost, the first one starting at about 70th minute. These sections show an increase in both the estimation error and uncertainty, especially for the attitude estimation.

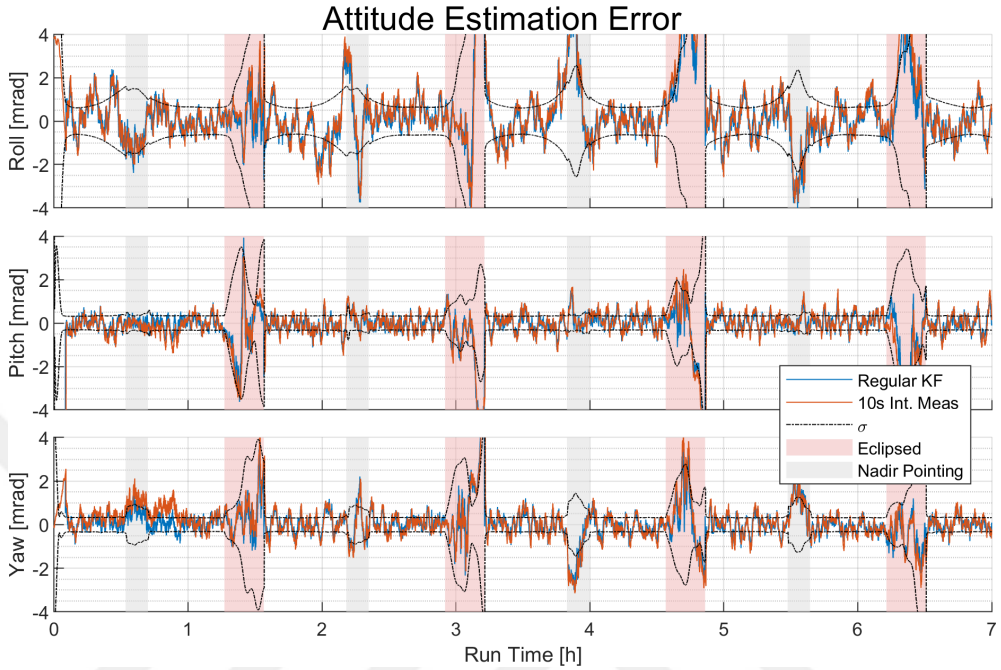

Figure 5.1: Attitude Estimation Errors for *Simple Configuration*

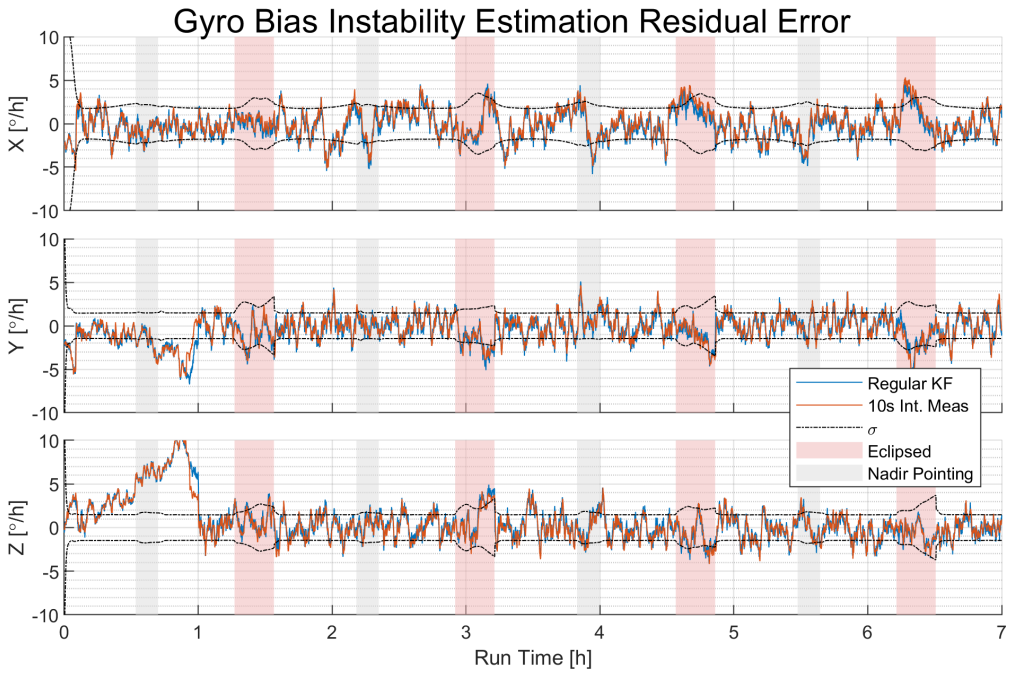

Figure 5.2: Gyroscope Bias Instability Estimation Errors for *Simple Configuration*

Figure 5.3 shows that both approaches result in normally distributed residual attitude errors. The previous figures show that no performance has been lost with the 10 seconds integrated measurements approach.

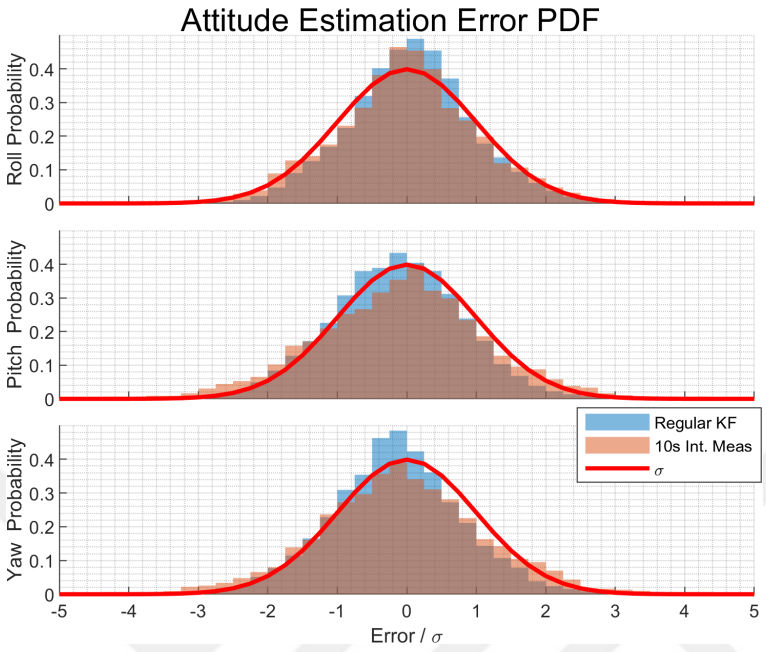

Figure 5.3: Attitude Estimation Error PDFs for *Simple Configuration*

Figure 5.4 and 5.5 show the end of simulation residual attitude and gyroscope bias error distributions across the Monte-Carlo simulation. The former also shows the median standard deviation at the same point for roll, pitch, and yaw while the latter shows the same for the gyroscope bias. They confirm that both the uncertainty and the estimation provided by the proposed approach match the performance of the baseline.

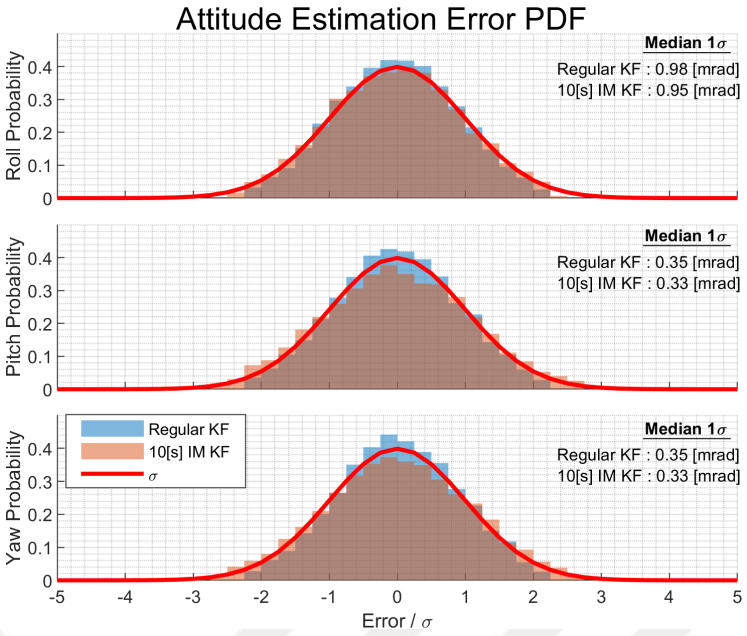

Figure 5.4: MC Final Attitude Estimation Error PDFs for *Simple Configuration*

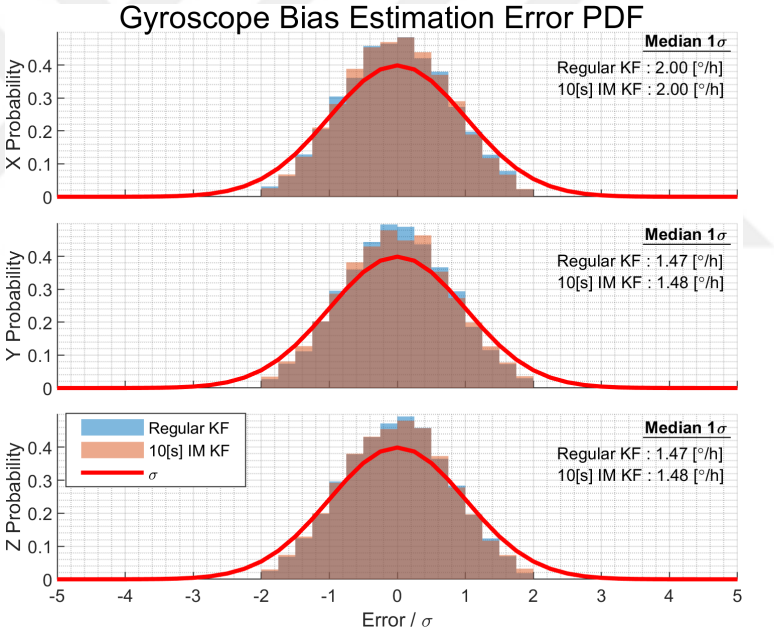

Figure 5.5: MC Final Gyroscope Bias Error PDFs for *Simple Configuration*

On a desktop pc (with Intel i7-8700 CPU, single core affinity), regular KF and 10s Integrated Measurements KF approach was compared using identical input sensor data and system architecture, with the exception of modifications shown in Figure 4.1. In this section, identical estimation performance was demonstrated. The speed-up was measured as the ratio of median compute times. Figure 5.6 shows the histogram of speed-up factors with a median value of 7.1x compared to the baseline.

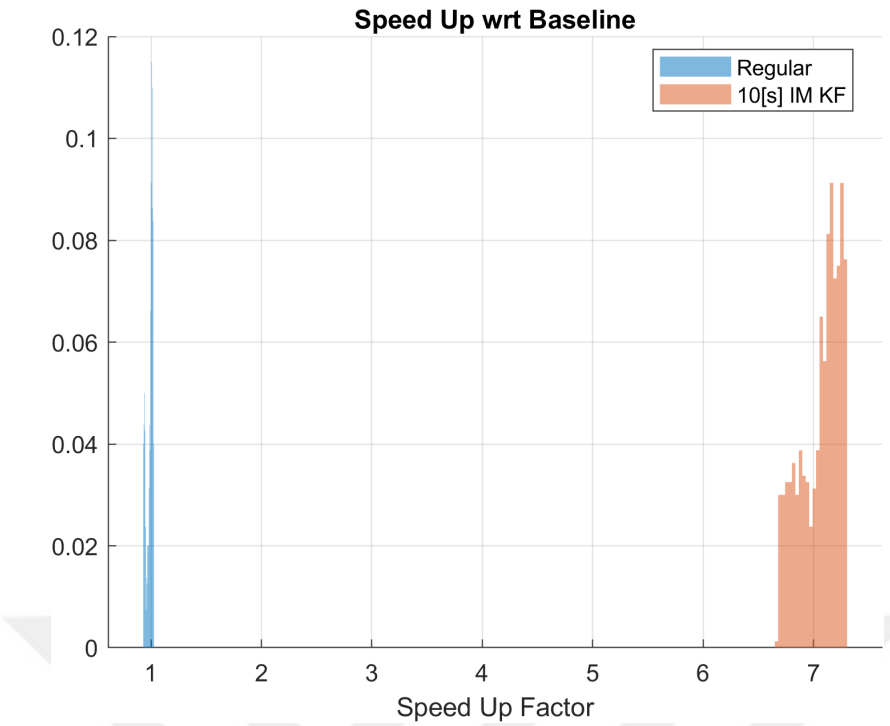

Figure 5.6: MC Simulation Compute Speed Comparison for *Simple Configuration*

5.2 Full Configuration

Full configuration includes magnetometer bias, orthogonality, and scale factor errors in addition to those already existing in the simple configuration. The full list of sensor errors and modeled states are as follows:

- Gyroscope Bias Repeatability & Instability Error
- Gyroscope Noise Error
- Magnetometer Bias Error
- Magnetometer Scale Factor Error
- Magnetometer Orthogonality Error
- Magnetometer Noise Error
- Sun Sensor Noise Error
- Sun Sensor Eclipse Measurement Cut-off

The system model amounts to a total of 15 states for the regular KF approach. However, 3 additional states, in the form of integrated magnetometer error states, are required to implement the integrated measurements approach, totaling 18 states.

In the sample simulation, similar to the *simple configuration*, Figure 5.7 and 5.8 show almost identical attitude and gyroscope bias estimations for both estimators. After the initial stages, overall estimation performance becomes similar to *simple configuration*. This is due to the good estimation of the additional errors, which is demonstrated in the upcoming figures.

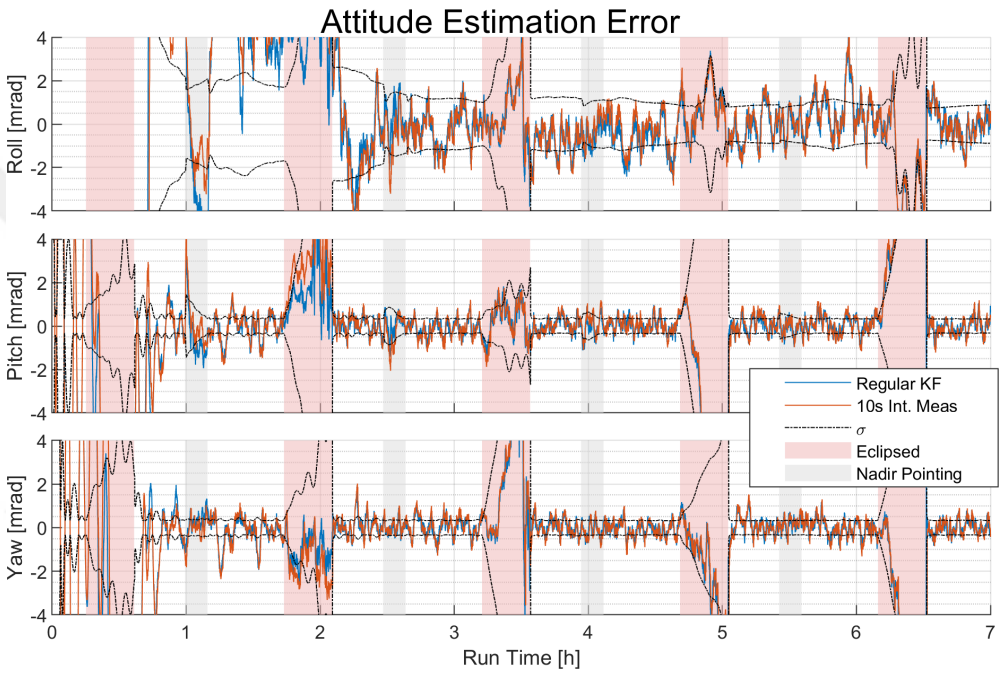

Figure 5.7: Attitude Estimation Error PDFs for *Full Configuration*

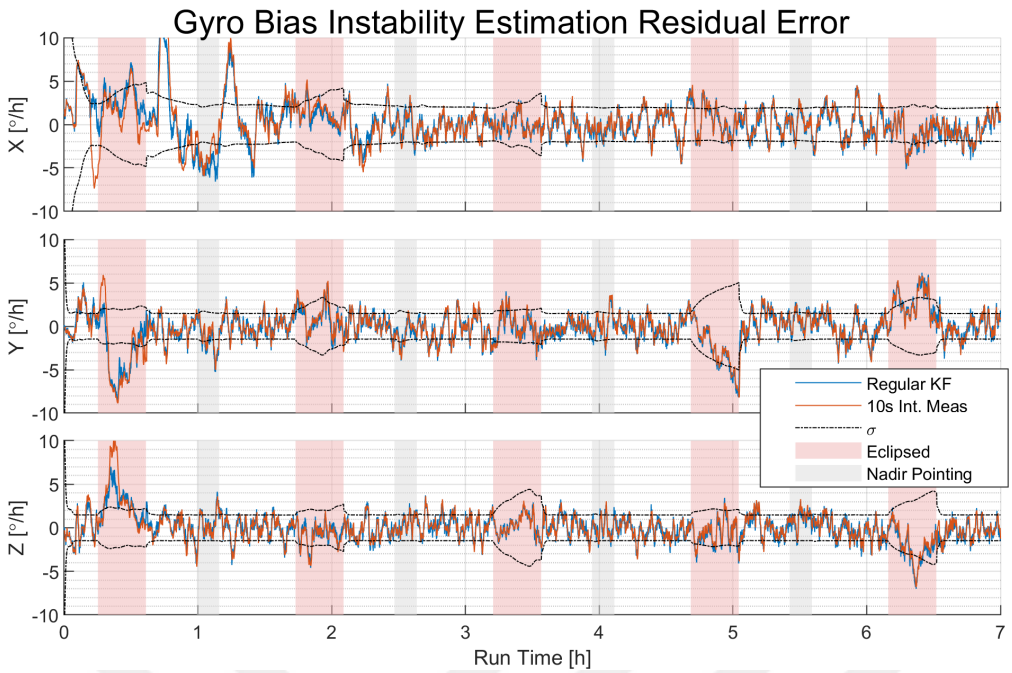

Figure 5.8: Gyroscope Bias Instability Estimation Errors for *Full Configuration*

Figure 5.9, 5.10 and 5.11 show that both estimators produce nearly identical magnetometer error estimates after the initial stages. At the end of the first 2 hours; bias was reduced to 30 nT, orthogonality was reduced to 1 mrad, and scale factor was reduced to 1000 ppm range. For a LEO satellite, these errors would produce a similar magnitude of measurement errors, as the main observable parameter is their sum.

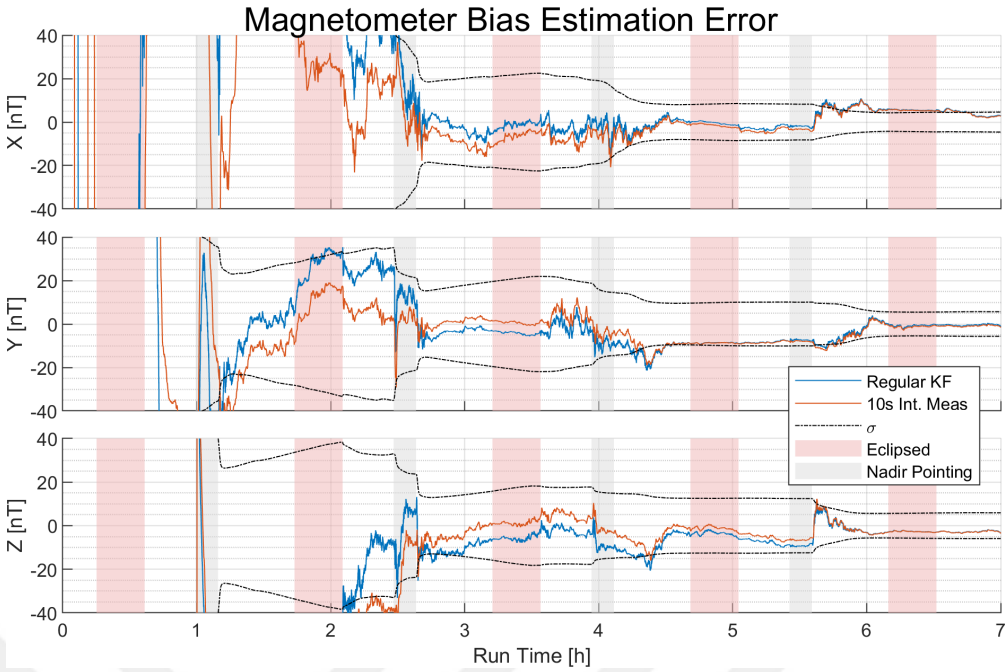

Figure 5.9: Magnetometer Bias Estimation Error for *Full Configuration*

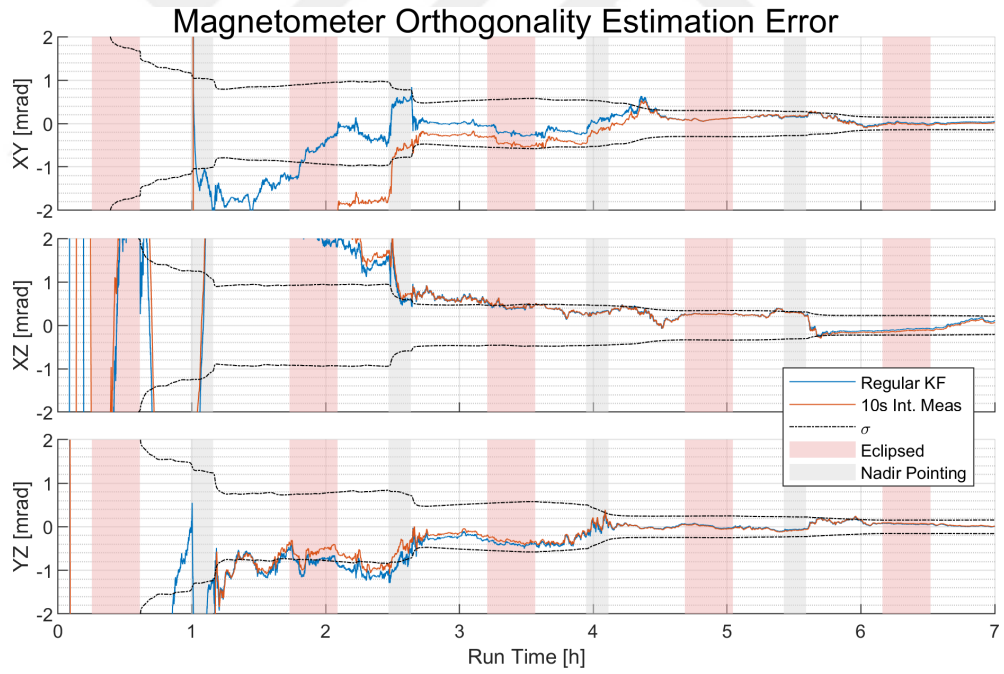

Figure 5.10: Magnetometer Orthogonality Estimation Error for *Full Configuration*

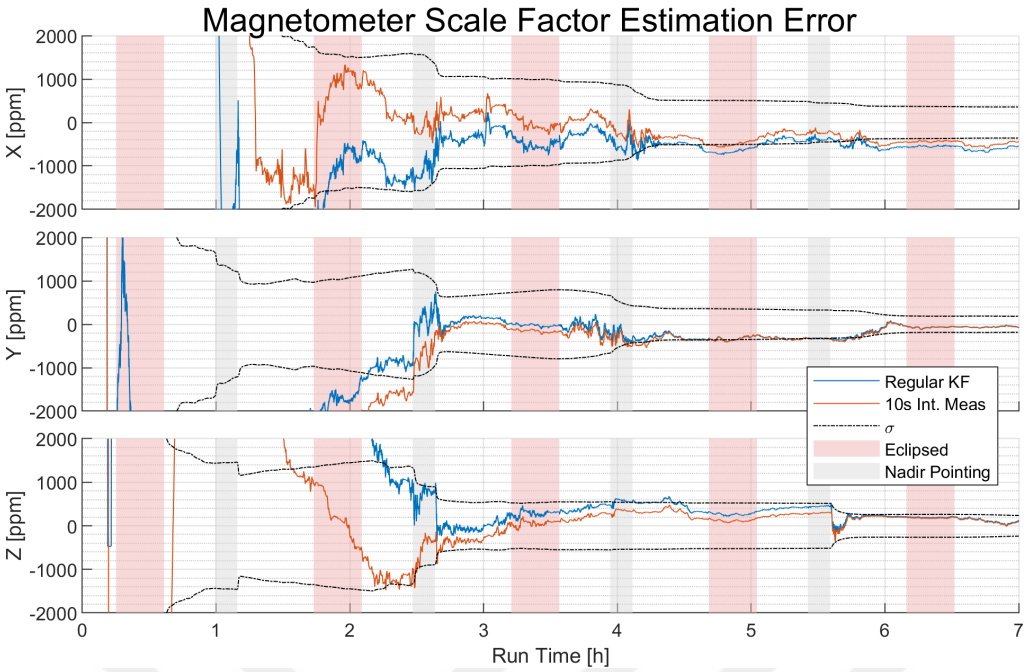

Figure 5.11: Magnetometer Scale Factor Estimation Error for *Full Configuration*

Figure 5.12 shows the attitude estimation error distribution throughout the sample run. The distribution is heavily affected by errors in the initial stages due to low amounts of rotation in the system. Thus, the observability is low and it takes time for the estimation to reduce errors down to the linear region. The performance will be demonstrated via the matured estimations in the MC simulation.

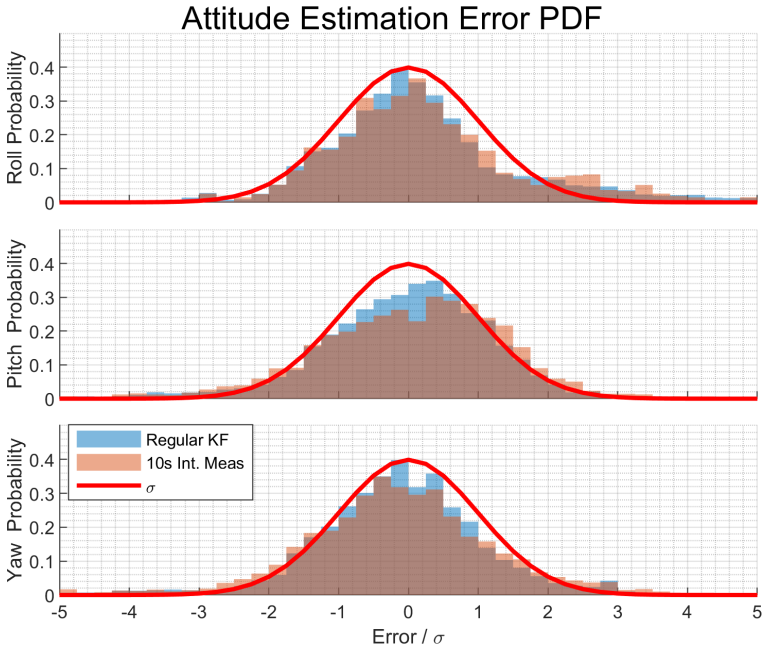

Figure 5.12: Attitude Estimation Error PDFs for *Full Configuration*

Figure 5.13 and 5.14 show the end of simulation residual attitude and gyroscope bias error distributions across the Monte-Carlo simulation for the *full configuration* case. The median uncertainties at these points are also denoted in the figures for reference. They confirm that both the uncertainties and the estimations provided by the proposed approach match those of the baseline. The overall distribution shape closely resembles the normal distribution, confirming that the modified MEKF's statistical properties are intact.

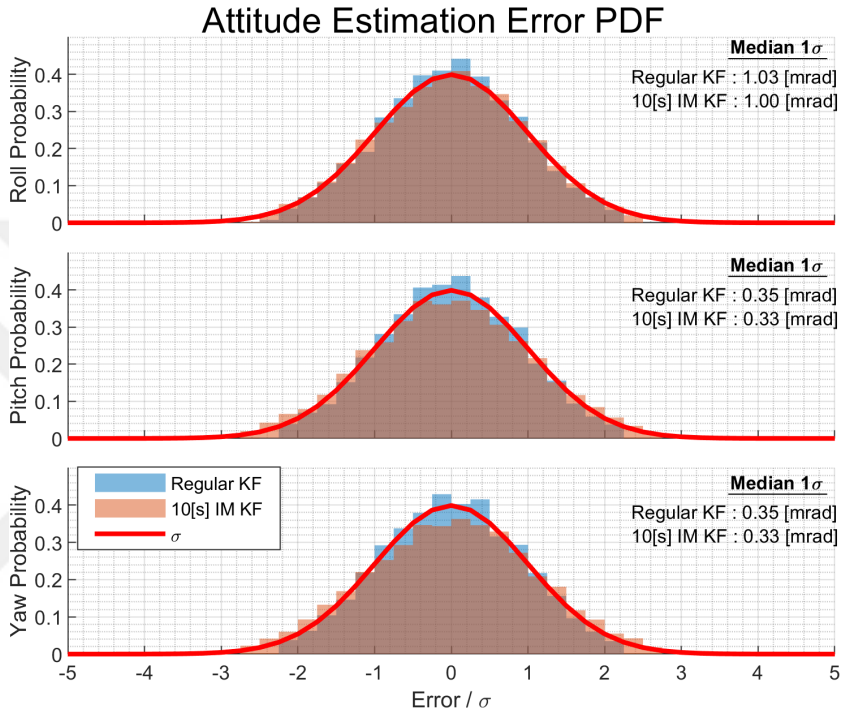

Figure 5.13: MC Final Attitude Estimation Error PDFs for *Full Configuration*

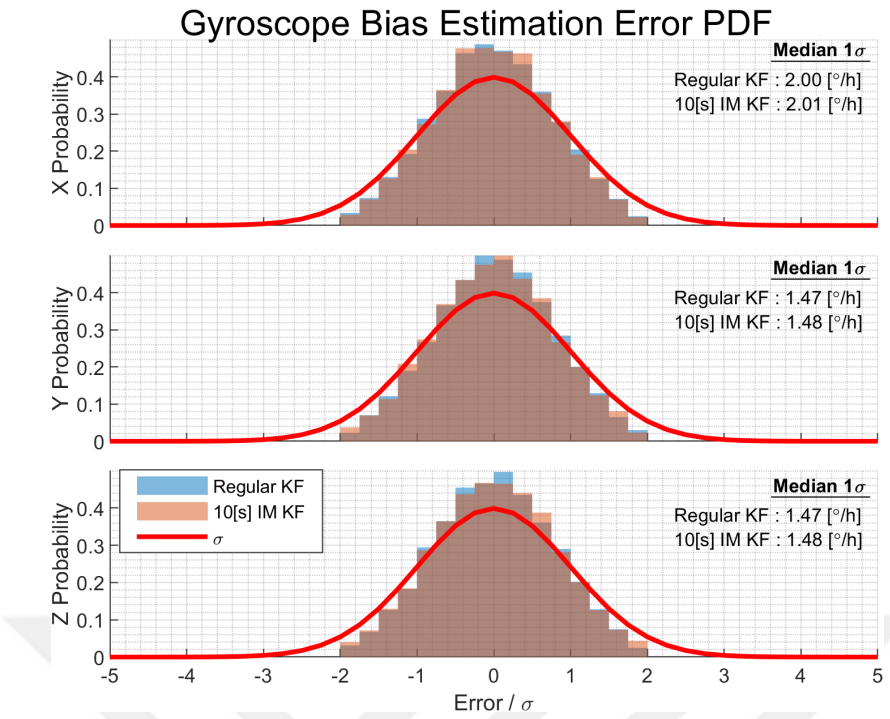

Figure 5.14: MC Final Gyroscope Bias Error PDFs for *Full Configuration*

Figure 5.15, 5.16, and 5.17 show the end of the simulation residual magnetometer bias, orthogonality, and scale factor error distributions across the Monte-Carlo simulation. They demonstrate that both estimators produce nearly identical uncertainties and probability distributions. These distributions are in the shape of normal distributions, confirming the integrity of the estimations.

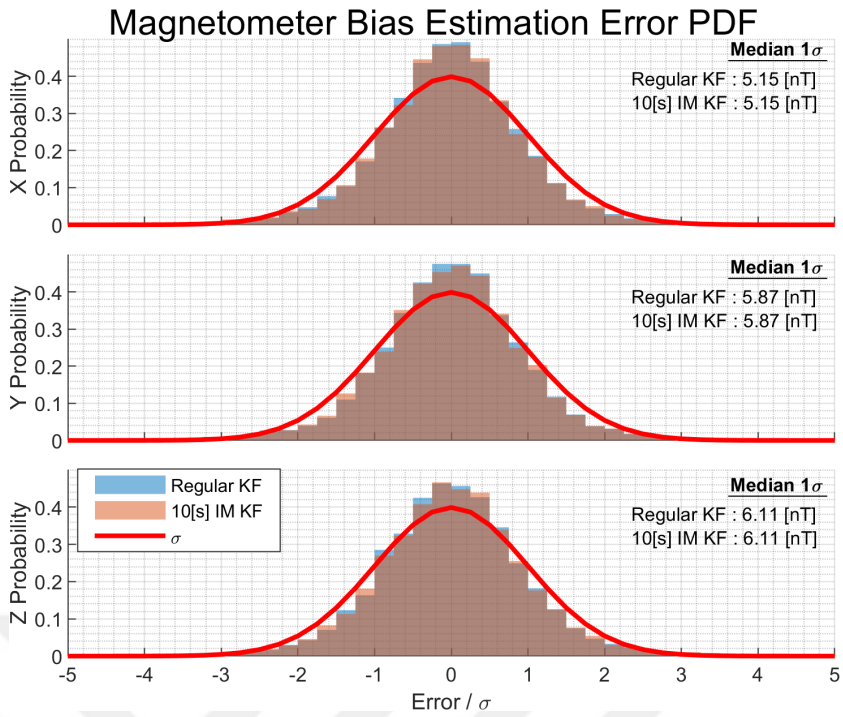

Figure 5.15: MC Final Magnetometer Bias Error PDFs for *Full Configuration*

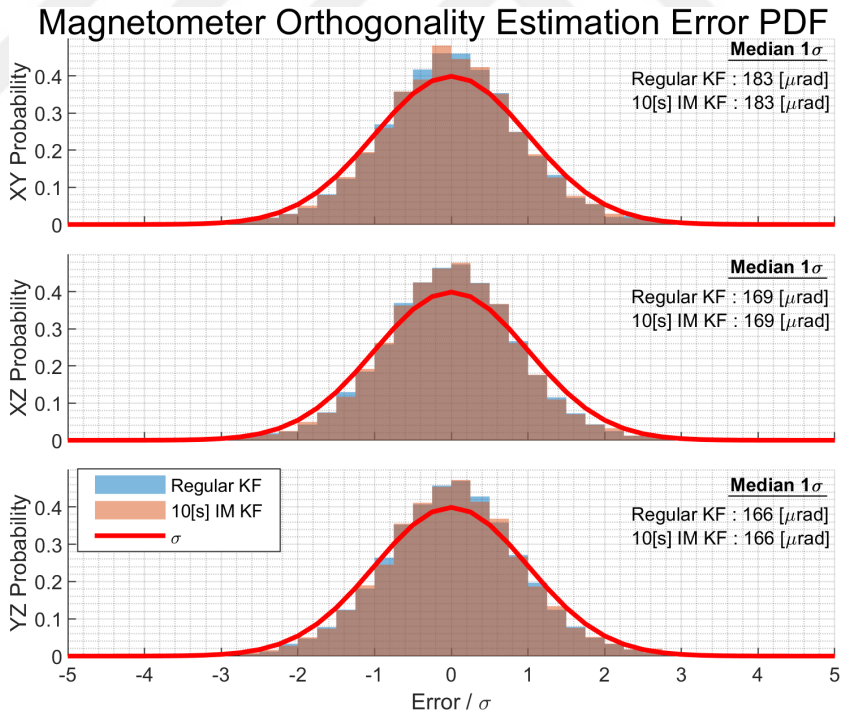

Figure 5.16: MC Final Magnetometer Orthogonality Error PDFs for *Full Configuration*

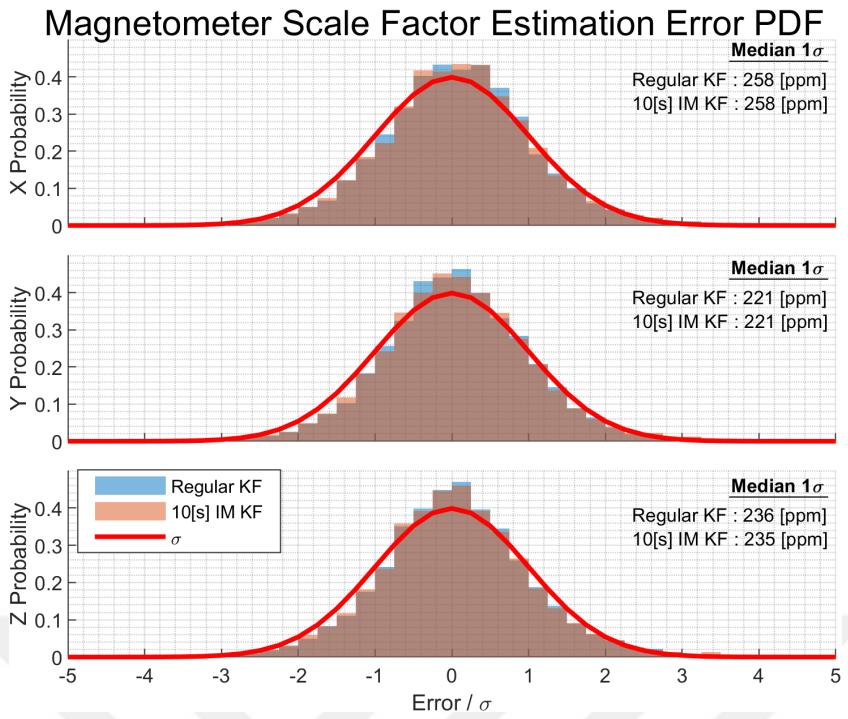

Figure 5.17: MC Final Magnetometer Scale Factor Error PDFs for *Full Configuration*

Figure 5.18 shows probability distributions of final pitch estimations in a Monte-Carlo simulation comparing different integration intervals for the *Integrated Measurements* along with their corresponding speed-ups. The computational load gets lighter as the integration interval is increased, however, the error distribution is adversely affected as a trade-off. Therefore, the interval should be carefully chosen so as not to cause divergence in the system.

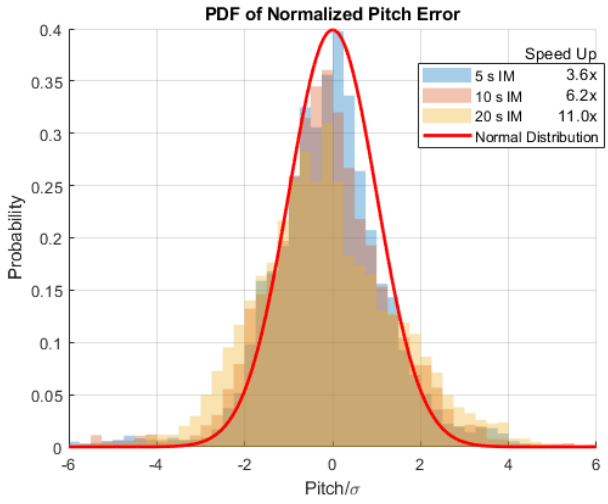

Figure 5.18: PDFs of Pitch Error for Different Integrated Measurements Intervals

Using the aforementioned setup in § 5.1, the speed-up of 10s integrated measurements KF was measured to be 6.2x, as shown in Figure 5.19. It should be noted that this speed-up value includes all the work the attitude algorithm does, including the *attitude integral* and KF operations. As the proposed approach only lightens the load of the latter, the former becomes a limiting factor for speed-up. This and the measurement integration overhead prevent the proposed approach from achieving linear speed-up with the integration window, which would have been 10x for 10s intervals.

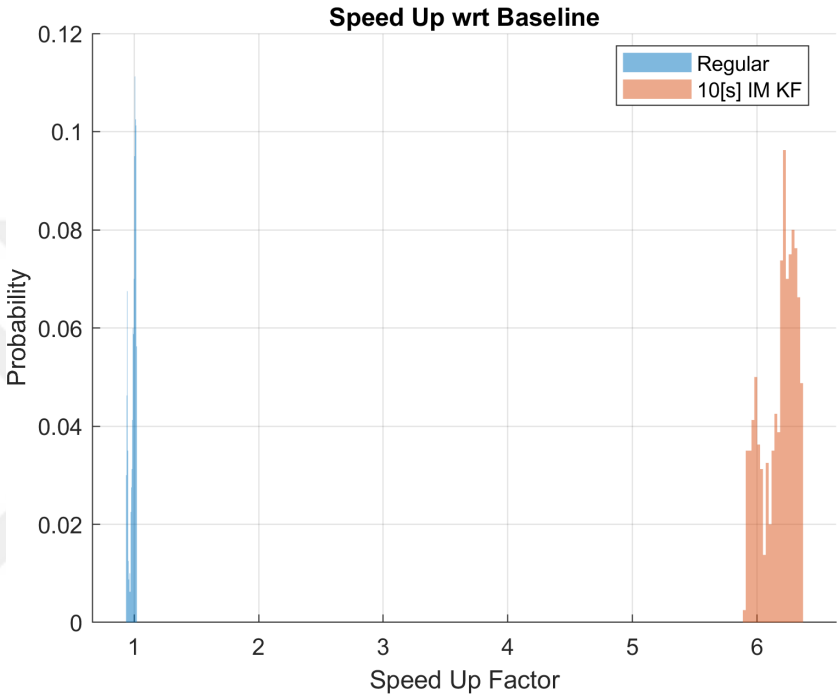

Figure 5.19: MC Simulation Compute Speed Comparison for *Full Configuration*

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, a complementary approach to literature was taken to reduce the computational load of a KF. This approach was applied to a real-time onboard attitude estimation and sensor calibration algorithm for a small satellite. It was tested via the sets of sensor data generated for a variety of LEO trajectories using a Monte-Carlo simulation. The performance of the proposed method was compared to the traditional approach in terms of estimation and computational performance. A lighter overall computational load was demonstrated without a loss in the estimation quality.

However, the integration interval for the method must be carefully chosen as the estimation performance may degrade for bigger windows depending on the system dynamics. Also, increasing the integration window provides diminishing returns since the speed-up is sub-linear.

In conclusion, the *Integrated Measurements* approach was shown to reduce the computational load of a KF without a visible impact on the estimation performance on the satellite attitude estimation problem. Having shown its effectiveness in the satellite attitude estimation problem, the method can be further assessed in other navigation-related problems as well. Moreover, it can be combined with the available methods in the literature for additional improvements.

6.2 Future Work

As a continuation of this research, the following work can be considered:

- Applying the method to computationally heavier estimators like UKF.
- Applying the method to other navigation-related problems
- Investigation of adaptive integration interval selection

REFERENCES

Asundi, S., Latchman, H., & Fitz-Coy, N. (2011).
Attitude estimation for picosatellites with distributed computing platform using murrell's algorithm of the extended kalman filter.
https://doi.org/10.13140/RG.2.1.1821.6162

Asundi, S., Fitz-Coy, N., & Latchman, H. (2021).
Evaluation of murrell's EKF-based attitude estimation algorithm for exploiting multiple attitude sensor configurations. *Sensors*, *21*(19), 6450.
https://doi.org/10.3390/s21196450

Balaguer, P., & Ibeas, A. (2021). Optimal averaging time for improving observer accuracy of stochastic dynamical systems. *ISA Transactions*, *108*, 207–219.
https://doi.org/10.1016/j.isatra.2020.08.039

Benli, D., & Söken, H. E. (2022). Reducing the computational load of attitude estimation for nano-satellites using integrated measurements.
11th Nanosatellite Symposium.
https://nanosat11th.itu.edu.tr/assets/papers/Reducing.pdf

Biswas, S. K., Southwell, B., & Dempster, A. G. (2018). Performance analysis of fast unscented kalman filters for attitude determination.
IFAC-PapersOnLine, *51*(1), 697–701.
https://doi.org/10.1016/j.ifacol.2018.05.117

Bonato, V., Marques, E., & Constantinides, G. A. (2009). A floating-point extended kalman filter implementation for autonomous mobile robots.
Journal of Signal Processing Systems, *56*(1), 41–50.
https://doi.org/10.1007/s11265-008-0257-8

Buchmann, A. (2011).
A brief history of quaternions and of the theory of holomorphic functions of quaternionic variables [Publisher: arXiv Version Number: 1].
https://doi.org/10.48550/ARXIV.1111.6088

Chia, J. W., Tissera, M. S. C., Low, K. S., Goh, S. T., & Xing, Y. T. (2016). A low complexity kalman filter for improving MEMS based gyroscope performance. *2016 IEEE Aerospace Conference*, 1–7. https://doi.org/10.1109/AERO.2016.7500795

Chulliat, A., Alken, P., & Nair, M. (2020). The US/UK world magnetic model for 2020-2025: Technical report [Publisher: National Centers for Environmental Information (U.S.); British Geological Survey]. https://doi.org/10.25923/YTK1-YX35

Curtis, H. D. (2008). *Orbital mechanics for engineering students* (1. ed., reprinted). Elsevier Butterworth Heinemann.

Fatehi, A., & Huang, B. (2017). State estimation and fusion in the presence of integrated measurement. *IEEE Transactions on Instrumentation and Measurement*, *66*(9), 2490–2499. https://doi.org/10.1109/TIM.2017.2701143

Friedberg, S. H., Insel, A. J., & Spence, L. E. (2014). *Linear algebra* (Pearson new international edition. Fourth edition) [OCLC: 856628767]. Pearson.

Garcia, R., Pardal, P., Kuga, H., & Zanardi, M. (2019). Nonlinear filtering for sequential spacecraft attitude estimation with real data: Cubature kalman filter, unscented kalman filter and extended kalman filter. *Advances in Space Research*, *63*(2), 1038–1050. https://doi.org/10.1016/j.asr.2018.10.003

Guivant, J., & Nebot, E. (2001). Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, *17*(3), 242–257. https://doi.org/10.1109/70.938382

Guivant, J. E. (2002). *Efficient simultaneous localization and mapping in large environments* (Doctoral dissertation). The University of Sydney. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=37c4798e4557916019cee1d027e6bba0dc71f8e4

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, *82*(1), 35–45. https://doi.org/10.1115/1.3662552

Markley, F. L. (2003). Attitude error representations for kalman filtering. *Journal of Guidance, Control, and Dynamics*, *26*(2), 311–317. https://doi.org/10.2514/2.5048

Markley, F. L., & Crassidis, J. L. (2014). *Fundamentals of spacecraft attitude determination and control*. Springer New York. https://doi.org/10.1007/978-1-4939-0802-8

Movassagh, S., Fatehi, A., Sedigh, A. K., & Shariati, A. (2023). Kalman filter fusion with smoothing for a process with continuous-time integrated sensor. *IEEE Sensors Journal*, *23*(7), 7279–7287. https://doi.org/10.1109/JSEN.2023.3244659

Pham, M. D., Low, K. S., Goh, S. T., & Chen, S. (2015). Gain-scheduled extended kalman filter for nanosatellite attitude determination system. *IEEE Transactions on Aerospace and Electronic Systems*, *51*(2), 1017–1028. https://doi.org/10.1109/TAES.2014.130204

Qiu, Z., & Guo, L. (2021). Improved cubature kalman filter for spacecraft attitude estimation. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–13. https://doi.org/10.1109/TIM.2020.3041077

Raitoharju, M., & Piché, R. (2019). On computational complexity reduction methods for kalman filter extensions. *IEEE Aerospace and Electronic Systems Magazine*, *34*(10), 2–19. https://doi.org/10.1109/MAES.2019.2927898

Savage, P. (2007). *Strapdown analytics*. Strapdown Associates.

Soken, H. E., & Sakai, S.-i. (2020). Attitude estimation and magnetometer calibration using reconfigurable TRIAD+filtering approach. *Aerospace Science and Technology*, *99*, 105754. https://doi.org/10.1016/j.ast.2020.105754

Stančin, S., & Tomažič, S. (2011). Angle estimation of simultaneous orthogonal rotations from 3d gyroscope measurements. *Sensors*, *11*(9), 8536–8549. https://doi.org/10.3390/s110908536

Stančin, S., & Tomažič, S. (2020). Computationally efficient 3d orientation tracking using gyroscope measurements. *Sensors*, *20*(8), 2240. https://doi.org/10.3390/s20082240

Titterton, D. H., & Weston, J. L. (2004). *Strapdown inertial navigation technology* (2nd ed) [OCLC: ocm56646289]. Institution of Electrical Engineers.

Vinther, K., Jensen, K. F., Larsen, J. A., & Wisniewski, R. (2011). Inexpensive cubesat attitude estimation using quaternions and unscented kalman filtering. *Automatic Control in Aerospace*, *4*. Retrieved August 3, 2023, from https://vbn.aau.dk/ws/portalfiles/portal/71439993/Inexpencive_CubeSat_Attitude_Estimation_Using_Quaternions_and_Unscented_Kalman_Filtering.pdf

WGS84. (2014). National geospatial-intelligence agency (NGA) standardization document - department of defense, world geodetic system 1984, its definition and relationships with local geodetic systems.

Yuan, Z., Song, N., Pan, X., Song, J., & Ma, F. (2021). Fault detection, isolation, and reconstruction for satellite attitude sensors using an adaptive hybrid method. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–12. https://doi.org/10.1109/TIM.2021.3097404