April 2020

Investigation of Navigation Systems for Size, Cost, and Mass Constrained Satellites

Omar Awad
University of South Florida

Investigation of Navigation Systems for Size, Cost, and Mass Constrained Satellites

by

Omar Awad

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Mechanical Engineering
Department of Mechanical Engineering
College of Engineering
University of South Florida

Major Professor: Robert Bishop, Ph.D.
Tansel Yucelen, Ph.D.
Tim Baxter, M.C.S.

Date of Approval:
April 17, 2020

Keywords: NASA, USSOCOM, Kalman Filter,
Cubesat, DARPA

Dedication

The efforts put forth in the completion of this thesis are dedicated to the future student researchers of USF and the Institute of Applied Engineering. My motivation in writing this thesis with as much detail as I could was to provide a strong foundation in spacecraft navigation research for future students to understand and pick up on to enable further advancement in state of the art navigation systems research and technology.

Acknowledgments

First and foremost, I want to thank my parents for supporting me throughout my career. This success could not have been possible without them.

This research was supported by a partnership between USF and SOFWERX/USSOCOM to prove that a team composed of students can design, build, and operate radically inexpensive cubesats given the constraints of our mission and schedules. Our success in building these cubesats could not have been possible without the strong leadership of our project manager, Peter Jorgensen. In addition to our amazing leadership I would like to acknowledge the hard work and persistence of the following people who played significant roles in our project's success: David D'Angelo, Nikolas Soulandros, Alex Hansen, Troy Young, Stephanie Whisman, Steven Gioiosa, and Jeremy Torres.

I would also like to give a special thanks to Tim Baxter for giving me this amazing opportunity to make history at USF in launching our first satellites. Words cannot explain how proud I am as a Tampa born and raised individual to represent my city and university through this historic milestone.

Most importantly, I would like to thank my research advisor and principle investigator of this project, Dr. Bishop, for taking me in as his student and providing me with funding, but much more importantly for his kindness and dedication to teaching me Kalman filtering and navigation the proper way. There were no courses offered on Kalman filtering or navigation and Dr. Bishop spent countless hours outside the classroom teaching me everything I need to know to succeed in this rigorous field, something I will never forget. I am also extremely grateful for the opportunity Dr. Bishop gave me to first author a re-

search paper alongside him and present our work at the American Astronautical Society Guidance, Navigation, and Controls conference in Breckenridge.

Finally, I would like to acknowledge the following mechanical engineering faculty, whom our college is very lucky to have: Ajit Mujumdar, Ahmad Alqasimi, Rasim Guldiken, and Tansel Yucelen.

Dr. Mujumdar supported me through the last two years of my bachelor's and additionally throughout my master's by offering me teaching assistantships for Computer-Aided Design (CAD) with Dr. Alqasimi and Finite Element Analysis (FEA) with him. I strongly believe that these teaching assistantships have shaped me into the person I am and opened up many doors for jobs and internships, specifically the opportunity to build cubesats under the supervision of Dr. Bishop.

I spent the majority of my time as a TA assisting Dr. Alqasimi in CAD and formed a strong friendship and would like to thank him for continuing to choose me as his TA each semester.

Dr. Guldiken supported me and many other graduate students as the department's graduate program advisor and coordinator. While Dr. Guldiken is a fantastic advisor, I would like to thank him specifically for his kindness, hard work, and dedication inside and outside the classroom to education and making the graduate experience enjoyable. I would also like to thank him for supporting my Ph.D. and fellowship applications in both writing me letters of recommendation and helping me write my statement of purpose.

Dr. Yucelen is the reason I am pursuing a more control systems-focused career for my Ph.D. alongside navigation. His passion for controls intrigued me through his linear systems course and inverted pendulum project that was first offered to our class as extra credit. He is dedicated not only to his research but education and I want to thank him for his support. I have never taken a course where a professor was so involved in the course and learning process that he would go out of his way to email the top students individually to notify them of their exam scores.

Table of Contents

List of Tables

List of Figures

Abstract

The feasibility of using radically inexpensive micro-electromechanical system (MEMS) technology for navigation of a nanosatellite is investigated with a focus on attitude estimation. Typically, larger satellites are equipped with star cameras, sun sensors, or Earth horizon sensors for attitude estimation. These sensors can provide very accurate attitude measurements. A nanosatellite is highly size, power, and cost constrained and cannot readily accommodate these sensors. Our mission is to design, build, and operate a radically inexpensive nanosatellite system. While there is no consensus on what constitutes a "radically inexpensive" satellite, our goal is a maximum cost of $10,000 per unit. This precludes the possibility of using costly high-precision attitude sensors. Instead, we employ a MEMS inertial measurement unit (IMU) and magnetometer coupled with global positioning system (GPS) to provide derived attitude measurements given an onboard and up-to-date version of the World Magnetic Model (WMM). This method of deriving attitude measurements will produce relatively inaccurate measurements. Since our sensors, particularly the derived attitude measurements and the IMU, are corrupted with systematic errors and random noise, a multiplicative extended Kalman filter (MEKF) is employed. We augment the state vector with stochastic models of the systematic errors to obtain real-time estimates of the errors as well as the spacecraft state vector. The MEKF design process can be divided into two main areas; modeling and estimation. The satellite system dynamics are modeled using well-known governing equations of motion that represent the position, velocity, and attitude of the satellite to obtain *a priori* state estimates prior to the measurement update. The position, velocity, and attitude are components of the state vector, which in our case we can measure directly. The IMU is used to propagate

the spacecraft position, velocity, and attitude between measurements. Since the IMU is corrupted with systematic errors, such as bias, misalignment, and scale factor errors, the *a priori* estimates obtained during propagation will be degraded. Models of these errors are included in the augmented state to improve the overall accuracy of the navigation system. Since the MEKF is known to be an *ad hoc* algorithm, careful consideration must be taken to ensure that the state estimation error covariance represents reality. This requires very careful filter tuning. A single run simulation provides a glimpse of how well the MEKF is estimating the state vector, however, this is only for a single run and may not be representative of the actual estimation error. Therefore, we perform a Monte Carlo analysis where hundreds of runs are completed and analyzed. This provides insight into how the MEKF will respond in as many different scenarios as possible. What we specifically analyze is if the average sample variance of our Monte Carlo runs closely matches the state estimation error covariance of a single-run simulation. If so, this indicates that the filter state estimation error covariance represents reality and that we can trust the results. Another analysis tool that we employ is the error budget. The error budget allows us to determine which of the error sources contribute the most to the overall uncertainty of our state estimates. The error budget is then used to generate a sensitivity analysis. While the error budget may illuminate the errors contributing the most to the overall uncertainty, the sensitivity analysis points out which of the error sources are most impactful if they are larger than expected. The benefit of these two analysis tools is that we may find that some error sources do not have significant contributions to the overall uncertainty and can be removed from the filter to simplify the structure. After our error budget and sensitivity analysis, we determined that six of these error sources fit this category and can be removed from the filter. These include the IMU scaling, non-orthogonality, and misalignment errors for the gyroscope and accelerometer. Removing them and creating a sub-optimal filter allows faster computation while still achieving very similar results to that of the optimal filter. The final MEKF design provided estimates of the attitude with an accuracy below 5°. This is acceptable for our mission which is a sun pointing attitude within ± 20°. The main source of error for the attitude estimate was the attitude measurement bias which was found to be unobservable. Since the magnetometer is the primary source of data for this derived measurement, bias in the attitude measurement comes from bias in the magnetometer readings. Magnetometer biases are due to hard and soft iron errors, which can be calibrated and minimized beforehand, thus reducing the uncertainty in our attitude bias, but not completely eliminating it. Position measurement bias was also unobservable, however, our mission is sun pointing, therefore precise positioning is not required. We were still able to achieve an uncertainty of ± 1 m when the bias uncertainty itself was ± 1 m and the sensor uncertainty was ± 3 m. The velocity measurement bias was observable, therefore we were able to achieve an uncertainty of under ± 0.05 m/s which is more than a 50% increase in accuracy from the 0.1 m/s accuracy provided by the sensor. The most sensitive error groups were found to be the biases of the position and attitude measurements and the least sensitive groups are the IMU systematic errors.

Chapter 1: Introduction

1.1 Cubesats

Cubesats are a class of nanosatellites that have been widely adopted by many universities and organizations for communications missions. The cubesat standard was invented by California Polytechnic State University and Stanford University and is recognized as "the origin of the new space revolution" [1]. Their cost of production and availability of parts and resources has made the cubesat a viable alternative to larger satellites as technological advancements continue to lead to a decrease in the size of components required for a successful operational satellite, such as solar panels, attitude sensors, power sources, communication boards, etc. The average cost of a nanosatellite is substantially lower than that of a regular satellite. For example, USSOCOM's Prometheus satellite (a 1.5U cubesat) has an estimated reproduction cost of $25,000 [2]. When compared to that of a weather satellite which costs millions of dollars, we discover a substantial difference and one of the many benefits of the cubesat program, along with the relative ease of fabrication and deployment.

Cubesats are classified by their size and can vary from 10 x 10 x 2.5 cm to 10 x 10 x 30 cm. A "U" is used to describe the height of a cubesat as all cubesats have a 10 x 10 cm cross sectional area [3]. Each "U" corresponds to 10 cm in height. Figure 1.1 depicts this form factor. Most satellite missions including cubesat missions can be accessed via the Earth Observation Portal [4].

Figure 1.1: 1U Cubesats (Courtesy of NASA).

1.2 Study Motivation

The motivation and methodologies are discussed in detail before conducting the investigation of our smallsat navigation system.

1.2.1 Mission

Our motivation stems from the challenge to design, build, and operate very small (0.5U) and radically inexpensive smallsats that can perform ground and inter-satellite communication, as well as active attitude determination and control. While there is no consensus on what defines a "radically inexpensive" satellite, we aim for a cost of $10,000 per unit. Since 1U smallsats have a mass limit of 1.33 kg [5], we imposed a limit of 1 kg on our units (there is no standard for 0.5U smallsats yet). Given these size, mass, and cost constraints, attitude determination and control becomes a challenge. While attitude control can be achieved with tri-axial magnetometers (although not full 3-dimensional control), attitude determination is much more challenging.

Typical sensors used for providing attitude measurements or information for attitude determination are star cameras, sun sensors, and Earth horizon sensors. However, these sensors do not fit within our size, mass, or cost constraints. A star camera designed for cubesats is capable of providing attitude measurements well below 0.1° with an average cost of $30,000 [6,7]. An Earth horizon sensor designed for cubesats can provide sub 0.2° accuracy for attitude measurements with an average cost of $15,000 [8]. A sun sensor designed for cubesats can provide sub 0.5° accuracy at an average cost of $3,000 [9,10]. The objective for our attitude determination and control system (ADCS) is to maintain a sun-pointing attitude. Therefore, the high accuracy associated with these sensors is not needed. Our acceptable window of sun-pointing attitude is ± 20°. We aim to drive our attitude accuracy within these bounds to ensure that we can point with confidence to the sun.

1.2.2 Attitude Determination

In order to provide the satellite with an attitude measurement with our cost, size, and power constraints, we consider the available on-board resources. We assume the nanosatellite is equipped with a GPS capable of position and velocity measurements, an IMU coupled with a magnetometer on the same integrated chip, and a Raspberry Pi (or similar processor) with an up-to-date version of the World Magnetic Model (WMM) [11]. The question we then address is "How can we create a derived attitude measurement from our limited resources?". While attitude is not directly measured from any of our available resources, we can use a combination of our resources to create a derived attitude measurement. Given a magnetometer, we can measure Earth's magnetic field in the satellite body reference frame. Then, using the GPS, we can provide a position estimate to the WMM to obtain an estimate of Earth's magnetic field in an inertial reference frame. This provides one vector pair described in two different reference frames. However, two vector pairs are required for attitude determination using the well-known TRIAD algorithm [12, 13]. Since we are in low earth orbit, we will take advantage of the fact that the smallsat will be traveling at an average speed of 7.8 km/s which leads to a measurable change in the magnetic field acting on the vehicle as time progresses. This rate of change can be computed through a finite difference approximation between two consecutive measurements in time. We now have two vector pairs (magnetic field and rate of change of magnetic field due to velocity) and can compute a derived attitude measurement using the TRIAD algorithm.

1.2.3 Dual-Purpose IMU

The IMU not only serves to support the TRIAD algorithm in producing a derived attitude measurement, but in propagating the position, velocity, and attitude between measurements. To propagate the position and velocity in a Kalman filter, the sum of all accelerations acting on the smallsat must be modeled or otherwise measured. These accelerations are gravity, drag, solar radiation pressure, etc. However, since we have an IMU capable of measuring non-gravitational acceleration, we do not need to model non-gravitational accelerations. We will still need a model in the Kalman filter for gravity, but a simple spherical planet model suffices for our application. To propagate the attitude in a Kalman filter, the angular rate of change must be modeled or otherwise measured. The IMU can also measure angular velocity. Therefore, we have a method of directly measuring the rate of change of our velocity and attitude without having to incorporate complex, mathematical models (other than gravity).

One issue with using the IMU to provide the Kalman filter the non-gravitational acceleration and angular velocity is that the IMU output is corrupted with systematic and random errors. In addition to random noise and bias, the IMU also possesses scaling, misalignment, and non-orthogonality errors. If these errors are not modeled and incorporated into our Kalman filter, our state estimation error covariance estimate may not be accurate. Using the IMU to propagate the position, velocity, and attitude will cause the estimates to drift over time, therefore, it is prudent to model the IMU systematic errors to provide the Kalman filter with information on how they affect our state dynamics and measurements.

1.3 Thesis Motivation

Guidance, navigation, and control (GN&C) is generally known as a challenging field. Our objective is not only to answer the question of feasibility of using a MEMS IMU for smallsat navigation, but to provide a systematic and concise derivation of this navigation system for future researchers to further advance state of the art navigation systems.

1.4 Other Applications

The navigation problem explored in this work is common not only among small satellites, but many types of aircraft and spacecraft. Similar solutions used for small satellites can be applied to larger vehicles as well. Once the solution to the navigation problem is understood, it can be used as a starting point on other vehicles.

1.5 Mathematical Notations

- Scalar values are denoted by lowercase italic letters: $t, q, etc.$
- Vectors are denoted by lowercase bold letters: $\mathbf{a}$, $\mathbf{v}$, etc.
- Unit vectors are denoted by lowercase bold letters with arrow overheads: $\vec{\mathbf{a}}$, $\vec{\mathbf{v}}$, etc.
- Quaternions are denoted by lowercase bold letters with overbars: $\bar{\mathbf{q}}$.
- Matrices are denoted by uppercase bold letters: $\mathbf{T}$, $\mathbf{M}$, etc.

1.5.1 Vector Descriptions

There are two vector descriptions used in this thesis that utilize a superscript and a subscript. The superscript will most commonly represent the reference frame in which the vector is being described. For example, the acceleration **a** measured in the body reference frame b is denoted by $\mathbf{a}^b$. The second piece of information will be used to describe specific information about the vector, such as the type of data it represents or the relative position with respect to a point. In this thesis, the subscripts will have two meanings:

1. A vector that represents the value of a quantity relative to another.
    - For example, if a vector **p** represents the position of a sensor s measured from or relative to an origin point r of a reference frame f, the vector description is $\mathbf{p}^f_{s/r}$ and is read as "The position of the sensor s with respect to the point r described in the reference frame f". The forward slash denotes "with respect to" and is the key piece of information for this specific superscript description.
2. A vector that represents a specific type of data
    - For example, if a vector **a** represents the non-gravitational acceleration of a sensor s in a reference frame f, the vector description is $\mathbf{a}^f_{s,ng}$ and is read as "The non-gravitational acceleration of the sensor s described in the reference frame f". The comma denotes a specific data type and is the key piece of information for this specific superscript description.

1.5.2 Transformation Descriptions

A transformation matrix transforms a vector from one reference frame to another through matrix-vector multiplication. This process is also referred to as "vector mapping". In the Chapter 2, transformation matrices will be discussed in detail. Transformation and rotation matrices will be denoted by **T** and **R**, respectively, with a subscript denoting the original reference frame and a super script denoting the transformed reference frame. For example, $\mathbf{T}_i^b$ denotes a transformation from an inertial reference frame i to a body reference frame b. Rotation matrices rotate vectors into different reference frames instead of transforming them, but are in the same form and have the same descriptions. For example, $\mathbf{R}_i^b$ rotates a vector about the axis and angle that describes the difference in orientation between the inertial reference frame i and the body reference frame b while still describing the vector in the original reference frame i. This is useful for visualizing reference frames with respect to another since they are made up of a triad of vectors. If the body reference frame is at an orientation that is rotated 30° from the original reference frame i about the x-axis, then the rotation matrix will rotate any vector 30° about that same axis.

1.6 Axis Representation

All figures utilize colors according to the axis. The x-axis is denoted by a red color, the y-axis is denoted by a green color, and the z-axis is denoted by a blue color, as illustrated in Figure 1.2.

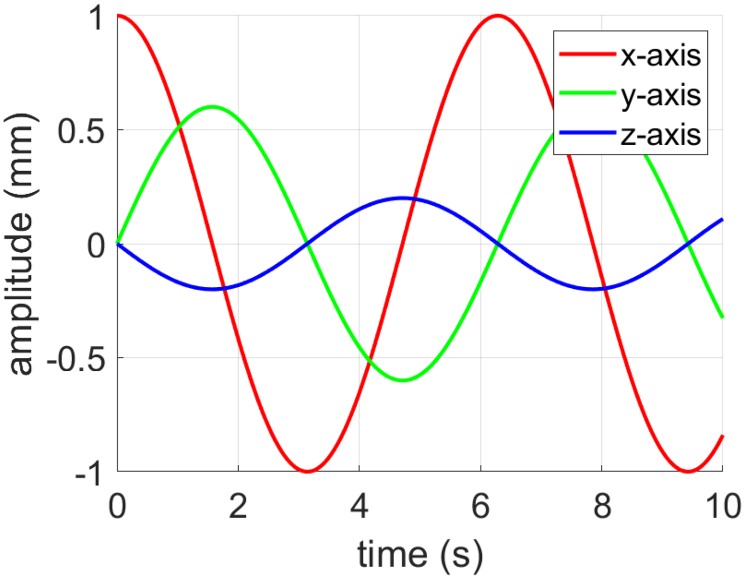


Figure 1.2: Axis Color Assignment

1.7 Reference Frames

Throughout this thesis many reference frames are used. These reference frames are all right-handed Cartesian coordinate reference frames and are depicted in Figures 1.3 - 1.5.

1.7.1 Earth-Centered, Earth-Fixed Frame (ECEF)

The ECEF reference frame (Figure 1.3) is fixed at the center of mass of the Earth and rotates. The ECEF reference frame coordinate axes are defined as

- The z^f-axis is coincident with true north.
- The x^f-axis is orthogonal to the z^f-axis and coincides with the prime meridian, 0° longitude and 0° latitude.
- The y^f-axis is orthogonal to the x^f and z^f axes and completes the right-handed Cartesian coordinate system.

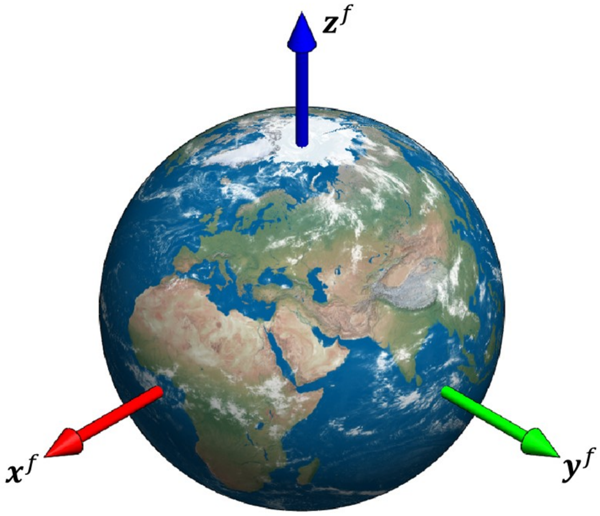


Figure 1.3: ECEF Reference Frame

Note that the ECEF reference frame does not rotate about z^f due to polar motion, also known as wobble. However, there is a known time-varying transformation for ECEF to ECI.

1.7.2 Earth-Centered Inertial Frame (ECI) - J2000

The J2000 ECI reference frame (Figure 1.4) is an an inertial reference frame centered at the Earth's center of mass. The J2000 ECI reference frame is aligned as

- The z^i-axis is initially coincident with Earth's spin axis oriented upwards with the northern hemisphere at the epoch January 1st, 2000 12:00 terrestrial time.
- The x^i-axis is orthogonal to the z^i-axis and lies along the line of intersection between the ecliptic plane and the equatorial plane pointing towards the sun at the J2000 epoch.
- The y^i-axis is orthogonal to the x^i and z^i axes and completes the standard right handed Cartesian coordinate system.

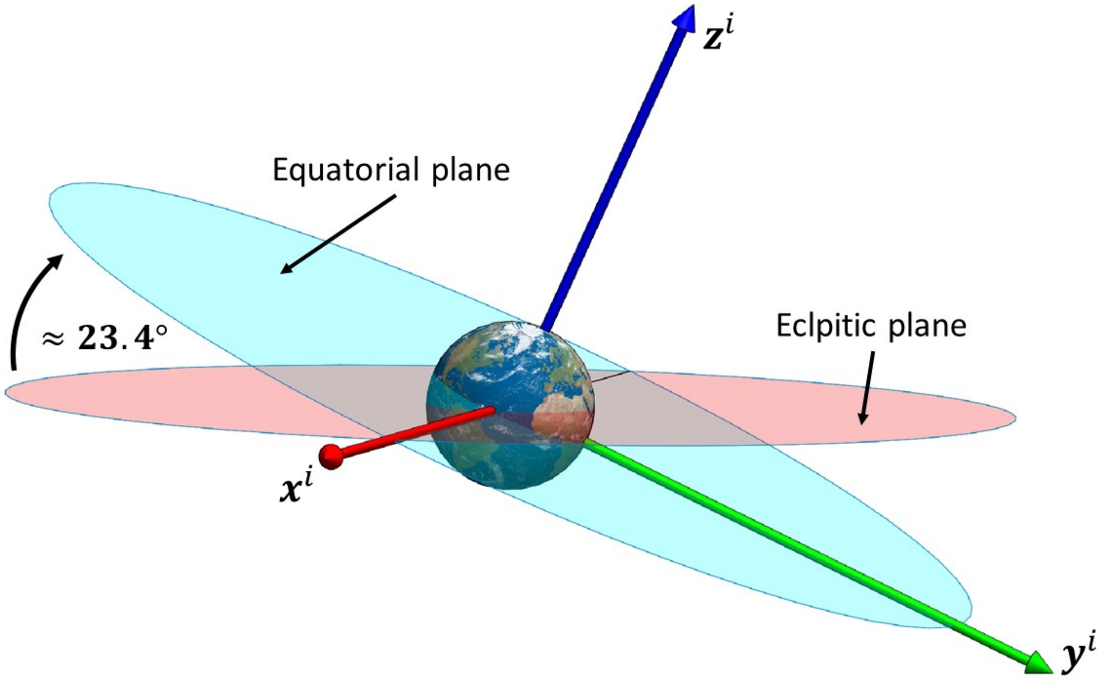


Figure 1.4: ECI Reference Frame

The ecliptic plane is the plane containing the motion describing Earth's orbit around the sun. The equatorial plane is the plane defined by Earth's equator. These planes are illustrated in Figure 1.4.

1.7.3 North-East-Down Frame (NED)

The NED reference frame is a tangential stationary reference frame that varies with latitude and longitude. It is local and fixed with respect to the coordinates. The NED reference frame is aligned as

- The z^{ned}-axis is normal to the Earth's surface at the given coordinates and points downwards.
- The x^{ned}-axis is orthogonal to the z^{ned}-axis and points north from the given local coordinates.
- The y^{ned}-axis is orthogonal to the x^{ned} and z^{ned} axes, points east from the given local coordinates, and completes the right handed Cartesian coordinate system

1.7.4 Spacecraft Body Frame

The spacecraft body reference frame is fixed with the spacecraft with its origin coincident with the center of mass. The spacecraft body reference frame is defined as

- The z^{b}-axis is normal to the top face of the satellite and points up
- The x^{b}-axis is orthogonal to the z^{b}-axis and is normal to one of the side faces.
- The y^{b}-axis is orthogonal to the x^{b} and z^{b} axes and completes the right handed Cartesian coordinate system

Note that this coordinate system is chosen by the design engineer and is somewhat arbitrary in terms of how the axes are defined.

1.7.5 IMU Case Frame

The IMU case reference frame is defined by the positioning of the IMU with respect to the spacecraft center of mass. This reference frame is designated by the spacecraft

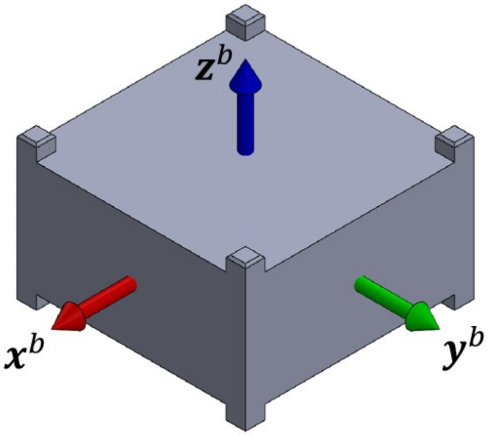


Figure 1.5: Body Reference Frame

design engineer and for the purposes of this thesis will be assumed to be aligned with the spacecraft body reference frame. However, the derivations in the following chapters will still include the case reference frame when transforming between reference frames.

1.7.6 TRIAD Frame

The TRIAD reference frame is a frame in which attitude measurements are represented. While the magnetometer used for the attitude computations is built into the IMU, it is a separate device and has its own reference frame. This reference frame is defined in the IMU data sheet.

1.7.7 Reference Frame Symbol Designations

All of the reference frames will be used to describe vectors and transformations using subscripts and superscripts. Their designations are listed in Table 1.1.

Table 1.1: Reference Frame Symbol Designations

Symbol	Reference Frame
i	Earth-Centered-Inertial
f	Earth-Centered, Earth-Fixed
ned	North-East-Down
b	Spacecraft Body
c	IMU Case
tr	TRIAD

1.8 Thesis Organization

Chapter 2 presents the mathematical fundamentals. Chapter 3 presents the Kalman filter fundamentals, the backbone of this navigation system. Chapter 4 derives the governing equations of motion of the satellite. These mathematical models are important and will be used later in the extended Kalman filter design. Chapter 5 derives the IMU models for state propagation and sensor models for state measurements. Chapter 6 employs the specific equations derived in previous chapters into the general format used in a Kalman filter for our specific application. Chapter 7 presents the results of the filter simulations and proposes a sub-optimal Kalman filter to improve computational efficiency when computational power is limited.

Chapter 2: Mathematical Fundamentals

The mathematical principles required in the subsequent derivations are presented.

2.1 Vectors

Vectors can be both 3-dimensional and 4-dimensional (in the case of a quaternion). Given a right-handed Cartesian coordinate system, each element of the vector describes the magnitude of that element in its respective axis. Vectors are denoted by

$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \in \mathbb{R}^3 ,$$

where v_x represents the magnitude in the x-direction, v_x represents the magnitude in the y-direction, and v_x represents the magnitude in the z-direction, in a given reference frame. The vector Euclidean norm is defined as

$$||\mathbf{v}|| = \sqrt{v_x^2 + v_y^2 + v_z^2} .$$

This returns a scalar value that represents the overall magnitude (or length) of a vector. Dividing a vector by its norm yields a unit vector. This process is called normalizing (or unitizing). A unit vector is given by

$$\vec{\mathbf{v}} = \frac{\mathbf{v}}{||\mathbf{v}||} .$$

For a satellite orbiting Earth, the state vector is the aggregate of multiple vectors describing position, velocity, attitude, and sensor errors. In the case of velocity, the velocity vector represents the direction of speed that an object is traveling with respect to a given reference frame and the norm represents the magnitude of that speed (directional velocity). For acceleration, it represents the direction that an object is accelerating with respect to a given reference frame and the norm represents the magnitude of that acceleration (directional acceleration). Vectors can also be used to represent orientation.

2.1.1 Vector Transpose

Given the column vector $\mathbf{v} \in \mathbb{R}^{nx1}$, the transpose is the row vector $\mathbf{v}^T \in \mathbb{R}^{1xn}$. For example, if

$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \in \mathbb{R}^{3x1} \; ,$$

then, it follows that

$$\mathbf{v}^T = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix} \in \mathbb{R}^{1x3} \; .$$

Note that we typically denote $\mathbf{v} \in \mathbb{R}^{3x1}$ as $\mathbf{v} \in \mathbb{R}^3$.

2.1.2 Vector Dot/Inner Product

Consider $\mathbf{v_1} \in \mathbb{R}^3$ and $\mathbf{v_2} \in \mathbb{R}^3$ given by

$$\mathbf{v}_1 = \begin{bmatrix} v_{1,x} \\ v_{1,y} \\ v_{1,z} \end{bmatrix} \quad \text{and} \quad \mathbf{v}_2 = \begin{bmatrix} v_{2,x} \\ v_{2,y} \\ v_{2,z} \end{bmatrix} \; . \tag{2.1}$$

The dot (or inner product) of these two vectors yields a scalar. The dot product is defined as

$$\mathbf{v}_1 \cdot \mathbf{v}_2 = \mathbf{v}_1^T \mathbf{v}_2 = v_{1,x}v_{2,x} + v_{1,y}v_{2,y} + v_{1,z}v_{2,z} \; .$$

2.1.3 Vector Cross Product

Consider the two vectors in Eq. 2.1. The cross product yields another vector that is orthogonal to both vectors. The cross product is defined as

$$\mathbf{v}_1 \times \mathbf{v}_2 = \begin{bmatrix} v_{1,y}v_{2,z} - v_{1,z}v_{2,y} \\ v_{1,z}v_{2,x} - v_{1,x}v_{2,z} \\ v_{1,x}v_{,2y} - v_{1,y}v_{2,x} \end{bmatrix} \; . \tag{2.2}$$

2.2 Matrices

A matrix is an arrangement of numbers in rows and columns. The size of a matrix is defined as $\mathbf{M}_{nxm}$ where n represents the number of rows and m represents the number of columns. The element of a matrix $\mathbf{M}$ is represented as $\mathbf{M}_{i,j}$ where i indicates the row index and j indicates the column index. For example,

$$\mathbf{M}_{5x3} = \begin{bmatrix} \mathbf{M}_{1,1} & \mathbf{M}_{1,2} & \mathbf{M}_{1,3} \\ \mathbf{M}_{2,1} & \mathbf{M}_{2,2} & \mathbf{M}_{2,3} \\ \mathbf{M}_{3,1} & \mathbf{M}_{3,2} & \mathbf{M}_{3,3} \\ \mathbf{M}_{4,1} & \mathbf{M}_{4,2} & \mathbf{M}_{4,3} \\ \mathbf{M}_{5,1} & \mathbf{M}_{5,2} & \mathbf{M}_{5,3} \end{bmatrix} \; , \quad \text{and} \quad \mathbf{M}_{3x5} = \begin{bmatrix} \mathbf{M}_{1,1} & \mathbf{M}_{1,2} & \mathbf{M}_{1,3} & \mathbf{M}_{1,4} & \mathbf{M}_{1,5} \\ \mathbf{M}_{2,1} & \mathbf{M}_{2,2} & \mathbf{M}_{2,3} & \mathbf{M}_{2,4} & \mathbf{M}_{2,5} \\ \mathbf{M}_{3,1} & \mathbf{M}_{3,2} & \mathbf{M}_{3,3} & \mathbf{M}_{3,4} & \mathbf{M}_{3,5} \end{bmatrix} \; .$$

A matrix $\mathbf{M} \in \mathbb{R}^{nxm}$ is called a square matrix if $n = m$. Matrices can be multiplied only if their inner dimensions match. For example, if we have $\mathbf{M}_1 \in \mathbb{R}^{mxn}$ and $\mathbf{M}_2 \in \mathbb{R}^{nxm}$, then

$$\mathbf{M} = \mathbf{M}_1\mathbf{M}_2 \in \mathbb{R}^{mxm} \; .$$

Also, matrix multiplication is not commutative. In general,

$$\mathbf{M}_1\mathbf{M}_2 \neq \mathbf{M}_2\mathbf{M}_1 \; .$$

2.2.1 Matrix Inverse and Transpose

For a square matrix $\mathbf{M}$, the inverse of $\mathbf{M}$ is denoted by $\mathbf{M}^{-1}$ and is a matrix such that

$$\mathbf{M}^{-1}\mathbf{M} = \mathbf{M}\mathbf{M}^{-1} = \mathbf{I} \; ,$$

where $\mathbf{I}$ is the identity matrix. However, not all matrices have an inverse. A matrix with an inverse is said to be nonsingular, which implies that its determinant is not equal to zero. If the determinant is equal to zero then the inverse does not exist and the matrix is said to be singular. The transpose of a matrix $\mathbf{M}$ is a matrix resulting from switching the row and column indices. For example, if we have

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{1,1} & \mathbf{M}_{1,2} & \mathbf{M}_{1,3} \\ \mathbf{M}_{2,1} & \mathbf{M}_{2,2} & \mathbf{M}_{2,3} \\ \mathbf{M}_{3,1} & \mathbf{M}_{3,2} & \mathbf{M}_{3,3} \\ \mathbf{M}_{4,1} & \mathbf{M}_{4,2} & \mathbf{M}_{4,3} \\ \mathbf{M}_{5,1} & \mathbf{M}_{5,2} & \mathbf{M}_{5,3} \end{bmatrix} \in \mathbb{R}^{5x3} \; ,$$

then, it follows that

$$\mathbf{M}^T = \begin{bmatrix} \mathbf{M}_{1,1} & \mathbf{M}_{2,1} & \mathbf{M}_{3,1} & \mathbf{M}_{4,1} & \mathbf{M}_{5,1} \\ \mathbf{M}_{1,2} & \mathbf{M}_{2,2} & \mathbf{M}_{3,2} & \mathbf{M}_{4,2} & \mathbf{M}_{5,2} \\ \mathbf{M}_{1,3} & \mathbf{M}_{2,3} & \mathbf{M}_{3,3} & \mathbf{M}_{4,3} & \mathbf{M}_{5,3} \end{bmatrix} \in \mathbb{R}^{3x5} \; .$$

2.3 Matrix Representation of Attitude/Orientation

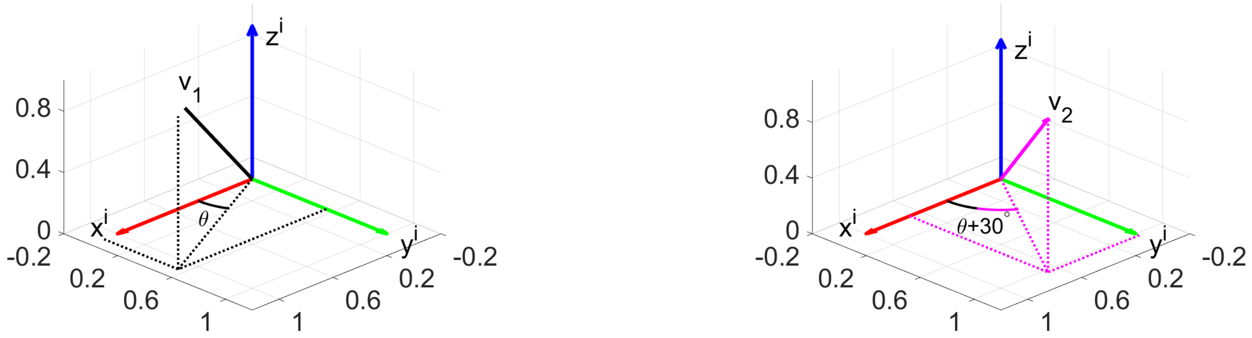
One of the primary uses of matrices for navigation systems is to represent different frames of reference relative to each other and to rotate vectors within a given reference frame. Since navigation information is provided in a variety of reference frames, we must define all reference frames and the relationship between them, and this is done with transformation matrices. We can describe the orientation of a vector using either the rotation matrix or an attitude quaternion (a 4-dimensional unit vector). Rotation and transformation matrices are orthonormal matrices with $\mathbf{R}^{-1} = \mathbf{R}^T$. Rotation and transformation matrices are similar in many ways, but are, in fact, different. The definitions for rotation and transformation matrices are now described, as well as their relationship with the attitude quaternion and applications to navigation.

2.3.1 Rotation Matrix

Consider a vector $\mathbf{v}_1 \in \mathbb{R}^3$ in a given reference frame and the rotation matrix $\mathbf{R} \in \mathbb{R}^{3x3}$. Then the product $\mathbf{v}_2 = \mathbf{R}\mathbf{v}_1 \in \mathbb{R}^3$ is a vector rotated from its original position $\mathbf{v}_1$ described in the same reference frame. This is referred to as vector rotation. During this process, the matrix $\mathbf{R}$ is rotating the vector $\mathbf{v}_1$. The matrix $\mathbf{R}$ is called a rotation matrix. Figure 2.1 depicts a vector $\mathbf{v}_1$ rotated by a rotation matrix that rotates the vector 30° about the z-axis of the reference frame.

2.3.2 Transformation Matrix

Vector rotation was described in Section 2.3.1 with an origin frame and a vector being rotated from one direction to another, giving it new coordinates in the same origin frame. A transformation matrix, denoted by $\mathbf{T}_a^b$, transforms a vector from frame a to frame b. This is essential in navigation system design because many reference frames are used, such as the inertial reference frame, the Earth-centered, Earth-fixed reference



(a) Vector Before Rotation

(b) Vector After Rotation

Figure 2.1: Rotating a Vector with a Rotation Matrix

frame, the body reference frame, and various sensor reference frames. For example, non-gravitational acceleration is measured by the IMU in the body reference frame, however, gravitational acceleration is naturally represented in the Earth-centered, Earth-fixed reference frame. A transformation matrix that maps from the body reference frame to the ECEF reference frame is required in many instances. Figure 2.2 depicts this with a simple z-axis transformation matrix. The vector in Figure 2.2 has two possible descriptions: $\mathbf{v}^f$ and $\mathbf{v}^b$. Suppose the vector was described in the body reference frame b and it is desired to know the vector coordinates in the ECEF reference frame f. To represent it in ECEF

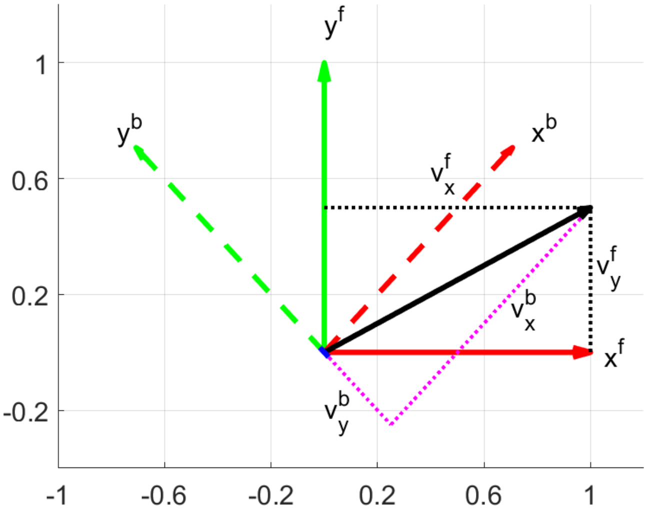


Figure 2.2: Frame Description Top View

reference frame coordinates, a body to fixed transformation $\mathbf{T}_b^f$ is employed with

$$\mathbf{v}^f = \mathbf{T}_b^f \mathbf{v}^b \,.$$

2.3.3 Rotation Matrix Construction

Each axis has its own rotation matrix definition given by

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}, \; \mathbf{R}_y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}, \; \mathbf{R}_z = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.3}$$

The angle γ is the angle of rotation about the x-axis, β is the angle of rotation about the y-axis and α is the angle of rotation about the z-axis. These matrices not only signify vector rotation, but also represent the orientation of an object with respect to a fixed frame of reference. Observing each column of the rotation matrix from left to right, it can be seen that all columns are orthogonal unit vectors, more specifically, they are the x, y, and z axes, respectively, of the body reference frame or orientation triad. For example, a rotation matrix that describes a 45°rotation about the z-axis is

$$\mathbf{R}_z(45) = \begin{bmatrix} 0.707 & -0.707 & 0 \\ 0.707 & 0.707 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The identity matrix defines a zero-rotation orientation,

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In Figure 2.3, each column of both the identity matrix and $\mathbf{R}_z(45)$ are plotted, where the first column is red, the second is green and the third is blue. This depicts a rotation between two reference frames. The inertial reference frame is plotted with solid lines using the identity matrix and the body reference frame is plotted with dashed lines using the rotation matrix, signifying an objects rotation or orientation with respect to the inertial reference frame.

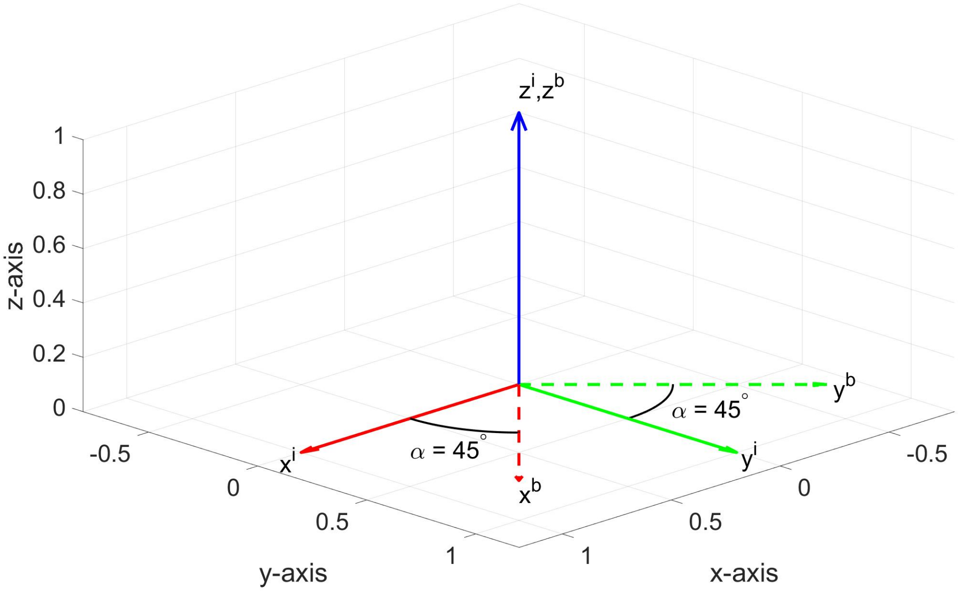


Figure 2.3: Plot of Fixed and Body Reference Frames

The individual rotation matrices in Eq. 2.3 are constrained to rotate about only each of their axes. For representing the orientation of an object in 3-dimensional space without single axis constraints, the Euler rotation sequence can be used. The Euler sequence illustrates that consecutive rotations of different angles about different axes can be represented as one equivalent rotation matrix. This is represented by the product of the rotation matrices. For example, if an object with its body reference frame initially aligned with a fixed reference frame is rotated by α about its z-axis, then rotated by β about its y-axis, and then finally rotated by γ about its x-axis, then the rotation matrix that represents the equivalent rotation of those three consecutive rotations is

$$\mathbf{R} = \mathbf{R}_x(\gamma)\mathbf{R}_y(\beta)\mathbf{R}_z(\alpha)\ . \tag{2.4}$$

This is known as the 3-2-1 or z-y-x Euler sequence, which is usually used to describe the attitude of airplanes since they follow that sequence of rotations during take-off. Note that the key assumption behind the Euler sequence is that the consecutive rotations are done about the body reference frame, not the fixed reference frame.

Another way to represent the orientation of an object in 3-dimensional space is by a single axis and angle of rotation. Euler's theorem states that any sequence of rotations can be described by one rotation sequence of θ about a single axis defined in the reference frame coordinates. Given the equivalent axis and angle of rotation, the rotation matrix is [14]

$$\mathbf{R} = [\mathbf{I} - \sin\theta[\mathbf{e}\times] + (1 - \cos\theta)[\mathbf{e}\times]^2]^T\ , \tag{2.5}$$

where $\mathbf{e}$ is the equivalent axis of rotation and θ is the equivalent angle.

The inverse of a rotation matrix always represents the same rotation in the opposite direction. For example, a rotation from an inertial reference frame i to a body reference frame b is given by $\mathbf{R}_i^b$. The inverse of that would be a rotation from a body reference frame b to a inertial reference frame i given by $\mathbf{R}_b^i$.

2.3.4 Transformation Matrix Construction

Like rotation matrices, transformation matrices have similar definitions. Their relationship is given by

$$\mathbf{T} = \mathbf{R}^T\ .$$

For single axes, the reference frame transformations are given by

$$\mathbf{T}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & \sin\gamma \\ 0 & -\sin\gamma & \cos\gamma \end{bmatrix} , \; \mathbf{T}_y = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} , \; \mathbf{T}_z = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} , \tag{2.6}$$

where

$$\mathbf{T}_x(\gamma) = \mathbf{R}_x^T(\gamma) \, , \; \mathbf{T}_y(\gamma) = \mathbf{R}_y^T(\gamma) \, , \quad \text{and} \quad \mathbf{T}_z^T(\gamma) = \mathbf{R}_z(\gamma) \, .$$

Eq. 2.4 then becomes

$$\mathbf{T} = \mathbf{T}_z(\alpha)\mathbf{T}_y(\beta)\mathbf{T}_x(\gamma) \, .$$

The single axis/angle transformation is

$$\mathbf{T} = \mathbf{I} - \sin\theta[\mathbf{e}\times] + (1 - \cos\theta)[\mathbf{e}\times]^2 \, . \tag{2.7}$$

which is the transpose of Eq. 2.5.

The inverse or transpose of a transformation matrix always represents the same transformation in the opposite direction. For example, a transformation from an inertial reference frame i to a body reference frame b is given by $\mathbf{T}_i^b$. The inverse of that would be a transformation from a body reference frame b to an inertial reference frame i given by $\mathbf{T}_b^i$.

$$\mathbf{T}_i^b\mathbf{T}_b^i = (\mathbf{T}_b^i)^{-1}\mathbf{T}_b^i = (\mathbf{T}_b^i)^T\mathbf{T}_b^i = \mathbf{I} \, .$$

2.3.5 Constraints for Rotation and Transformation Matrices

The orthonormal constraints for rotation and transformation matrices are given in terms of individual columns. Suppose $\mathbf{T} = \begin{bmatrix} \mathbf{x} \mid \mathbf{y} \mid \mathbf{z} \end{bmatrix}$ where $\mathbf{x} \in \mathbb{R}^3, \mathbf{y} \in \mathbb{R}^3$, and $\mathbf{z} \in \mathbb{R}^3$. Then,

$$||\mathbf{x}|| = 1 \, , \; ||\mathbf{y}|| = 1 \, , \; ||\mathbf{z}|| = 1 \, ,$$

and

$$\mathbf{x} \cdot \mathbf{y} = 0\,,\ \mathbf{x} \cdot \mathbf{z} = 0\,,\ \mathbf{y} \cdot \mathbf{z} = 0\,.$$

2.3.6 Transformation and Rotation Matrix Sequence Inverse

Given a sequence of rotation or transformation matrices, the inverse of this sequence is given by

$$[\mathbf{R}_x(\gamma)\mathbf{R}_y(\beta)\mathbf{R}_z(\alpha)]^{-1} = \mathbf{R}_z^{-1}(\alpha)\mathbf{R}_y^{-1}(\beta)\mathbf{R}_x^{-1}(\gamma) = \mathbf{T}_z(\alpha)\mathbf{T}_y(\beta)\mathbf{T}_x(\gamma)\,.$$

2.3.7 Vectors in Matrix Form

For specific applications, vector operations can be represented by matrices. The three matrices that will be utilized are the vector cross product matrix, the vector purely off-diagonal matrix, and the vector diagonal matrix. Each matrix has a unique property that is utilized in the IMU accelerometer and gyroscope model, as well as relating the small angle quaternion to the transformation matrix.

2.3.7.1 Vector Cross Product Matrix

The vector cross product matrix represents the cross product of two vectors, rather than using Eq. 2.2. The cross product and vector cross product matrix are related through

$$\mathbf{v}_1 \times \mathbf{v}_2 = [\mathbf{v}_1 \times]\mathbf{v}_2\,, \tag{2.8}$$

where

$$[\mathbf{v}_1 \times] = \begin{bmatrix} 0 & -v_{1,z} & v_{1,y} \\ v_{1,z} & 0 & -v_{1,x} \\ -v_{1,y} & v_{1,x} & 0 \end{bmatrix}\,,$$

which achieves the same result as taking the cross product of two vectors. The cross product matrix is a skew-symmetric matrix, which satisfies

$$\mathbf{A}^T = -\mathbf{A}\,,$$

and

$$\mathbf{v}_1 \times \mathbf{v}_2 = [\mathbf{v}_1 \times]\mathbf{v}_2 = -[\mathbf{v}_2 \times]\mathbf{v}_1\,. \tag{2.9}$$

2.3.7.2 *Vector Purely Off-Diagonal Matrix*

The vector purely off-diagonal matrix is the absolute value of the vector cross product matrix where

$$[\mathbf{v}_1|\times|] = \begin{bmatrix} 0 & v_{1,z} & v_{1,y} \\ v_{1,z} & 0 & v_{1,x} \\ v_{1,y} & v_{1,x} & 0 \end{bmatrix}.$$

The purely off-diagonal matrix is a symmetric matrix that satisfies

$$\mathbf{A}^T = \mathbf{A}\,,$$

and

$$[\mathbf{v}_1|\times|]\mathbf{v}_2 = [\mathbf{v}_2|\times|]\mathbf{v}_1\,. \tag{2.10}$$

2.3.7.3 *Vector Diagonal Matrix*

The diagonal matrix is a matrix in which the elements of a vector are placed along the diagonal of a matrix,

$$[\mathbf{v}_1 \diagdown] = \begin{bmatrix} v_{1,x} & 0 & 0 \\ 0 & v_{1,y} & 0 \\ 0 & 0 & v_{1,z} \end{bmatrix}.$$

The diagonal matrix is a symmetric matrix that satisfies

$$\mathbf{A}^T = \mathbf{A} \, ,$$

and

$$[\mathbf{v}_1\diagdown]\mathbf{v}_2 = [\mathbf{v}_2\diagdown]\mathbf{v}_1 \, . \tag{2.11}$$

2.3.8 Unknown Rotation Matrix Calculation

Given $\mathbf{v}_1 \in \mathbb{R}^3$ and $\mathbf{v}_2 \in \mathbb{R}^3$, there exists an infinite amount of solutions for $\mathbf{M} \in \mathbb{R}^{3x3}$ given

$$\mathbf{M}\mathbf{v}_1 = \mathbf{v}_2 \, . \tag{2.12}$$

Eq. 2.12 is under-determined. We are only given a single pair of vectors to describe the 3-dimensional orientation of one with respect to another. One solution can be found by computing the angle θ between the vectors and the axis **e** orthogonal to both vectors.

$$\theta = cos^{-1}\frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{||\mathbf{v}_1||||\mathbf{v}_2||} \, , \tag{2.13}$$

and

$$\vec{\mathbf{e}} = \frac{\mathbf{v}_1 \times \mathbf{v}_2}{||\mathbf{v}_1||||\mathbf{v}_2||} \, . \tag{2.14}$$

If we assume that this axis is truly representative of the vector transformation, then Eqs. 2.13 and 2.14 can be substituted into Eqs. 2.5 and 2.7, yielding the rotation and transformation matrices, respectively. However, this is not always valid. An example that invalidates this assumption is when the two vectors are co-linear, but rotate about their own axes. It would be impossible to know this with only a single pair of vectors. Therefore, another pair is required, and for attitude determination two vector pairs are always required. The TRIAD algorithm is one method of attitude determination that utilizes two vector pairs and will be employed in Chapter 5 to compute quaternion measurements.

2.3.9 Derivative of a Rotation and Transformation Matrix

Given a rotation matrix representing the rotation from an inertial reference frame to a body reference frame $\mathbf{R}_i^b$, the derivative of the rotation matrix with respect to time is given by

$$\dot{\mathbf{R}}_i^b = [\boldsymbol{\omega}_{b/i}^b \times]\mathbf{R}_i^b \,, \tag{2.15}$$

where $\boldsymbol{\omega}_{b/i}^b$ is the angular velocity of the body reference frame with respect to the inertial reference frame, given in the body reference frame.

Given a transformation matrix representing the transformation from an inertial reference frame to a body reference frame $\mathbf{T}_i^b$, the derivative of the transformation matrix with respect to time is given by

$$\dot{\mathbf{T}}_i^b = -[\boldsymbol{\omega}_{b/i}^b \times]\mathbf{T}_i^b \,, \tag{2.16}$$

where $\boldsymbol{\omega}_{b/i}^b$ is the angular velocity of the body reference frame with respect to the inertial reference frame, given in the body frame.

2.4 Attitude Quaternions

We utilize attitude quaternions to represent the orientation of the spacecraft body reference frame with respect to the inertial reference frame. The attitude quaternion uses an equivalent angle and Euler axis to describe the orientation of an object , given by

$$\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{q} \\ q_0 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_0 \end{bmatrix} \in \mathbb{R}^4 \quad \text{and} \quad \mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \in \mathbb{R}^3 \,,$$

which is defined in terms of the equivalent angle and the Euler axis as

$$\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{e}\sin(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) \end{bmatrix} = \begin{bmatrix} e_x \sin(\frac{\theta}{2}) \\ e_y \sin(\frac{\theta}{2}) \\ e_z \sin(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) \end{bmatrix} , \tag{2.17}$$

where θ is the angular displacement between the body reference frame and the inertial reference frame and $\mathbf{e}$ is the unit vector about which the body rotates. The attitude quaternion is also constrained by a unit norm

$$||\bar{\mathbf{q}}|| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_0^2} = 1 \, . \tag{2.18}$$

Like rotation matrices, a sequence of consecutive quaternion rotations can be represented by one quaternion rotation, which is the product of all consecutive rotations. However, the attitude quaternion product has a special definition for two cases of rotation. For a rotation (described by the quaternion $\bar{\mathbf{q}}$) given in the body reference frame, the product of two attitude quaternions $\bar{\mathbf{q}}$ and $\bar{\mathbf{d}}$ is given by

$$\bar{\mathbf{q}} \otimes \bar{\mathbf{d}} = \begin{bmatrix} \mathbf{q} \\ q_0 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{d} \\ d_0 \end{bmatrix} = \begin{bmatrix} \mathbf{q}d_0 + q_0\mathbf{d} - \mathbf{q} \times \mathbf{d} \\ q_0 d_0 - \mathbf{q} \cdot \mathbf{d} \end{bmatrix} . \tag{2.19}$$

For a rotation (described by the quaternion $\bar{\mathbf{q}}$) given in the inertial reference frame, the product of two attitude quaternions $\bar{\mathbf{q}}$ and $\bar{\mathbf{d}}$ is given by

$$\bar{\mathbf{q}} \otimes \bar{\mathbf{d}} = \begin{bmatrix} \mathbf{q} \\ q_0 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{d} \\ d_0 \end{bmatrix} = \begin{bmatrix} \mathbf{q}d_0 + q_0\mathbf{d} + \mathbf{q} \times \mathbf{d} \\ q_0 d_0 - \mathbf{q} \cdot \mathbf{d} \end{bmatrix} . \tag{2.20}$$

We use Eq. 2.19 to compute the product of two quaternions in our navigation system since the angular rate of change and displacement due to the angular rate of change (over a small time Δt) of the spacecraft is more naturally represented in the body reference frame, given that we utilize a gyroscope to measure the body angular rate. The inverse of the attitude quaternion is defined as

$$\bar{\mathbf{q}}^{-1} = \begin{bmatrix} -\mathbf{q} \\ q_0 \end{bmatrix} , \tag{2.21}$$

which (like the inverse of a rotation matrix) represents the same rotation in the opposite direction. One very important rule when working with attitude quaternions (and rotation matrices) is that they cannot be added (or subtracted) to create another attitude quaternion. Regular quaternions can be added (or subtracted), but all quaternions in this thesis are attitude quaternions. The order of multiplication for quaternions is similar to that of rotation and transformation matrices. Given a *z*-*y*-*x* rotation sequence, we may represent the sequence in matrix format by

$$\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z \ .$$

The same *z*-*y*-*x* rotation sequence using quaternions is represented by

$$\bar{\mathbf{q}} = \bar{\mathbf{q}}_x \otimes \bar{\mathbf{q}}_y \otimes \bar{\mathbf{q}}_z \ .$$

The inverse of a quaternion sequence is given as

$$[\bar{\mathbf{q}}_x \otimes \bar{\mathbf{q}}_y \otimes \bar{\mathbf{q}}_z]^{-1} = [\bar{\mathbf{q}}_z]^{-1} \otimes [\bar{\mathbf{q}}_y]^{-1} \otimes [\bar{\mathbf{q}}_x]^{-1} \ .$$

2.4.1 Small Angle Quaternion

For a quaternion that represents a small angle rotation, the small angle quaternion can be approximated as

$$\delta\bar{\mathbf{q}} \approx \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\alpha} \\ 1 \end{bmatrix} , \tag{2.22}$$

where $\delta\boldsymbol{\alpha}$ is the vector of small Euler angles given by

$$\delta\boldsymbol{\alpha} = \begin{bmatrix} \gamma \\ \beta \\ \alpha \end{bmatrix} .$$

The small angle quaternion can also be converted into an equivalent angle and axis form of rotation rather than an Euler sequence. The equivalent angle is given by

$$\theta = ||\delta\boldsymbol{\alpha}|| \ , \tag{2.23}$$

and the equivalent axis is given by

$$\mathbf{e} = \frac{\delta\boldsymbol{\alpha}}{||\delta\boldsymbol{\alpha}||} . \tag{2.24}$$

2.4.2 Quaternion to Transformation Matrix

A quaternion describing a rotation from an inertial reference frame i to a body reference frame b in the form of a transformation matrix is given by [14]

$$\mathbf{T}_i^b(\bar{\mathbf{q}}) = (q_0^2 + ||\mathbf{q}||^2)\mathbf{I} - 2q_0[\mathbf{q}\times] + 2[\mathbf{q}\times]^2 . \tag{2.25}$$

The rotation matrix of a quaternion is just the transpose of the transformation matrix above,

$$\mathbf{R}_i^b(\bar{\mathbf{q}}) = \left[(q_0^2 + ||\mathbf{q}||^2)\mathbf{I} - 2q_0[\mathbf{q}\times] + 2[\mathbf{q}\times]^2\right]^T . \tag{2.26}$$

If Eq. 2.22 is substituted into Eq. 2.25 and approximated to first order, then the transformation matrix that represents the same small angle transformation from the inertial reference frame *i* to the body reference frame *b* is the small angle transformation matrix

$$\delta\mathbf{T}_i^b = \mathbf{I} - [\delta\boldsymbol{\alpha}\times] . \tag{2.27}$$

The inverse of $\delta\mathbf{T}_i^b = \mathbf{I} - [\delta\boldsymbol{\alpha}\times]$ can be approximated through the matrix inversion lemma [15] to first order as

$$(\mathbf{I} - [\delta\boldsymbol{\alpha}\times])^{-1} = \mathbf{I} + [\delta\boldsymbol{\alpha}\times] . \tag{2.28}$$

2.4.3 Rotation Matrix to Quaternion

We can convert from a rotation matrix, denoted by $\mathbf{R}$, to a quaternion by using [14]

$$\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{q} \\ q_0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \frac{\mathbf{R}_{3,2} - \mathbf{R}_{2,3}}{\sqrt{1 + \mathbf{R}_{1,1} + \mathbf{R}_{2,2} + \mathbf{R}_{3,3}}} \\ \frac{\mathbf{R}_{1,3} - \mathbf{R}_{3,1}}{\sqrt{1 + \mathbf{R}_{1,1} + \mathbf{R}_{2,2} + \mathbf{R}_{3,3}}} \\ \frac{\mathbf{R}_{2,1} - \mathbf{R}_{1,2}}{\sqrt{1 + \mathbf{R}_{1,1} + \mathbf{R}_{2,2} + \mathbf{R}_{3,3}}} \\ \sqrt{1 + \mathbf{R}_{1,1} + \mathbf{R}_{2,2} + \mathbf{R}_{3,3}} \end{bmatrix} . \tag{2.29}$$

2.4.4 Derivative of a Quaternion

Similar to the derivative of a rotation matrix with respect to time, the derivative of a quaternion with respect to time is given by

$$\dot{\bar{\mathbf{q}}}_i^b = \frac{1}{2}\bar{\boldsymbol{\omega}}_{b/i}^b \otimes \bar{\mathbf{q}}_i^b \tag{2.30}$$

where $\bar{\omega}^b_{b/i}$ is the angular rate quaternion given by

$$\bar{\omega}^b_{b/i} = \begin{bmatrix} \omega^b_{b/i} \\ 0 \end{bmatrix} . \tag{2.31}$$

Note that unlike the attitude quaternion, the angular rate quaternion does not have a magnitude of 1 since it is used to describe the angular rate of a rotating body. This is also known as a pure quaternion, a representation of a 3 dimensional vector as a quaternion.

2.4.5 Quaternion Vector Transformation

Similar to how frame transformation matrices can transform a vector from one reference frame to another, the quaternion can be utilized to do the same. Given a vector in the inertial reference frame $\mathbf{v}^i$ and the quaternion that represents the orientation of the body reference frame with respect to the inertial reference frame $\bar{\mathbf{q}}^b_i$, the vector can be described in the body reference frame by

$$\bar{\mathbf{v}}^b = \bar{\mathbf{q}}^b_i \otimes \bar{\mathbf{v}}^i \otimes [\bar{\mathbf{q}}^b_i]^{-1} , \tag{2.32}$$

where $\bar{\mathbf{v}}^b$ and $\bar{\mathbf{v}}^i$ are the pure vector quaternions defined as

$$\begin{bmatrix} \mathbf{v}^b \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{v}^i \\ 0 \end{bmatrix} .$$

A similar equation allows the transformation matrix $\mathbf{T}^b_i(\bar{\mathbf{q}})$ to be used to achieve the same result while still using quaternion multiplication,

$$\bar{\mathbf{v}}^b = \begin{bmatrix} \mathbf{T}^b_i(\bar{\mathbf{q}})\mathbf{v}^i \\ 0 \end{bmatrix} , \tag{2.33}$$

which implies that

$$\bar{\mathbf{q}}_i^b \otimes \bar{\mathbf{v}}^i \otimes [\bar{\mathbf{q}}_i^b]^{-1} = \begin{bmatrix} \mathbf{T}_i^b(\bar{\mathbf{q}})\mathbf{v}^i \\ 0 \end{bmatrix} .$$

In fact, for any quaternion $\bar{\mathbf{q}}$ it can be shown that

$$\bar{\mathbf{q}}_i^b \otimes \bar{\mathbf{q}} \otimes [\bar{\mathbf{q}}_i^b]^{-1} = \begin{bmatrix} \mathbf{T}_i^b\mathbf{q} \\ q_0 \end{bmatrix} . \tag{2.34}$$

Eq. 2.34 will be very useful for simplifying the quaternion measurement deviation equation in Chapter 5.

2.5 Dynamical Systems Theory

This section focuses on linear systems theory as it applies to the Kalman filter.

2.5.1 Linear System Representation

Given a linear dynamical system such as the spring-mass damper system in Figure 2.4, the equation of motion describing the system is

$$m\ddot{y} = -c\dot{y} - ky + u ,$$

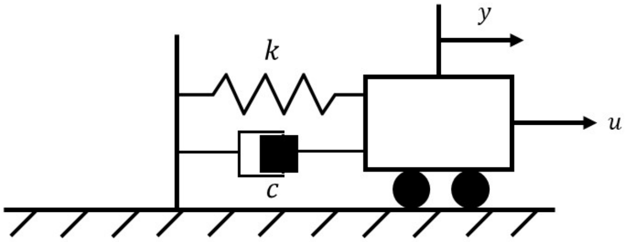


Figure 2.4: Spring-Mass-Damper System

where u is the force input. A more systematic way of representing this system behavior is through state-space representation. Re-arranging the equation so that $\ddot{y}$ is isolated on the left side by dividing by the mass m yields

$$\ddot{y} = -\frac{c}{m}\dot{y} - \frac{k}{m}y + \frac{u}{m} \, .$$

Now, let $x_1 = y$ and $x_2 = \dot{y}$, which also results in $\dot{x}_1 = x_2 = \dot{y}$ and $\dot{x}_2 = \ddot{y}$. The equation now becomes

$$\dot{x}_2 = -\frac{c}{m}x_2 - \frac{k}{m}x_1 + \frac{u}{m} \, .$$

This equation can now be represented in matrix form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u \, ,$$

where the output is defined as

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \, ,$$

assuming that y is the output being measured or observed in the system. In the case that $\dot{y}$ is being measured, then the output is defined as

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \, .$$

This yields the state-space representation of the system. A linear time-invariant dynamical system can be represented by the state-space representation,

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \, , \tag{2.35}$$

and

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \, , \tag{2.36}$$

where $\mathbf{A} \in \mathbb{R}^{nxn}$ represents the system dynamics and stability, $\mathbf{B} \in \mathbb{R}^{nxm}$ describes the control input mapping, and $\mathbf{C} \in \mathbb{R}^{nxm}$ describes the output mapping. Typically, **C** is determined by the engineer since the desired state to be measured as an output is determined by sensor choice. States such as position, velocity, and attitude can be measured as outputs by sensors. However, there are cases where **C** is not entirely determined by the engineer, rather it is influenced by the system itself and the capabilities of sensors. For example, given a system with the state

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} ,$$

if it is impossible to measure each state individually but possible to measure an output that consists of a combination of the states,

$$y_1 = x_1 + x_3 \quad \text{and} \quad y_2 = x_2 + x_4 ,$$

then the measurement mapping matrix that yields the output **y** from the state **x** is a constant matrix

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} .$$

In this scenario, the measurement mapping matrix was not fully determined by the engineer. A similar case is the output of a biased sensor. Since sensors are often corrupted with biases, the measurement often consists of the state element plus the corresponding bias. One way to account for these biases is pre-calibration. In some applicable cases the bias may be included in the state as part of the system and estimated through the

Kalman filter. However, biases are not always observable. This will be discussed in the subsequent chapter in more depth.

For a linear time-invariant system such as the system described above, the continuous-time solution to $\dot{\mathbf{x}}(t)$ is represented by the state transition matrix $\boldsymbol{\Phi}(t)$. In the case of a discrete system with time step Δt, this matrix propagates the state from a previous point in time t_{k-1} to the current point in time, $t_k = t_{k-1} + \Delta t$. Given a time-invariant system, the continuous time state transition matrix for a linear time-invariant system is found using the matrix exponential of the system matrix $\mathbf{A}$,

$$\Phi(t) = e^{\mathbf{A}(t-t_0)} \,. \tag{2.37}$$

The solution to the time-invariant system at any time $t \geq t_0$ is

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) \,.$$

The state transition matrix is an essential part of the linear Kalman filter structure and is utilized in the subsequent chapter.

2.5.2 Non-Linear System Representation

For a non-linear system, a linear, time-invariant state-space representation in Eq. 2.35 and 2.36 of dynamics is not often possible. For example, Euler's equations of rigid body rotation are

$$\begin{aligned}
I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2\omega_3 &= M_1 \,, \\
I_2\dot{\omega}_2 + (I_1 - I_3)\omega_3\omega_1 &= M_2 \,, \\
I_3\dot{\omega}_3 + (I_2 - I_1)\omega_1\omega_2 &= M_3 \,,
\end{aligned}$$

where $\omega_2\omega_3$, $\omega_3\omega_1$, and $\omega_1\omega_1$ represent non-linear terms. However, the dynamics can be linearized about an equilibrium point to describe the system behavior in a neighborhood around the equilibrium point. This allows state-space representation to be used for control and estimation purposes, but only within the small linear range. If we leave the linear range, we have to re-linearize the system. A non-linear system is linearized through the Jacobian function, which evaluates the partial derivative of $\dot{\mathbf{x}}$ with respect to $\mathbf{x}$. The system Jacobian matrix applied to a state vector takes the form,

$$\mathbf{F} = \begin{bmatrix} \frac{\partial \dot{x}_1}{\partial x_1} & \frac{\partial \dot{x}_1}{\partial x_2} & \cdots & \frac{\partial \dot{x}_n}{\partial x_n} \\ \frac{\partial \dot{x}_2}{\partial x_1} & \frac{\partial \dot{x}_1}{\partial x_2} & \cdots & \frac{\partial \dot{x}_n}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \dot{x}_n}{\partial x_1} & \frac{\partial \dot{x}_n}{\partial x_2} & \cdots & \frac{\partial \dot{x}_n}{\partial x_n} \end{bmatrix}_{\mathbf{x}\,=\,\mathbf{x}^*} , \qquad (2.38)$$

where $\mathbf{x}^*$ represents the equilibrium point. Note that $\mathbf{F}$ can also remain time-varying (although linear) and the Kalman filter methodology can accommodate. The same holds for the input control matrix $\mathbf{B}$ and output matrix $\mathbf{C}$. If the output of the system is a non-linear function of the state itself, then the Jacobian is applied to find the linearized measurement mapping matrix. The measurement mapping Jacobian matrix in terms of a state vector is

$$\mathbf{H} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}_{\mathbf{x}\,=\,\mathbf{x}^*} . \qquad (2.39)$$

For Euler's rigid body rotation with no control input, the state vector and Jacobian is

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \begin{bmatrix} \frac{I_2 - I_3}{I_1}\omega_2\omega_3 \\ \frac{I_3 - I_1}{I_2}\omega_1\omega_3 \\ \frac{I_1 - I_2}{I_3}\omega_1\omega_2 \end{bmatrix} \quad \text{and} \quad \mathbf{F} = \begin{bmatrix} 0 & \frac{I_2 - I_3}{I_1}\omega_3^* & \frac{I_2 - I_3}{I_1}\omega_2^* \\ \frac{I_3 - I_1}{I_2}\omega_3^* & 0 & \frac{I_3 - I_1}{I_2}\omega_1^* \\ \frac{I_1 - I_2}{I_3}\omega_2^* & \frac{I_1 - I_2}{I_3}\omega_1^* & 0 \end{bmatrix}_{\boldsymbol{\omega} = \boldsymbol{\omega}^*} .$$

The matrix is then linearized about an equilibrium point $\boldsymbol{\omega} = \boldsymbol{\omega}^*$ of the states where the state variables are chosen to be constant. For the purposes of simulating a control system, the equilibrium point can be the most recent estimate of the state. State transition can be computed numerically. This is repeated in a discrete fashion to provide an approximation of the solution. These concepts are applied to the extended Kalman filter, since the dynamics will be non-linear.

Chapter 3: Navigation Algorithm - The Kalman Filter

The backbone of the navigation algorithm is the Kalman filter. The Kalman filter is an algorithm that incorporates mathematical models, external measurements, and *a priori* state estimates, to compute current state estimates as accurately as possible. The Kalman filter is capable of estimating states that we cannot directly measure, such as sensor biases.

3.1 The Kalman Filter Structure

The Kalman filter employs two main sources of information: a state estimate and measurements of combinations of the state. In fact, the Kalman filter optimally weights external measurements and estimates of the measurements based on the available state estimate and determines the best state estimate based on measures of their associated uncertainties. The Kalman filter also relies heavily on mathematical models of the spacecraft dynamics, measurements, and error sources. For this reason, Kalman filter design process is as much of an art form as it is a science, because the mathematical models being used will not perfectly reflect reality. The more accurate and often complex the mathematical models are, the better the Kalman filter can generally estimate the state.

The Kalman filter algorithm flow is illustrated in Figure 3.1. To begin the process, initial conditions are chosen for the estimated state vector and the state estimation error covariance. There are different approaches to selecting the initial values, and most follow from knowledge of the problem at hand. After the initial conditions at time t_0 are chosen, the filter process begins. The state is propagated from the previous time step t_{k-1} to the current time step t_k, which is the prediction of the state using the mathematical

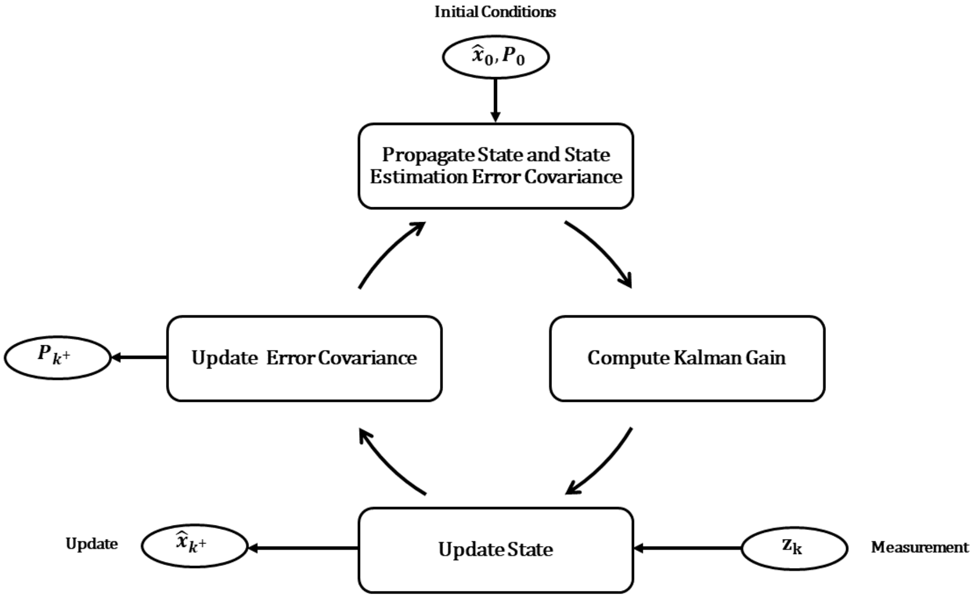


Figure 3.1: General Kalman Filter Flow

models that describe the system behavior. To propagate the spacecraft state, a solution (or approximation of the solution) to the differential equations that represent the state dynamics is required. Since the dynamics can be linear or non-linear, there are different ways to propagate the state. Several methods will be discussed. Once the state is propagated, from t_{k-1} to t_k, the Kalman gain is computed just prior to incorporating the external measurement. With the measurement and state estimate at t_k, an update to the state estimate and state estimation error covariance matrix is made. We denote the state estimate at t_k just prior to the measurement update as $\hat{\mathbf{x}}_k^-$ and the associated error covariance as $\mathbf{P}_k^-$. After the update, we denote the state estimate and state estimation error covariance as $\hat{\mathbf{x}}_k^+$ and $\mathbf{P}_k^+$, respectively. The update involves taking the difference between the measurements and the predicted measurement, defined as the measurement residual, to correct and update the state estimate. Once the state estimated is updated, the state estimation error covariance is updated as well. This process is then repeated in a loop.

The state estimation error covariance matrix is a measure of accuracy of the state estimate. A "large" covariance implies that the Kalman filter is uncertain about the state estimate accuracy, where as a "small" state estimation error covariance indicates that the Kalman filter is more certain that the Kalman filter estimate is accurate. Like the estimated state, the state estimation error covariance is propagated and updated once a measurement is available. Each diagonal element of the state estimation error covariance matrix represents the uncertainty for the corresponding state element. The square root of the state estimation error covariance matrix diagonal elements represents the $\pm 1\sigma$ of the associated estimation error.

When designing Kalman filters, the state propagation is typically accomplished at a much higher frequency than the state update rate. This is because sensors sample at much lower frequencies than that of the embedded processor internal or external clock source, as well as the higher sampling rate of IMU. For example, a microprocessor may be able to propagate the state estimate and state estimation error covariance at a frequency of 100 Hz while the sensor can only output measurements at a frequency of 10 Hz. If no measurements are available, the Kalman filter will continue to propagate the state vector and state estimation error covariance forward in time without updating. This is known as dead reckoning. If the state vector continues to propagate through time without a measurement update, it will accumulate error and propagate that error through time and eventually diverge from the true state vector trajectory.

3.2 Kalman Filter - Linear Dynamics

Figure 3.2 illustrates the algorithm for a linear Kalman filter. The state transition matrix, denoted by $\mathbf{\Phi}_k$, is the solution to the differential equations that represent the state dynamics. The state transition matrix is then used to propagate the state vector and state estimation error covariance between measurements. Prior to an external measurement update, the Kalman gain is computed. Once an external measurement is available, the state vector and state estimation error covariance are update using the Kalman gain. The general model for the Kalman filter is derived in the subsequent sections.

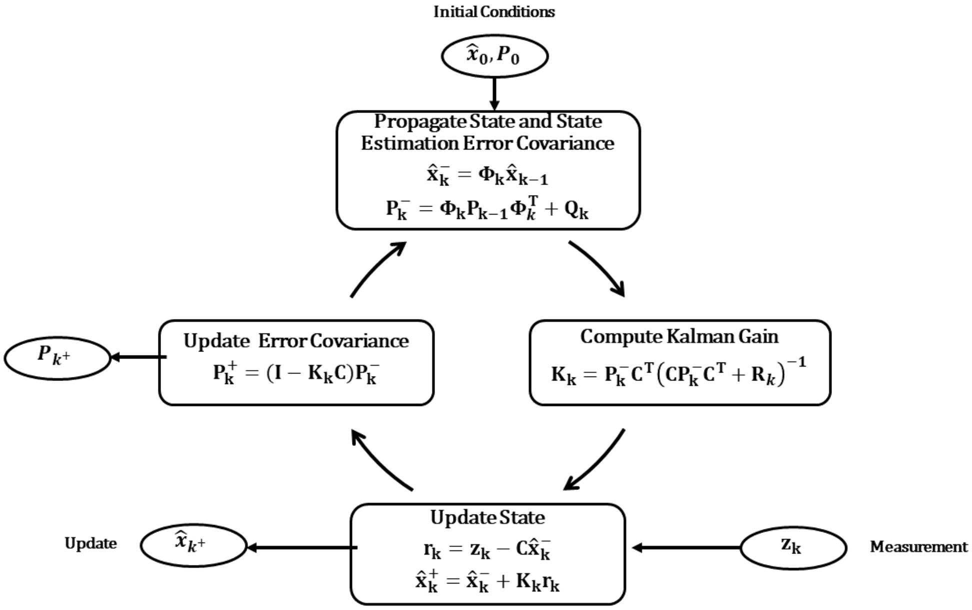

Initial Conditions
$\hat{x}_0, P_0$
Propagate State and State Estimation Error Covariance
$\hat{x}_k^- = \Phi_k \hat{x}_{k-1}$
$P_k^- = \Phi_k P_{k-1} \Phi_k^T + Q_k$
Update Error Covariance
$P_k^+ = (I - K_k C) P_k^-$
P_{k^+}
Compute Kalman Gain
$K_k = P_k^- C^T (C P_k^- C^T + R_k)^{-1}$
Update
$\hat{x}_{k^+}$
Update State
$r_k = z_k - C\hat{x}_k^-$
$\hat{x}_k^+ = \hat{x}_k^- + K_k r_k$
z_k
Measurement

Figure 3.2: Linear Kalman Filter Algorithm

3.2.1 Propagating Between Measurements

For a linear Kalman filter the system model is given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{G}\mathbf{w}(t)\ . \tag{3.1}$$

The discrete form of the system is

$$\mathbf{x}_k = \mathbf{\Phi}_k \mathbf{x}_{k-1} + \mathbf{\Gamma}_k \mathbf{w}_k\ . \tag{3.2}$$

Note that $\mathbf{G}$ and $\mathbf{\Gamma}_k$ are process noise mapping matrices, typically constant and time-invariant. The measurement model is

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \eta_k\ . \tag{3.3}$$

Here $\mathbf{w}_k$ and $\mathbf{w}(t)$ represent the model uncertainty (process noise). Since we cannot model our system perfectly, we include process noise as a tuning parameter to achieve acceptable state estimation results. The system model in Eq. 3.2 assumes that the process noise $\mathbf{w}_k$ is a zero-mean, white noise sequence with

$$E\{\mathbf{w}_k\} = \mathbf{0}\,, \qquad E\{\mathbf{w}_k\mathbf{w}_j^T\} = \mathbf{Q}_k\delta_{jk}\,.$$

And for the system in Eq. 3.1 we assume $\mathbf{w}(t)$ is a zero-mean, white noise sequence with

$$E\{\mathbf{w}(t)\} = \mathbf{0}\;\forall\; t\,, \qquad E\{\mathbf{w}(t)\mathbf{w}^T(\tau)\} = \mathbf{Q}\delta(t-\tau)\;\forall\; t,\tau\,.$$

The Kronecker delta function δ_{kj} and the Dirac delta function $\delta(t-\tau)$ indicate that the sequences $\mathbf{w}_k$ and $\mathbf{w}(t)$ are uncorrelated in time, i.e.,

$$E\{\mathbf{w}_k\mathbf{w}_j^T\} = \begin{cases} \mathbf{Q}_k\,, & k = j \\ \mathbf{0}\,, & k \neq j \end{cases}\,,$$

and

$$E\{\mathbf{w}(t)\mathbf{w}^T(\tau)\} = \begin{cases} \mathbf{Q}\,, & t = \tau \\ \mathbf{0}\,, & t \neq \tau \end{cases}\,.$$

The measurement model assumes that the measurement noise $\boldsymbol{\eta}_k$ is a zero-mean, white noise sequence with covariance $\mathbf{R}_k$. That is,

$$E\{\boldsymbol{\eta}_k\} = \mathbf{0}\,, \qquad E\{\boldsymbol{\eta}_k\boldsymbol{\eta}_j^T\} = \mathbf{R}_k\delta_{kj}\,,$$

or,

$$E\{\boldsymbol{\eta}_k\boldsymbol{\eta}_j^T\} = \begin{cases} \mathbf{R}_k\,, & k = j \\ \mathbf{0}\,, & k \neq j \end{cases}\,.$$

The relationship between $\mathbf{Q}$ and $\mathbf{Q}_k$is given by

$$\mathbf{\Gamma}_k \mathbf{Q}_k \mathbf{\Gamma}_k^T = \int_{t_{k-1}}^{t_k} \mathbf{\Phi}(t_k, \tau) \mathbf{G} \mathbf{Q} \mathbf{G}^T \mathbf{\Phi}^T (t_k, \tau) d\tau \, . \tag{3.4}$$

Analyzing the first block in Figure 3.2, the state transition matrix $\mathbf{\Phi}_k$ propagates the previous state estimate $\hat{\mathbf{x}}_{k-1}^+$ forward in time to the current time step to obtain $\hat{\mathbf{x}}_k^-$. The state transition matrix computation for time-invariant systems has the form $\mathbf{\Phi}_k = e^{\mathbf{A}\Delta t}$ where $\Delta t = t_k - t_{k-1}$. For the time-variant case, the state transition matrix is computed through numerical integration.

Using the initial conditions $\mathbf{x}_0$ and $\mathbf{P}_0$, we propagate the state vector and state estimation error covariance forward to the next available measurement. For the initial conditions $(\hat{\mathbf{x}}_0, \mathbf{P}_0)$ required to initialize the filter, we have

$$E\{\mathbf{x}_0\} = \hat{\mathbf{x}}_0 \quad \text{and} \quad E\{(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T\} = \mathbf{P}_0 \, .$$

For our initial estimate of $\hat{\mathbf{x}}_0$, there is an associated uncertainty $\mathbf{P}_0$. When initializing our filter, we must do our best to encapsulate our uncertainty of $\hat{\mathbf{x}}_0$ within $\mathbf{P}_0$. If we choose a very large initial uncertainty $\mathbf{P}_0$ so that we may never make an estimate with an error beyond the bounds of $\mathbf{P}_0$ then we have safely initialized our filter with the potential consequence of a larger estimation error or convergence time. Alternatively, if we fail to encapsulate our initial estimate within the bounds of our initial uncertainty, our filter may diverge and performance will be degraded. The tuning process, along with knowledge of the problem at hand, can help determine a good value for initializing the covariance.

Once the state transition matrix, $\mathbf{\Phi}_k$, has been computed, the state vector and state estimation error covariance are propagated forward to the current time step. The state is propagated via

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_{k-1}^+ \, , \tag{3.5}$$

and the state estimation error covariance is propagated via

$$\mathbf{P}_k^- = \mathbf{\Phi}_k \mathbf{P}_{k-1}^+ \mathbf{\Phi}_k^T + \mathbf{Q}_k \, , \tag{3.6}$$

where

$$\mathbf{P}_k^- = E\{(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T\} \, .$$

An example of the use of process noise is estimating the altitude of an airplane flying through a patch of unexpected turbulence or the position of a car driving over an unexpected bumpy patch on a road. If we model the car or airplane dynamics to describe a trajectory without regard to disturbances, then the state estimation error covariance will not include the effects of the external disturbances. Insight into how much model uncertainty is to be expected is essential to the performance of the Kalman filter. Therefore, $\mathbf{Q}_k$ can be viewed as a tuning parameter. A large $\mathbf{Q}_k$ means that there is a large uncertainty in the model. A $\mathbf{Q}_k$ of zero means that the system is modeled perfectly. Setting $\mathbf{Q}_k = 0$ will lead to the optimal gain $\mathbf{K}_k \rightarrow \mathbf{0}$ as $k \rightarrow \infty$, which is highly undesirable.

3.2.2 Kalman Gain

The Kalman filter optimally weights the estimated states and the available measurements. The Kalman gain is a function of the propagated covariance $\mathbf{P}_k^-$, the measurement mapping matrix $\mathbf{C}$, and the sensor noise covariance matrix $\mathbf{R}_k$. The Kalman gain is given by

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}^T (\mathbf{C}\mathbf{P}_k^- \mathbf{C}^T + \mathbf{R}_k)^{-1} \, . \tag{3.7}$$

3.2.3 Update

Once a measurement is available, the measurement residual is computed. The measurement residual is the difference between the external measurement and the predicted external measurement, as

$$\mathbf{r}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k \,. \tag{3.8}$$

The predicted external measurement is computed using the the measurement mapping matrix $\mathbf{C}$ and the predicted state $\hat{\mathbf{x}}_k^-$, as

$$\hat{\mathbf{y}}_k = \mathbf{C}\hat{\mathbf{x}}_k^- \,. \tag{3.9}$$

After the residual is computed, the state estimate is updated via

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\mathbf{r}_k \,, \tag{3.10}$$

and the state estimation error covariance updates via

$$\hat{\mathbf{P}}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{C})\hat{\mathbf{P}}_k^- \,, \tag{3.11}$$

where

$$\mathbf{P}_k^+ = E\{(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)^T\} \,,$$

and $\mathbf{K}_k$ is given in Eq. 3.7. If measurements are not available, the propagated state estimate will over time accumulate error and cause the state estimation error to grow. As measurements are processed, the estimate is updated and converges towards the true trajectory. The Kalman filter timeline is illustrated in Figure 3.3.

3.2.4 Example 1 - Linear Kalman Filter

Consider the dynamics of a system, given by

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ 0 \end{bmatrix} \,. \tag{3.12}$$

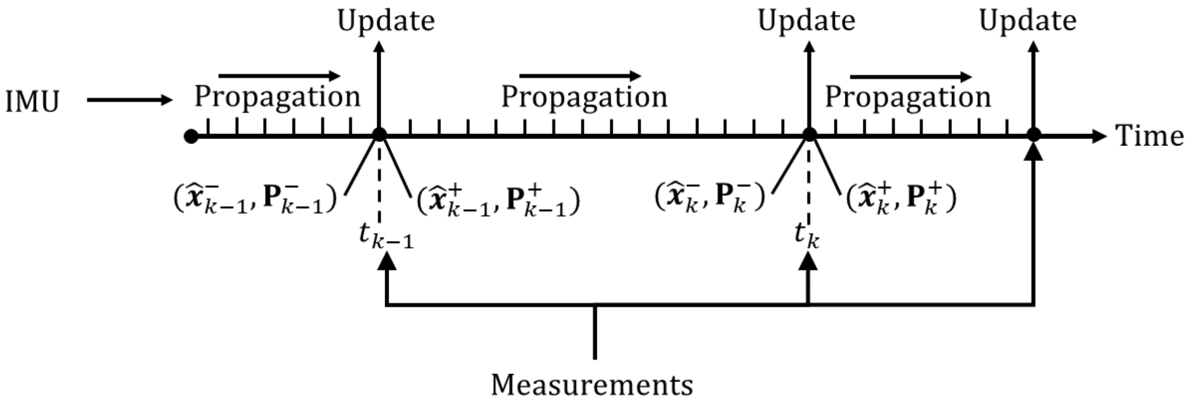

Update
Update
Update
IMU
Propagation
Propagation
Propagation
Time
$(\hat{x}_{k-1}^-, \mathbf{P}_{k-1}^-)$
$(\hat{x}_{k-1}^+, \mathbf{P}_{k-1}^+)$
$(\hat{x}_k^-, \mathbf{P}_k^-)$
$(\hat{x}_k^+, \mathbf{P}_k^+)$
t_{k-1}
t_k
Measurements

Figure 3.3: Kalman Filter Algorithm Timeline

Assume we can measure x_1 and x_2. We have the output

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k \,,$$

where

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} .$$

Consider that we add process noise to account for uncertainty in the model. Since $\dot{x}_1 = x_2$ is a perfect model of position, we set the process noise for that state element to zero. Velocity as shown in Eq. 3.12 may not be perfectly known so process noise is added. The Kalman filter model is then given as

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t) \,, \tag{3.13}$$

where $E\{w(t)\} = 0 \;\forall\; t$ and $E\{w(t)w(\tau)\} = Q\delta(t-\tau) \;\forall\; t, \tau$. Since $\mathbf{A}$ is constant, this system is time-invariant, therefore the state transition matrix is

$$\mathbf{\Phi}_k = e^{\mathbf{A}\Delta t} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} ,$$

where $\Delta t = t_k - t_{k-1}$. The discrete state solution is

$$\begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,k-1} \\ x_{2,k-1} \end{bmatrix} = \begin{bmatrix} x_{1,k-1} + x_{2,k-1}\Delta t \\ x_{2,k-1} \end{bmatrix} .$$

Then, from Eq. 3.2 the discrete Kalman filter system model is

$$\mathbf{x}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k \,,$$

where

$$E\{\mathbf{w}_k \mathbf{w}_k^T\} = \mathbf{Q}_k = Q \begin{bmatrix} \frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^2 & \Delta t \end{bmatrix} ,$$

and Q is the PSD of $w(t)$. The Kalman filter measurement model is

$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \boldsymbol{\eta}_k \,,$$

where

$$E\{\boldsymbol{\eta}_k\} = 0 \; \forall \; t \,, \qquad \text{and} \qquad E\{\boldsymbol{\eta}_k \boldsymbol{\eta}_j^T\} = \mathbf{R}_k \delta_{k,j} \; \forall \; k, j \,.$$

The state is propagated at 50Hz and the measurements are sampled at 5Hz. The simulation time is 100 seconds.

To gain insight into the Kalman filter performance, the state estimation error covariance and state error are plotted and observed along with the state estimate. The input parameters for this example are

$$\mathbf{R}_k = \begin{bmatrix} 9 & 0 \\ 0 & 0.01 \end{bmatrix} , \qquad \text{and} \qquad Q = 0.00001 \,,$$

with the initial conditions,

$$\mathbf{P}_0 = \begin{bmatrix} 12 & 0 \\ 0 & 0.02 \end{bmatrix}, \quad \text{and} \quad \mathbf{x}_0 = \begin{bmatrix} -2.2482 \\ 0.9082 \end{bmatrix},$$

and $\Delta t = 0.02\ s$. The simulated results are shown in Figures 3.4 and 3.5, which illustrate a comparison between the measured, true, and estimated state, as well as the state estimation error and error covariance. Through the state estimation error covariance plots it is shown that both states are observable, as indicated by the reduction in the state estimation error covariance between t_0 and t.

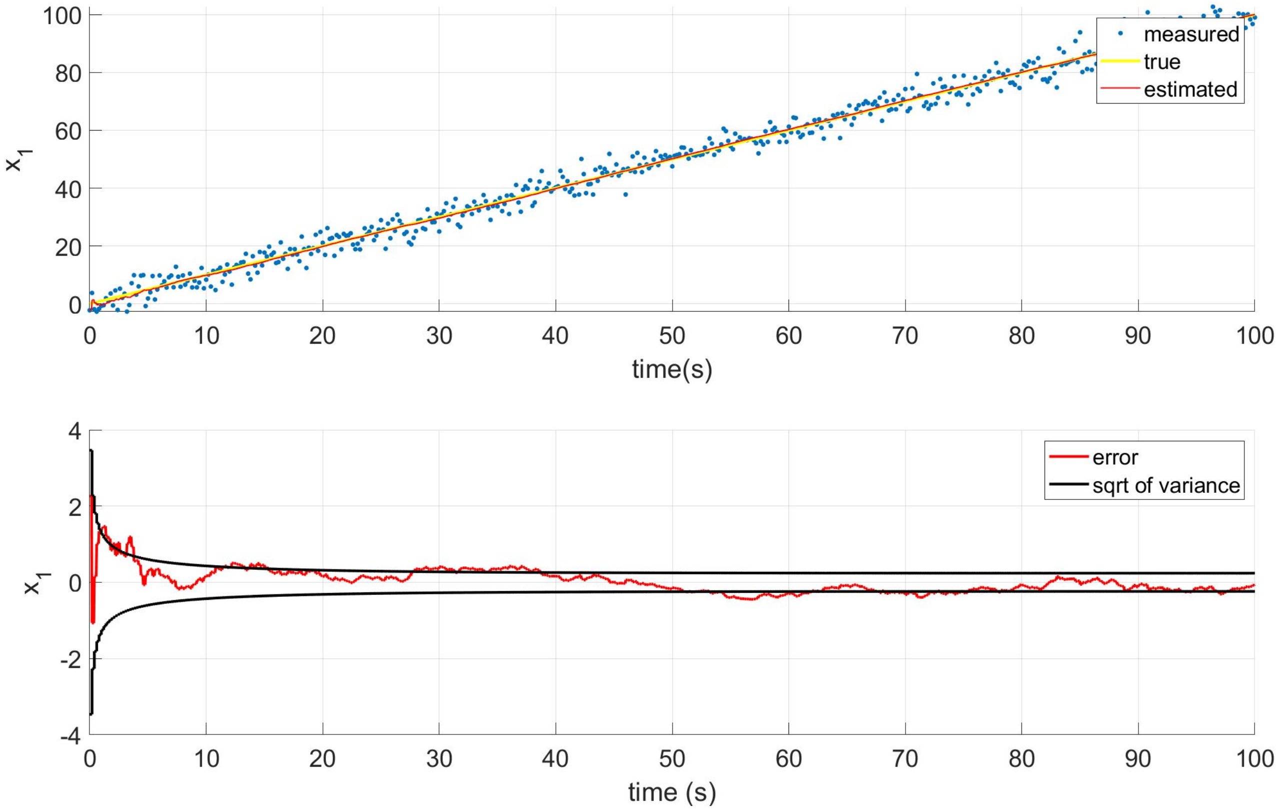


Figure 3.4: Example 1 - x_1

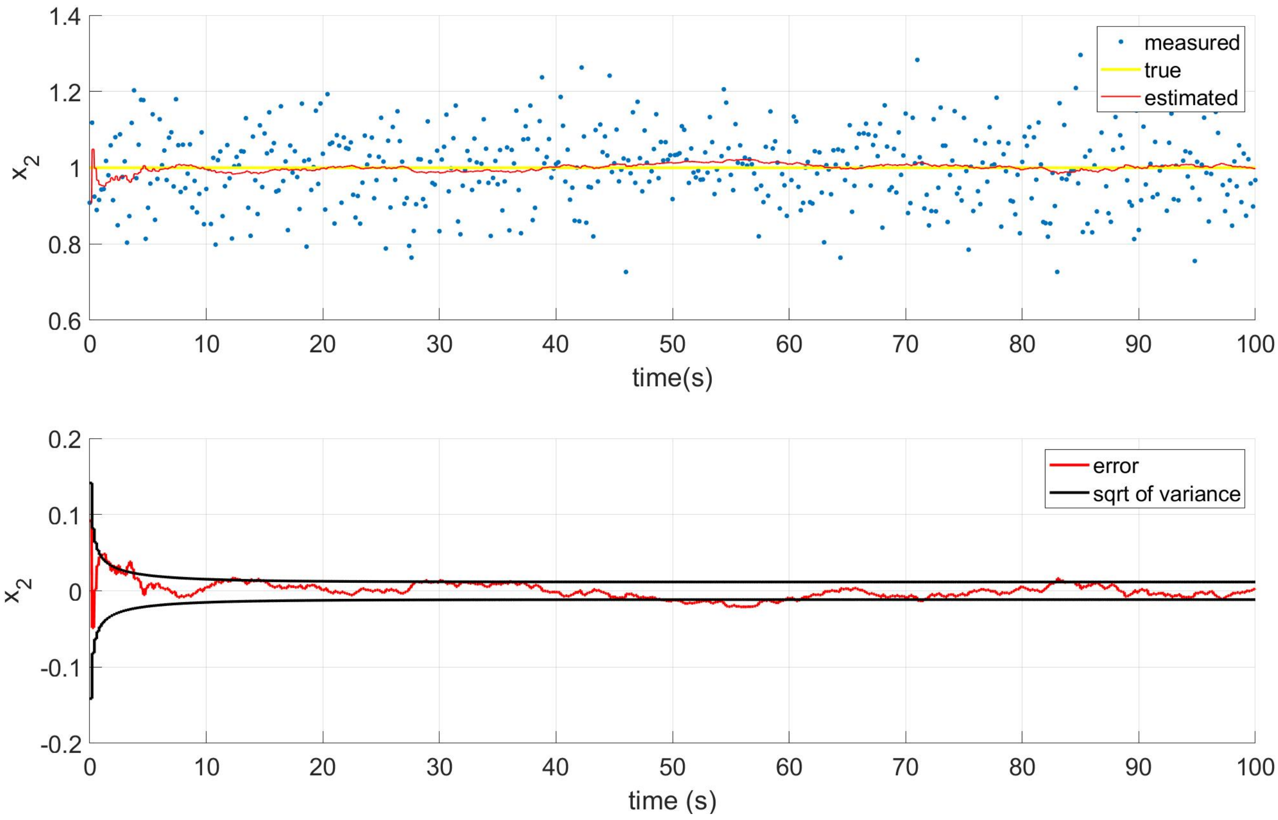


Figure 3.5: Example 1 - x_2

3.2.5 Example 2 - Linear Kalman Filter with Bias

Returning to Example 1, we now look at the scenario where the sensors are not only corrupted with random noise, but also random constant biases. Biases are modeled as

$$\dot{\beta} = 0\,,$$

with

$$E\{\beta\} = 0\,, \qquad E\{\beta\beta^T\} = P_\beta\,.$$

If the biases were not constant and did exhibit change over time then they could be modeled as a random walk through the use of process noise. The system model remains the same with the addition of the biases to the state vector so that they can be estimated and accounted for to ensure optimal state estimation. The augmented state is

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T ,$$

where $x_3 = \beta_{x_1}$ and $x_4 = \beta_{x_2}$. The model is given by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} ,$$

and the output

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k ,$$

where

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} .$$

From Eq. 3.1 Kalman filter model is given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{G}w(t) ,$$

where

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T .$$

The the state transition matrix is

$$\mathbf{\Phi}_k = e^{\mathbf{A}\Delta t} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ,$$

where $\Delta t = t_k - t_{k-1}$. From Eq. 3.2, the discrete Kalman filter system model is given as

$$\mathbf{x}_k = \mathbf{\Phi}_k \mathbf{x}_{k-1} + \mathbf{w}_k \; ,$$

where

$$E\{\mathbf{w}_k\} = 0 \; \forall \; t \; , \qquad \text{and} \qquad E\{\mathbf{w}_k \mathbf{w}_j^T\} = \mathbf{Q}_k \delta_{k,j} \; \forall \; k, j \; .$$

The Kalman filter measurement model is given by

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \boldsymbol{\eta}_k \; ,$$

where

$$E\{\boldsymbol{\eta}_k\} = 0 \; \forall \; t \; , \qquad \text{and} \qquad E\{\boldsymbol{\eta}_k \boldsymbol{\eta}_j^T\} = \mathbf{R}_k \delta_{k,j} \; \forall \; k, j \; .$$

From Eq. 3.4 the process noise matrix is given as

$$\mathbf{Q}_k = Q \begin{bmatrix} \frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 & 0 & 0 \\ \frac{1}{2}\Delta t^2 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} ,$$

where Q is the PSD of $w(t)$.

The input parameters are

$$\mathbf{R}_k = \begin{bmatrix} 9 & 0 \\ 0 & 0.01 \end{bmatrix} , \qquad \text{and} \qquad Q = 0.00001 \; ,$$

with the initial conditions,

$$\mathbf{P}_0 = \begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}, \quad \text{and} \quad \mathbf{x}_0 = \begin{bmatrix} -3.0338 \\ 0.8304 \\ -1.4075 \\ 0.1337 \end{bmatrix},$$

and $\Delta t = 0.02\ s$. The simulated results are shown in Figures 3.6 and 3.8 which illustrate a comparison between the measured, true, and estimated state elements x_1 and x_2, as well as the state estimation error and error covariance for all elements. One difference in this example is that the position bias covariance does not experience any reduction in uncertainty. Therefore, the position bias is unobservable. Since the bias is unobservable, the position estimation error covariance exhibits an increase in magnitude.

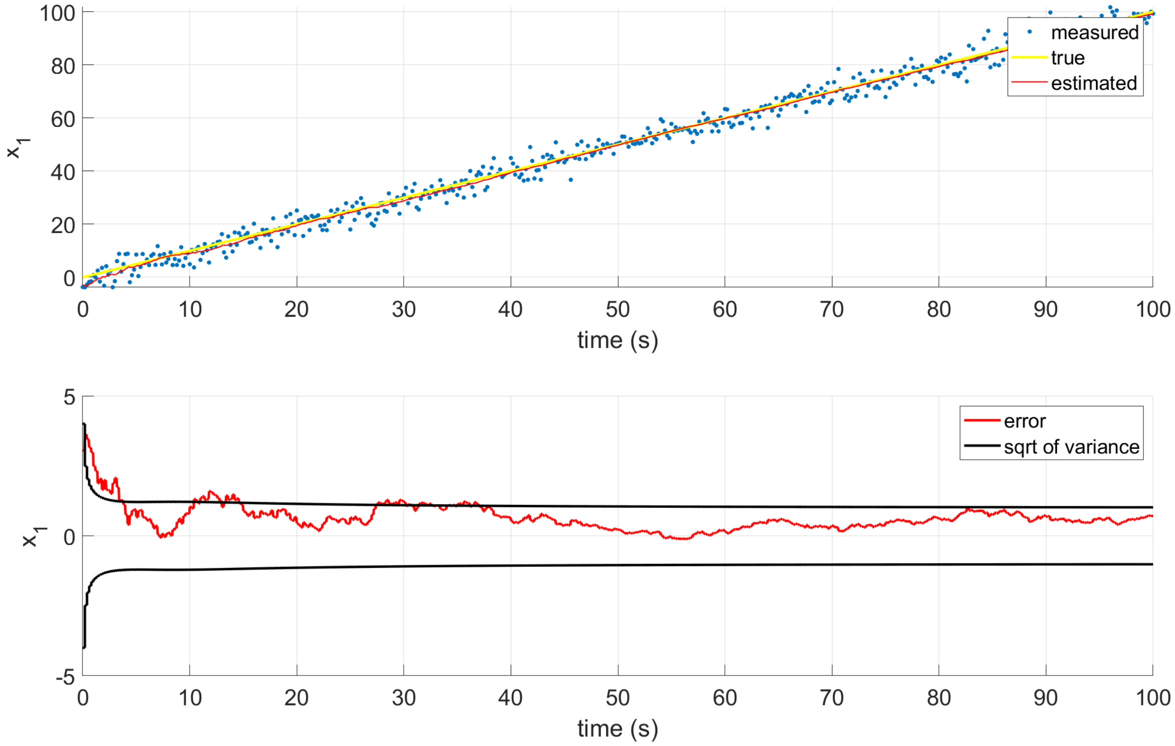


Figure 3.6: Example 2 - x_1

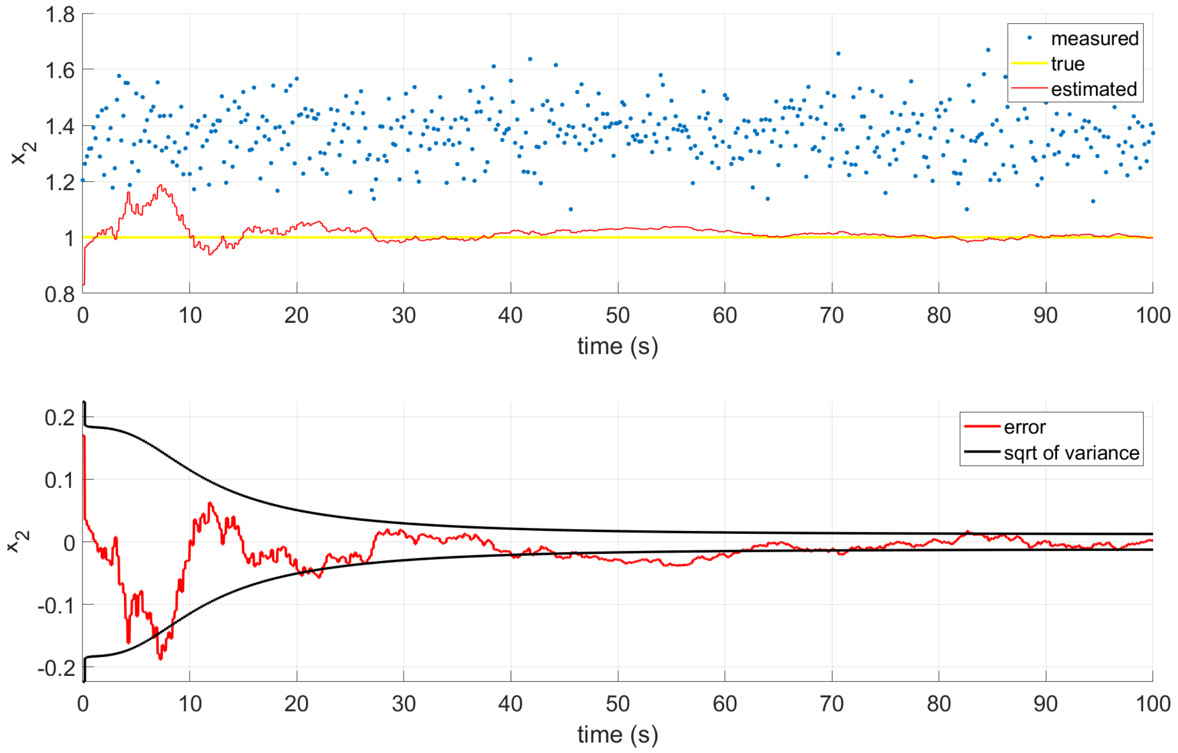


Figure 3.7: Example 2 - x_2

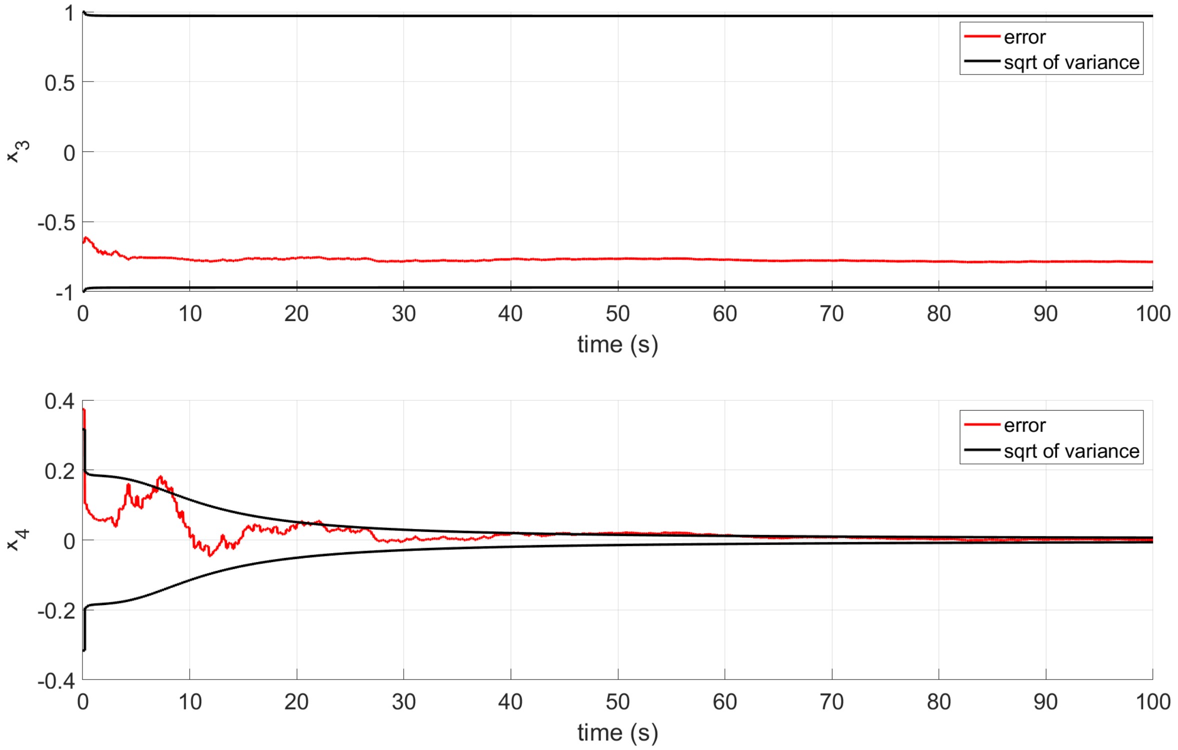


Figure 3.8: Example 2 - Biases

3.3 Extended Kalman Filter - Non-Linear Dynamics

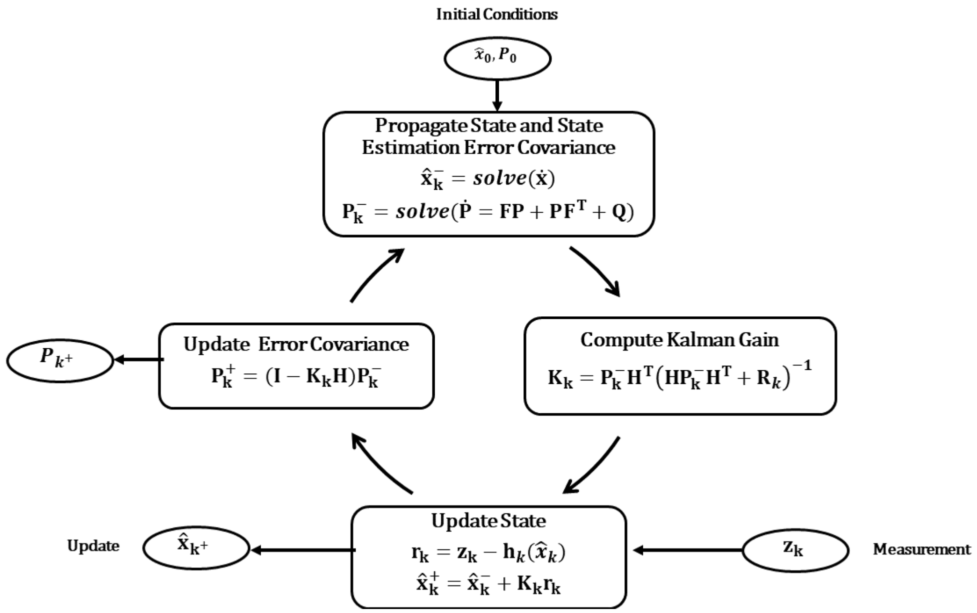

Initial Conditions
$\hat{x}_0, P_0$
Propagate State and State Estimation Error Covariance
$\hat{\mathbf{x}}_k^- = solve(\dot{\mathbf{x}})$
$\mathbf{P}_k^- = solve(\dot{\mathbf{P}} = \mathbf{FP} + \mathbf{PF}^T + \mathbf{Q})$
Compute Kalman Gain
$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{HP}_k^- \mathbf{H}^T + \mathbf{R}_k)^{-1}$
Update Error Covariance
$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-$
P_{k^+}
Update State
$\mathbf{r}_k = \mathbf{z}_k - \mathbf{h}_k(\hat{x}_k)$
$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\mathbf{r}_k$
Update
$\hat{\mathbf{x}}_{k^+}$
$\mathbf{z}_k$
Measurement

Figure 3.9: Non-Linear Kalman Filter Procedure

Figure 3.9 illustrates the algorithm for a non-linear Kalman filter, also known as an extended Kalman filter. Due to the non-linearity of the dynamics, we utilize numerical methods to approximate solutions to the differential equations. We employed the Runge-Kutta 4th order (RK4) tool to numerically integrate the state estimate and state estimation error covariance.

3.3.1 Propagating Between Measurements

For an extended Kalman filter, the system model is

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), t) + \mathbf{w}(t)\,, \tag{3.14}$$

where $\mathbf{w}(t)$ is a zero-mean, white noise process with

$$E\{\mathbf{w}(t)\} = \mathbf{0}\,, \qquad \text{and} \qquad E\{\mathbf{w}(t)\mathbf{w}(\tau)^T\} = \mathbf{Q}\delta(t-\tau)\ \forall\, t,\tau\,.$$

The measurement model takes the form

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}(t_k)) + \boldsymbol{\eta}_k \qquad , \tag{3.15}$$

where the measurement model assumes that the measurement noise $\boldsymbol{\eta}_k$ is a zero-mean, white noise sequence of covariance $\mathbf{R}_k$, and

$$E\{\boldsymbol{\eta}_k\} = \mathbf{0}\,, \qquad \text{and} \qquad E\{\boldsymbol{\eta}_k\boldsymbol{\eta}_j^T\} = \mathbf{R}_k\delta_{kj}\,.$$

For propagation between measurements, a numerical tool, such as the RK4 algorithm, would suffice for both the estimated state and state estimation error covariance. Therefore, for the *a priori* state estimate,

$$\hat{\mathbf{x}}_k^- = solve\Big(\mathbf{f}(\hat{\mathbf{x}}(t), t)\Big)\,, \qquad \text{for} \qquad t_{k-1} \leq t \leq t_k\,,$$

with the initial condition $\hat{\mathbf{x}}(t_{k-1}) = \hat{\mathbf{x}}_{k-1}^+$. To obtain the *a priori* state estimation error covariance estimate, we use the RK4 algorithm to numerically integrate the state estimation error covariance differential equation,

$$\mathbf{P}_k^- = solve\Big(\mathbf{F}(\hat{\mathbf{x}}(t), t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(\hat{\mathbf{x}}(t), t) + \mathbf{Q}\Big)\,, \qquad \text{for} \qquad t_{k-1} \leq t \leq t_k\,, \tag{3.16}$$

with $\mathbf{P}(t_{k-1}) = P_{k-1}^+$, where the Jacobian matrix $\mathbf{F}$ is obtained by linearizing the state differential equations about the most recent estimate of the state, given by

$$\mathbf{F}(\hat{\mathbf{x}}(t), t) = \left.\frac{\partial \mathbf{f}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)}\right|_{\mathbf{x}(t) = \hat{\mathbf{x}}(t)} \,. \tag{3.17}$$

3.3.2 Kalman Gain

The Kalman gain equation retains the same form with the exception that it is now using the Jacobian matrix,

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T(\hat{\mathbf{x}}_k^-) \left[\mathbf{H}_k(\hat{\mathbf{x}}_k^-) \mathbf{P}_k^- \mathbf{H}_k^T(\hat{\mathbf{x}}_k^-) + \mathbf{R}_k \right]^{-1} , \tag{3.18}$$

where the Jacobian matrix $\mathbf{H}$ is obtained by linearizing the measurement equations about the most recent estimate of the state,

$$\mathbf{H}_k(\hat{\mathbf{x}}_k^-) = \left. \frac{\partial \mathbf{h}_k(\mathbf{x}(t_k))}{\partial \mathbf{x}(t_k)} \right|_{\mathbf{x}(t_k) = \hat{\mathbf{x}}_k^-} . \tag{3.19}$$

3.3.3 Update

The update stage retains the same form for the extended Kalman filter. The measurement residual is

$$\mathbf{r}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k .$$

The predicted output, $\hat{\mathbf{y}}_k$, is estimated by evaluating the measurement equation with the most recent state estimate

$$\hat{\mathbf{y}}_k = \mathbf{h}_k(\hat{\mathbf{x}}_k^-) .$$

After the residual is computed, the state estimate is then updated via

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{r}_k ,$$

and the state estimation error covariance via

$$\hat{\mathbf{P}}_k^+ = \left[\mathbf{I} - \mathbf{K}_k \mathbf{H}_k(\hat{\mathbf{x}}_k^-) \right] \hat{\mathbf{P}}_k^- .$$

3.3.4 Example 3 - Extended Kalman Filter

Consider the following second-order non-linear differential equation representing an object in free fall,

$$\ddot{r} = \frac{-\mu}{r^2}\,, \tag{3.20}$$

where μ is the gravitational constant. If we can measure the position and velocity of this object directly, our measurement mapping equation is linear and given by

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} .$$

The second-order differential equation in Eq. 3.20 can be represented as two first-order differential equations, as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ \frac{-\mu}{r^2} \end{bmatrix} .$$

While this is a second-order differential equation, we have decomposed it into two first-order equations. We then numerically solve each equation separately.

$$r_k = \Delta t v_{k-1} + r_{k-1} \qquad \text{and} \qquad v_k = -\Delta t \frac{\mu}{r_k^2} + v_{k-1}\,.$$

The Jacobian of the state differential equations is

$$\mathbf{F} = \begin{bmatrix} \frac{\partial \dot{r}}{\partial r} & \frac{\partial \dot{r}}{\partial v} \\ \frac{\partial \dot{v}}{\partial r} & \frac{\partial \dot{v}}{\partial v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{2\mu}{r^3} & 0 \end{bmatrix} ,$$

where the system is re-linearized at each time step about $r = \hat{r}_k^-$ after propagating the state.

The input parameters for the filter shown in the plots are

$$\mathbf{R}_k = \begin{bmatrix} 9 & 0 \\ 0 & 0.01 \end{bmatrix}, \quad \text{and} \quad \mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & 0.00001 \end{bmatrix},$$

with the initial conditions,

$$\mathbf{P}_0 = \begin{bmatrix} 12 & 0 \\ 0 & 0.02 \end{bmatrix}, \quad \text{and} \quad \mathbf{x}_0 = \begin{bmatrix} 6378098.053 \\ 499.9082156 \end{bmatrix}.$$

The state estimation error covariance is depicted in Figure 3.10.

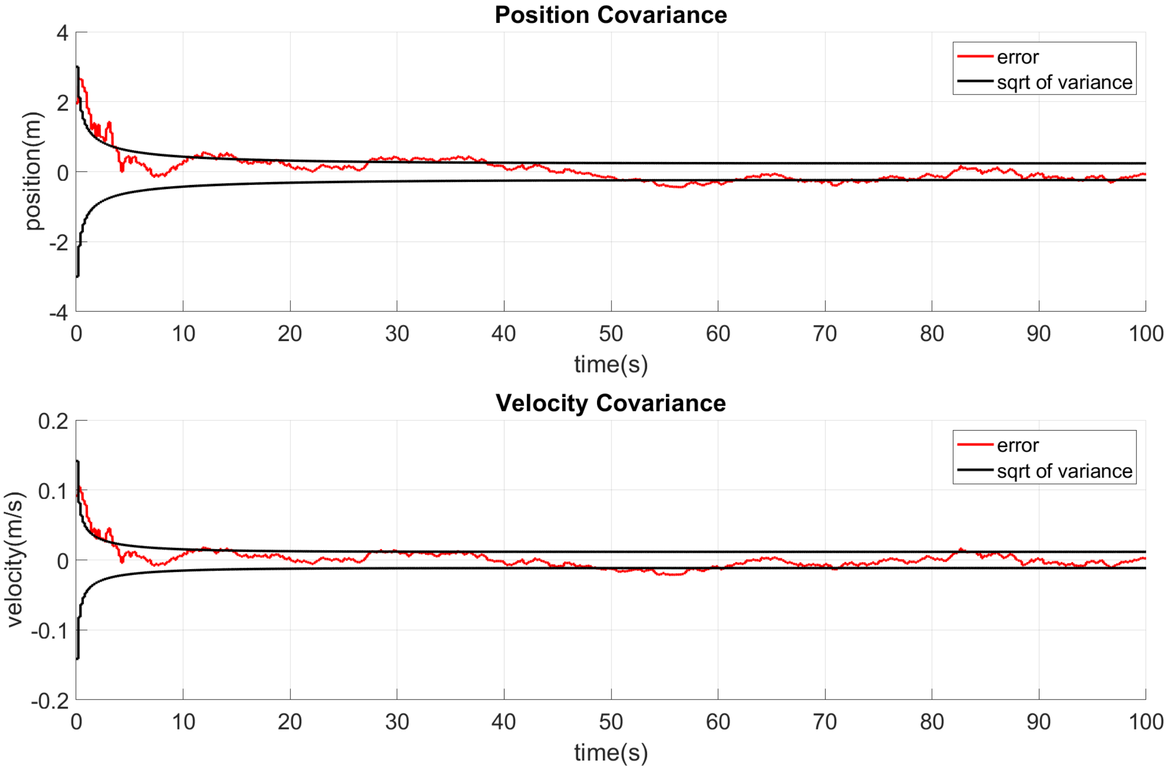


Figure 3.10: Example 3 - Position and Velocity Covariance

Chapter 4: State Dynamics and Equations of Motion

To estimate the state of our spacecraft, the Kalman filter relies on a mathematical model, typically represented by differential equations, to represent the vehicle dynamics. This mathematical model will never be perfect, yet the overall accuracy of the Kalman filter state estimates will largely depend on this model, hence this model is very important. However, we have an IMU that can provide us with the rate of change of our velocity (non-gravitational) and the rate of change of our attitude (angular velocity). The IMU directly measures non-gravitational acceleration and we can obtain gravitational acceleration from a two-body gravity model given a position estimate. The IMU also directly measures angular velocity so we can compute the attitude rate of change. Using the IMU, we can propagate our state forward through time to provide *a priori* estimates in between state updates in lieu of mathematical models of the external accelerations in conjunction with a gravity model.

As we are using the IMU to propagate position, velocity, and attitude, we will estimate the position, velocity, and attitude of the IMU, not the center of gravity of the body. We can readily obtain information about the center of gravity state after obtaining estimates of the IMU state.

4.1 State Dynamics

The three main state components of interest are $\mathbf{r}^i_{imu}$, $\mathbf{v}^i_{imu}$ and $\bar{\mathbf{q}}^b_i$. These state components will also be affected by the systematic errors of the IMU, therefore these errors must also be accounted for by augmenting our state vector to include them.

4.1.1 Position

The rate of change of the position of the IMU is

$$\dot{\mathbf{r}}^i_{imu} = \mathbf{v}^i_{imu} \, . \tag{4.1}$$

4.1.2 Velocity

The rate of change of the velocity of the IMU is

$$\dot{\mathbf{v}}^i_{imu} = \mathbf{a}^i_{imu,g} + \mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)\mathbf{T}^b_c\mathbf{a}^c_{imu,ng} \, , \tag{4.2}$$

where $\mathbf{a}^i_g$ is computed via a mathematical gravity model and $\mathbf{a}^i_{imu,ng}$ is provided by the IMU. Since $\mathbf{a}^i_{imu,ng}$ exists in the IMU case reference frame, it must be transformed into the inertial reference frame. To this end, we transform $\mathbf{a}^c_{imu,ng}$ to the spacecraft body reference frame via $\mathbf{T}^b_c$ and from the spacecraft body reference frame to the inertial reference frame via $\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)$. The matrix $\mathbf{T}^b_c$ is assumed to be known and constant. We will also make the assumption that the gravitational acceleration of the IMU and the gravitational acceleration of the spacecraft center of gravity are approximately equivalent due to the small relative distance between them. Therefore,

$$\mathbf{a}^i_{imu,g} \approx \mathbf{a}^i_g \, .$$

The velocity differential equation then becomes

$$\dot{\mathbf{v}}^i_{imu} = \mathbf{a}^i_g + \mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)\mathbf{T}^b_c\mathbf{a}^c_{imu,ng} \, . \tag{4.3}$$

4.1.3 Attitude

The attitude quaternion rate of change is given by

$$\dot{\bar{\mathbf{q}}}_i^b = \frac{1}{2}\bar{\boldsymbol{\omega}}_{b/i}^b \otimes \bar{\mathbf{q}}_i^b \,, \tag{4.4}$$

where $\bar{\boldsymbol{\omega}}_{b/i}^b$ is the angular rate quaternion

$$\bar{\boldsymbol{\omega}}_{b/i}^b = \begin{bmatrix} \boldsymbol{\omega}_{b/i}^b \\ 0 \end{bmatrix} \,, \tag{4.5}$$

and $\boldsymbol{\omega}_{b/i}^b$ is the relative angular rate represented in the body reference frame and $\otimes$ represents quaternion multiplication.

4.2 State Dynamics Summary

The three main state differential equations are summarized as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}}_{imu}^i \\ \dot{\mathbf{v}}_{imu}^i \\ \dot{\bar{\mathbf{q}}}_i^b \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{imu}^i \\ \mathbf{a}_g^i + \mathbf{T}_b^i(\bar{\mathbf{q}}_b^i)\mathbf{T}_c^b\mathbf{a}_{imu,ng}^c \\ \frac{1}{2}\bar{\boldsymbol{\omega}}_{b/i}^b \otimes \bar{\mathbf{q}}_i^b \end{bmatrix} \,. \tag{4.6}$$

4.3 Estimation Error Dynamics

Because of the complex nature of attitude quaternion dynamics and their incompatibility with regular mathematical operations, the MEKF algorithm is used. During the update stage of the Kalman filter the measurement residual is used to compute the updated state. However, this operation is a challenge for attitude quaternions due to their inability to be added or subtracted, more specifically, the addition or subtraction of attitude quaternions does not produce an attitude quaternion. Rather than estimate the quaternion directly, we estimate the quaternion error or deviation from the true quaternion, $\delta\boldsymbol{\alpha}$. This is done so that the vector component of the quaternion, which is compatible with mathematical operations such as addition and subtraction, can be included in the models of the other state elements without having to worry about compliance with the attitude quaternion mathematical constraints. Therefore, the differential equation for the quaternion error $\delta\boldsymbol{\alpha}$ must be derived. Since the attitude error dynamics are being derived, the same must be done for position and velocity to ensure compatibility. The position and velocity of the IMU are dependent on the attitude of the spacecraft since the IMU and the center of gravity are not aligned, therefore they cannot be estimated in a separate filter. The attitude, however, can be estimated separately.

In order to compute the **F** matrix of the state estimation error dynamics, the partial derivative of the error state differential equations must be taken with respect to the error states. The estimated state differential equations are

$$\dot{\hat{\mathbf{x}}} = \begin{bmatrix} \dot{\hat{\mathbf{r}}}^i_{imu/i} \\ \dot{\hat{\mathbf{v}}}^i_{imu} \\ \dot{\hat{\mathbf{q}}}^b_i \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{v}}^i_{imu} \\ \hat{\mathbf{a}}^i_g + \hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)\mathbf{T}^b_c\hat{\mathbf{a}}^c_{imu,ng} \\ \frac{1}{2}\hat{\boldsymbol{\omega}}^b_{b/i} \otimes \hat{\mathbf{q}}^b_i \end{bmatrix} . \tag{4.7}$$

The state estimation error dynamics are now derived in the following sections.

4.3.1 Position

The position estimation error is defined as

$$\delta\mathbf{r}^i_{imu} = \mathbf{r}^i_{imu} - \hat{\mathbf{r}}^i_{imu} .$$

Differentiating the position estimation error with respect to time yields the position estimation error differential equation,

$$\delta\dot{\mathbf{r}}^{i}_{imu} = \dot{\mathbf{r}}^{i}_{imu} - \dot{\hat{\mathbf{r}}}^{i}_{imu} \ ,$$

which is the first component of the estimation error differential equation vector, and when substituted with Eqs. 4.6 and 4.7 becomes

$$\delta\dot{\mathbf{r}}^{i}_{imu} = \mathbf{v}^{i}_{imu} - \hat{\mathbf{v}}^{i}_{imu} \ .$$

The final form of the position estimation error differential equation is

$$\delta\dot{\mathbf{r}}^{i}_{imu} = \delta\mathbf{v}^{i}_{imu} \ . \tag{4.8}$$

4.3.2 Velocity

The velocity estimation error is defined as

$$\delta\mathbf{v}^{i}_{imu} = \mathbf{v}^{i}_{imu} - \hat{\mathbf{v}}^{i}_{imu} \ .$$

Differentiating the velocity estimation error with respect to time yields the velocity estimation error differential equation,

$$\delta\dot{\mathbf{v}}^{i}_{imu} = \dot{\mathbf{v}}^{i}_{imu} - \dot{\hat{\mathbf{v}}}^{i}_{imu} \ ,$$

which is the second component of the estimation error dynamics vector, and when substituted with Eqs. 4.6 and 4.7 becomes

$$\delta\dot{\mathbf{v}}^i_{imu} = \mathbf{a}^i_g + \mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)\mathbf{T}^b_c\mathbf{a}^c_{imu,ng} - \hat{\mathbf{a}}^i_g - \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)\mathbf{T}^b_c\hat{\mathbf{a}}^c_{imu,ng}\ . \tag{4.9}$$

Eq. 4.9 represents the differential equation for the velocity estimation error. The gravitational acceleration is given by

$$\mathbf{a}^i_g = -\frac{\mu}{\left\|\mathbf{r}^i_{cg}\right\|^3}\mathbf{r}^i_{cg}\ ,$$

which is a function of the position of the center of gravity of the spacecraft, which is not in the state vector, rather than the position of the IMU, which is in the state vector. We expand the gravitational acceleration in a Taylor series and keeping only the first order terms yields

$$\mathbf{a}^i_g = \hat{\mathbf{a}}^i_g + \left[\left.\frac{\partial\mathbf{a}^i_g}{\partial\mathbf{r}^i_g}\right|_{\mathbf{r}^i_g=\hat{\mathbf{r}}^i_g}\right]\delta\mathbf{r}^i_{cg}\ . \tag{4.10}$$

Define

$$\mathbf{G} := \left[\left.\frac{\partial\mathbf{a}^i_g}{\partial\mathbf{r}^i_g}\right|_{\mathbf{r}^i_{cg}=\hat{\mathbf{r}}^i_{cg}}\right]\ .$$

Eq. 4.10 can be written as

$$\mathbf{a}^i_g = \hat{\mathbf{a}}^i_g + \mathbf{G}\delta\mathbf{r}^i_{cg}\ , \tag{4.11}$$

where the matrix **G** is given as

$$\mathbf{G} = \begin{bmatrix} \frac{\partial a^i_{g,x}}{\partial r^i_{cg,x}} & \frac{\partial a^i_{g,x}}{\partial r^i_{cg,y}} & \frac{\partial a^i_{g,x}}{\partial r^i_{cg,z}} \\ \frac{\partial a^i_{g,y}}{\partial r^i_{cg,x}} & \frac{\partial a^i_{g,y}}{\partial r^i_{cg,y}} & \frac{\partial a^i_{g,y}}{\partial r^i_{cg,z}} \\ \frac{\partial a^i_{g,z}}{\partial r^i_{cg,x}} & \frac{\partial a^i_{g,z}}{\partial r^i_{cg,y}} & \frac{\partial a^i_{g,z}}{\partial r^i_{cg,z}} \end{bmatrix}_{\mathbf{r}^i_{cg}=\hat{\mathbf{r}}^i_{cg}}\ , \tag{4.12}$$

and the matrix partitions are given as

$$
\begin{aligned}
\frac{\partial a^i_{g,x}}{\partial r^i_{cg,x}} &= 3\mu {r^i_{cg,x}}^2 \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{5}{2}} - \mu \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{3}{2}} \\
\frac{\partial a^i_{g,x}}{\partial r^i_{cg,y}} &= 3\mu r^i_{cg,x} r^i_{cg,y} \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{5}{2}} \\
\frac{\partial a^i_{g,x}}{\partial r^i_{cg,z}} &= 3\mu r^i_{cg,x} r^i_{cg,z} \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{5}{2}} \\
\frac{\partial a^i_{g,y}}{\partial r^i_{cg,x}} &= 3\mu r^i_{cg,y} r^i_{cg,x} \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{5}{2}} \\
\frac{\partial a^i_{g,y}}{\partial r^i_{cg,y}} &= 3\mu {r^i_{cg,y}}^2 \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{5}{2}} - \mu \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{3}{2}} \\
\frac{\partial a^i_{g,y}}{\partial r^i_{cg,z}} &= 3\mu r^i_{cg,y} r^i_{cg,z} \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{5}{2}} \\
\frac{\partial a^i_{g,z}}{\partial r^i_{cg,x}} &= 3\mu r^i_{cg,z} r^i_{cg,x} \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{5}{2}} \\
\frac{\partial a^i_{g,z}}{\partial r^i_{cg,y}} &= 3\mu r^i_{cg,z} r^i_{cg,y} \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{5}{2}} \\
\frac{\partial a^i_{g,z}}{\partial r^i_{cg,z}} &= 3\mu {r^i_{cg,z}}^2 \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{5}{2}} - \mu \left[{r^i_{cg,x}}^2 + {r^i_{cg,y}}^2 + {r^i_{cg,z}}^2\right]^{-\frac{3}{2}} .
\end{aligned}
$$

We have now introduced $\delta \mathbf{r}^i_{cg}$. This quantity is not in the state vector, therefore it must be written in terms of known or estimated state variables. From Figure 4.1, $\mathbf{r}^i_{cg}$ and $\mathbf{r}^i_{imu}$ are related through

$$\mathbf{r}^i_{cg} = \mathbf{r}^i_{imu} + \mathbf{r}^i_{cg/imu} .$$

However, $\mathbf{r}_{cg/imu}$ is known in the body reference frame, therefore,

$$\mathbf{r}^i_{cg} = \mathbf{r}^i_{imu} + \mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)\mathbf{r}^b_{cg/imu} ,$$

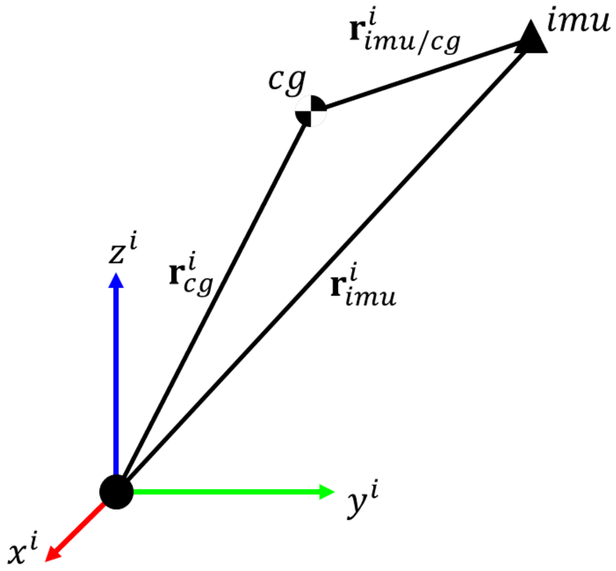


Figure 4.1: Center of Gravity and the IMU in the Inertial Reference Frame

and the estimated position of the center of gravity given by

$$\hat{\mathbf{r}}^i_{cg} = \hat{\mathbf{r}}^i_{imu} + \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)\mathbf{r}^b_{cg/imu} \, .$$

The center of gravity is assumed to be constant and not shift with respect to the IMU. For purposes of our smallsat (with no propulsion system), $\mathbf{r}^b_{cg/imu}$ is known and constant. The estimation error for the position of the center of gravity then becomes

$$\begin{aligned} \delta\mathbf{r}^i_{cg} &= \mathbf{r}^i_{imu} + \mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)\mathbf{r}^b_{cg/imu} - \hat{\mathbf{r}}^i_{imu} - \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)\mathbf{r}^b_{cg/imu} \\ &= \delta\mathbf{r}^i_{imu} + [\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b) - \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)]\mathbf{r}^b_{cg/imu} \, . \end{aligned} \tag{4.13}$$

Recall from Eq. 2.27 that a small angle transformation matrix $\delta\mathbf{T}^b_i$ can be expressed in terms of $\delta\boldsymbol{\alpha}$ as

$$\delta\mathbf{T}^b_i = \mathbf{I} - [\delta\boldsymbol{\alpha}\times] \, .$$

The small angle transformation $\delta\mathbf{T}_i^b$ represents the small angle deviation between the true and estimated orientation,

$$\delta\mathbf{T}_i^b = \mathbf{T}_i^b(\bar{\mathbf{q}}_i^b)[\hat{\mathbf{T}}_i^b(\hat{\mathbf{q}}_i^b)]^{-1} .$$

Then, we have

$$\mathbf{T}_i^b(\bar{\mathbf{q}}_i^b)[\hat{\mathbf{T}}_i^b(\hat{\mathbf{q}}_i^b)]^{-1} = \mathbf{I} - [\delta\boldsymbol{\alpha}\times] ,$$

from which it follows that

$$\mathbf{T}_i^b(\bar{\mathbf{q}}_i^b) = \hat{\mathbf{T}}_i^b(\hat{\mathbf{q}}_i^b) - [\delta\boldsymbol{\alpha}\times]\hat{\mathbf{T}}_i^b(\hat{\mathbf{q}}_i^b) . \tag{4.14}$$

and

$$\mathbf{T}_b^i(\bar{\mathbf{q}}_b^i) = \hat{\mathbf{T}}_b^i(\hat{\mathbf{q}}_b^i) + \hat{\mathbf{T}}_b^i(\hat{\mathbf{q}}_b^i)[\delta\boldsymbol{\alpha}\times] . \tag{4.15}$$

Substituting Eq. 4.15 into Eq. 4.13 yields

$$\delta\mathbf{r}_{cg}^i = \delta\mathbf{r}_{imu}^i + \hat{\mathbf{T}}_b^i(\hat{\mathbf{q}}_b^i)[\delta\boldsymbol{\alpha}\times]\mathbf{r}_{cg/imu}^b . \tag{4.16}$$

Substituting Eq. 4.16 into Eq. 4.11 yields

$$\mathbf{a}_g^i = \hat{\mathbf{a}}_g^i + \mathbf{G}\left[\delta\mathbf{r}_{imu}^i + \hat{\mathbf{T}}_b^i(\hat{\mathbf{q}}_b^i)[\delta\boldsymbol{\alpha}\times]\mathbf{r}_{cg/imu}^b\right] . \tag{4.17}$$

Eq. 4.17 can then be substituted into Eq. 4.9, yielding

$$\delta\dot{\mathbf{v}}_{imu}^i = \mathbf{G}\left[\delta\mathbf{r}_{imu}^i + \hat{\mathbf{T}}_b^i(\hat{\mathbf{q}}_b^i)[\delta\boldsymbol{\alpha}\times]\mathbf{r}_{cg/imu}^b\right] + \mathbf{T}_b^i(\bar{\mathbf{q}}_b^i)\mathbf{T}_c^b\mathbf{a}_{imu,ng}^c - \hat{\mathbf{T}}_b^i(\hat{\mathbf{q}}_b^i)\mathbf{T}_c^b\hat{\mathbf{a}}_{imu,ng}^c . \tag{4.18}$$

Substituting $\mathbf{T}_b^i(\bar{\mathbf{q}}_b^i)$ from Eq. 4.15 in Eq. 4.18 yields

$$\begin{aligned}\delta\dot{\mathbf{v}}_{imu}^i &= \mathbf{G}\left[\delta\mathbf{r}_{imu}^i + \hat{\mathbf{T}}_b^i(\hat{\mathbf{q}}_b^i)[\delta\boldsymbol{\alpha}\times]\mathbf{r}_{cg/imu}^b\right] \\ &\quad + \hat{\mathbf{T}}_b^i(\hat{\mathbf{q}}_b^i)\mathbf{T}_c^b\mathbf{a}_{imu,ng}^c + \hat{\mathbf{T}}_b^i(\hat{\mathbf{q}}_b^i)[\delta\boldsymbol{\alpha}\times]\mathbf{T}_c^b\mathbf{a}_{imu,ng}^c - \hat{\mathbf{T}}_b^i(\hat{\mathbf{q}}_b^i)\mathbf{T}_c^b\hat{\mathbf{a}}_{imu,ng}^c . \end{aligned} \tag{4.19}$$

Substituting the non-gravitational acceleration error,

$$\delta \mathbf{a}^c_{imu,ng} = \mathbf{a}^c_{imu,ng} - \hat{\mathbf{a}}^c_{imu,ng} \, ,$$

into Eq. 4.19 yields

$$\begin{aligned} \delta \dot{\mathbf{v}}^i_{imu} = \mathbf{G}\left[\delta \mathbf{r}^i_{imu} + \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\delta\boldsymbol{\alpha}\times]\mathbf{r}^b_{cg/imu}\right] \\ + \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)\mathbf{T}^b_c\delta\mathbf{a}^c_{imu,ng} + \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\delta\boldsymbol{\alpha}\times]\mathbf{T}^b_c\hat{\mathbf{a}}^c_{imu,ng} + \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\delta\boldsymbol{\alpha}\times]\mathbf{T}^b_c\delta\mathbf{a}^c_{imu,ng} \, . \end{aligned} \tag{4.20}$$

Neglecting second-order and higher-order terms, Eq. 4.20 reduces to

$$\delta \dot{\mathbf{v}}^i_{imu} = \mathbf{G}\left[\delta \mathbf{r}^i_{imu} + \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\delta\boldsymbol{\alpha}\times]\mathbf{r}^b_{cg/imu}\right] + \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)\mathbf{T}^b_c\delta\mathbf{a}^c_{imu,ng} + \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\delta\boldsymbol{\alpha}\times]\mathbf{T}^b_c\hat{\mathbf{a}}^c_{imu,ng} \, . \tag{4.21}$$

Equation 4.21 represents the velocity estimation error differential equation. In order to compute the Jacobian of the dynamics, the estimated states must be isolated. It can be seen that $\delta\boldsymbol{\alpha}$ is not isolated, rather, it is inside the cross product matrix. In order to bring it out and isolate it so that the Jacobian can be computed, the property of Eq. 2.9 is utilized to yield the final form of the velocity estimation error differential equation,

$$\delta \dot{\mathbf{v}}^i_{imu} = \mathbf{G}\left[\delta \mathbf{r}^i_{imu} - \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{cg/imu}\times]\delta\boldsymbol{\alpha}\right] + \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)\mathbf{T}^b_c\delta\mathbf{a}^c_{imu,ng} - \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)\mathbf{T}^b_c[\hat{\mathbf{a}}^c_{imu,ng}\times]\delta\boldsymbol{\alpha} \, . \tag{4.22}$$

We consider $\delta\mathbf{a}^c_{imu,ng}$ and $\hat{\mathbf{a}}^c_{imu,ng}$ more closely in Chapter 5 when the IMU accelerometer model is derived.

4.3.3 Attitude

The quaternion estimation error is defined as

$$\delta\bar{\mathbf{q}}^b_i = \bar{\mathbf{q}}^b_i \otimes [\hat{\bar{\mathbf{q}}}^b_i]^{-1} \, .$$

Differentiating the quaternion estimation error with respect to time yields the quaternion estimation error differential equation,

$$\delta\dot{\bar{\mathbf{q}}}_i^b = \dot{\bar{\mathbf{q}}}_i^b \otimes [\hat{\bar{\mathbf{q}}}_i^b]^{-1} + \bar{\mathbf{q}}_i^b \otimes [\dot{\hat{\bar{\mathbf{q}}}}_i^b]^{-1} \,.$$

Substituting in the third row from Eqs. 4.6 and 4.7 yields

$$\begin{aligned} \delta\dot{\bar{\mathbf{q}}}_i^b &= [\frac{1}{2}\bar{\boldsymbol{\omega}}_{b/i}^b \otimes \bar{\mathbf{q}}_i^b] \otimes [\hat{\bar{\mathbf{q}}}_i^b]^{-1} + \bar{\mathbf{q}}_i^b \otimes [\frac{1}{2}\hat{\bar{\boldsymbol{\omega}}}_{b/i}^b \otimes \hat{\bar{\mathbf{q}}}_i^b]^{-1} \\ &= \frac{1}{2}\bar{\boldsymbol{\omega}}_{b/i}^b \otimes \bar{\mathbf{q}}_i^b \otimes [\hat{\bar{\mathbf{q}}}_i^b]^{-1} + \bar{\mathbf{q}}_i^b \otimes [\hat{\bar{\mathbf{q}}}_i^b]^{-1} \otimes \frac{1}{2}[\hat{\bar{\boldsymbol{\omega}}}_{b/i}^b]^{-1} \\ &= \frac{1}{2}\bar{\boldsymbol{\omega}}_{b/i}^b \otimes \delta\bar{\mathbf{q}}_i^b + \delta\bar{\mathbf{q}}_i^b \otimes \frac{1}{2}[\hat{\bar{\boldsymbol{\omega}}}_{b/i}^b]^{-1} \\ &= \frac{1}{2}\begin{bmatrix} \boldsymbol{\omega}_{b/i}^b \\ 0 \end{bmatrix} \otimes \delta\bar{\mathbf{q}}_i^b + \delta\bar{\mathbf{q}}_i^b \otimes \frac{1}{2}\begin{bmatrix} -\hat{\boldsymbol{\omega}}_{b/i}^b \\ 0 \end{bmatrix} \,. \end{aligned} \tag{4.23}$$

Expanding Eq. 4.23 using the quaternion multiplication definition from Eq. 2.19 yields

$$\delta\dot{\bar{\mathbf{q}}}_i^b = \begin{bmatrix} \frac{1}{2}\boldsymbol{\omega}_{b/i}^b \delta {q_0}_i^b - \frac{1}{2}\boldsymbol{\omega}_{b/i}^b \times \delta\mathbf{q}_i^b - \frac{1}{2}\hat{\boldsymbol{\omega}}_{b/i}^b \delta {q_0}_i^b + \delta\mathbf{q}_i^b \times \frac{1}{2}\hat{\boldsymbol{\omega}}_{b/i}^b \\ \frac{1}{2}\hat{\boldsymbol{\omega}}_{b/i}^b \cdot \delta\mathbf{q}_i^b - \frac{1}{2}\boldsymbol{\omega}_{b/i}^b \cdot \delta\mathbf{q}_i^b \end{bmatrix} \,. \tag{4.24}$$

Using Eq. 2.8 to group cross product terms into matrices yields

$$\delta\dot{\bar{\mathbf{q}}}_i^b = \begin{bmatrix} \frac{1}{2}\boldsymbol{\omega}_{b/i}^b \delta {q_0}_i^b - \frac{1}{2}[\boldsymbol{\omega}_{b/i}^b \times]\delta\mathbf{q}_i^b - \frac{1}{2}\hat{\boldsymbol{\omega}}_{b/i}^b \delta {q_0}_i^b + [\delta\mathbf{q}_i^b \times]\frac{1}{2}\hat{\boldsymbol{\omega}}_{b/i}^b \\ \frac{1}{2}\hat{\boldsymbol{\omega}}_{b/i}^b \cdot \delta\mathbf{q}_i^b - \frac{1}{2}\boldsymbol{\omega}_{b/i}^b \cdot \delta\mathbf{q}_i^b \end{bmatrix} \,. \tag{4.25}$$

Recalling Eq. 2.9, Eq. 4.25 reduces to

$$\delta\dot{\bar{\mathbf{q}}}_i^b = \begin{bmatrix} \frac{1}{2}\boldsymbol{\omega}_{b/i}^b \delta {q_0}_i^b - \frac{1}{2}[\boldsymbol{\omega}_{b/i}^b \times]\delta\mathbf{q}_i^b - \frac{1}{2}\hat{\boldsymbol{\omega}}_{b/i}^b \delta {q_0}_i^b - \frac{1}{2}[\hat{\boldsymbol{\omega}}_{b/i}^b \times]\delta\mathbf{q}_i^b \\ \frac{1}{2}\hat{\boldsymbol{\omega}}_{b/i}^b \cdot \delta\mathbf{q}_i^b - \frac{1}{2}\boldsymbol{\omega}_{b/i}^b \cdot \delta\mathbf{q}_i^b \end{bmatrix} \,. \tag{4.26}$$

Applying the definition of the angular velocity estimation error $\delta\boldsymbol{\omega}_{b/i}^{i} = \boldsymbol{\omega}_{b/i}^{i} - \hat{\boldsymbol{\omega}}_{b/i}^{i}$ to Eq. 4.26 yields

$$\delta\dot{\mathbf{q}}_i^b = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\omega}_{b/i}^{b}\delta {q_0}_i^b - \frac{1}{2}[\boldsymbol{\omega}_{b/i}^{b}\times]\delta\mathbf{q}_i^b - \frac{1}{2}[\hat{\boldsymbol{\omega}}_{b/i}^{b}\times]\delta\mathbf{q}_i^b \\ -\frac{1}{2}\delta\boldsymbol{\omega}_{b/i}^{b}\cdot\delta\mathbf{q}_i^b \end{bmatrix} . \tag{4.27}$$

Now applying the definition of the angular velocity estimation error re-arranged for the true angular velocity, $\boldsymbol{\omega}_{b/i}^{i} = \delta\boldsymbol{\omega}_{b/i}^{i} + \hat{\boldsymbol{\omega}}_{b/i}^{i}$, to Eq. 4.27 yields

$$\begin{aligned} \delta\dot{\mathbf{q}}_i^b &= \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\omega}_{b/i}^{b}\delta {q_0}_i^b - \frac{1}{2}[\delta\boldsymbol{\omega}_{b/i}^{b}\times]\delta\mathbf{q}_i^b - \frac{1}{2}[\hat{\boldsymbol{\omega}}_{b/i}^{b}\times]\delta\mathbf{q}_i^b - \frac{1}{2}[\hat{\boldsymbol{\omega}}_{b/i}^{b}\times]\delta\mathbf{q}_i^b \\ -\frac{1}{2}\delta\boldsymbol{\omega}_{b/i}^{b}\cdot\delta\mathbf{q}_i^b \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\omega}_{b/i}^{b}\delta {q_0}_i^b - \frac{1}{2}[\delta\boldsymbol{\omega}_{b/i}^{b}\times]\delta\mathbf{q}_i^b - [\hat{\boldsymbol{\omega}}_{b/i}^{b}\times]\delta\mathbf{q}_i^b \\ -\frac{1}{2}\delta\boldsymbol{\omega}_{b/i}^{b}\cdot\delta\mathbf{q}_i^b \end{bmatrix} . \end{aligned} \tag{4.28}$$

Eq. 4.28 can then be reduced in complexity based on the following assumptions:

- A small angle assumption implies that $\delta {q_0}_i^b \rightarrow 1$ since the scalar part of any small angle quaternion is near 1.
- Under first order assumptions, all second and higher order terms may be neglected, i.e., $\delta\boldsymbol{\omega}_{b/i}^{b} \times \delta\mathbf{q}_i^b \rightarrow \mathbf{0}$ and $-\delta\boldsymbol{\omega}_{b/i}^{b}\cdot\delta {q_0}_i^b \rightarrow \mathbf{0}$.

Applying these simplifications yields the final form of the quaternion estimation error differential equation,

$$\delta\dot{\mathbf{q}}_i^b = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\omega}_{b/i}^{b} - [\hat{\boldsymbol{\omega}}_{b/i}^{b}\times]\delta\mathbf{q}_i^b \\ 0 \end{bmatrix} . \tag{4.29}$$

However, since the angle deviation (the vector component of the small angle quaternion) is the quantity being estimated, one more expansion is needed. Recalling the small angle quaternion approximation from Eq. 2.22

$$\delta\bar{\mathbf{q}}_i^b \approx \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\alpha} \\ 1 \end{bmatrix} ,$$

it can be seen from Eq. 4.29 that the rate of change with respect to time of the small angle quaternion is

$$\delta\dot{\bar{\mathbf{q}}}_i^b \approx \begin{bmatrix} \frac{1}{2}\delta\dot{\boldsymbol{\alpha}} \\ 0 \end{bmatrix} . \tag{4.30}$$

Applying the approximation in Eq. 4.30 to Eq. 4.29 yields

$$\begin{bmatrix} \frac{1}{2}\delta\dot{\boldsymbol{\alpha}} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\omega}_{b/i}^b - [\hat{\boldsymbol{\omega}}_{b/i}^b\times]\frac{1}{2}\delta\boldsymbol{\alpha} \\ 0 \end{bmatrix} ,$$

where the final form of the angle deviation estimation error differential equation is

$$\delta\dot{\boldsymbol{\alpha}} = \delta\boldsymbol{\omega}_{b/i}^b - [\hat{\boldsymbol{\omega}}_{b/i}^b\times]\delta\boldsymbol{\alpha} . \tag{4.31}$$

We consider $\delta\boldsymbol{\omega}_{b/i}^b$ and $\hat{\boldsymbol{\omega}}_{b/i}^b$ more closely in Chapter 5 when the model for the gyroscope is derived.

Chapter 5: IMU and Sensor Models

Models for both the onboard sensors and IMU are presented. These models will include the IMU systematic and random errors, as well as sensor noise and bias.

5.1 IMU - Accelerometer

The IMU measures the non-gravitational acceleration of the spacecraft. We model the measurement as being corrupted with random noise, bias, axes non-orthogonality, axes misalignment, and scale factor errors. The measured non-gravitational acceleration is represented in terms of these errors as

$$\mathbf{a}^{c}_{imu,ng,m} = (\mathbf{I} + \mathbf{M}_a + \mathbf{N}_a)(\mathbf{I} + \mathbf{S}_a)(\mathbf{a}^{c}_{imu,ng} + \boldsymbol{\beta}_a + \boldsymbol{\eta}_a)\ , \tag{5.1}$$

where $\boldsymbol{\beta}_a$ is the accelerometer bias, $\boldsymbol{\eta}_a$ is the accelerometer noise, $\mathbf{M}_a$ is the accelerometer axes misalignment error matrix, $\mathbf{N}_a$ is the accelerometer axes non-orthogonality error matrix, and $\mathbf{S}_a$ is the accelerometer scale factor uncertainty matrix. The systematic error matrices are

$$\mathbf{M}_a = \begin{bmatrix} 0 & m_{a,z} & -m_{a,y} \\ -m_{a,z} & 0 & m_{a,x} \\ m_{a,y} & -m_{a,x} & 0 \end{bmatrix} = -[\mathbf{m}_a \times] \quad \text{with} \quad \mathbf{m}_a = \begin{bmatrix} m_{a,x} \\ m_{a,y} \\ m_{a,z} \end{bmatrix}\ ,$$

$$\mathbf{N}_a = \begin{bmatrix} 0 & n_{a,z} & n_{a,y} \\ n_{a,z} & 0 & n_{a,x} \\ n_{a,y} & n_{a,x} & 0 \end{bmatrix} = [\mathbf{n}_a | \times |] \quad \text{with} \quad \mathbf{n}_a = \begin{bmatrix} n_{a,x} \\ n_{a,y} \\ n_{a,z} \end{bmatrix} ,$$

$$\text{and} \quad \mathbf{S}_a = \begin{bmatrix} s_{a,x} & 0 & 0 \\ 0 & s_{a,y} & 0 \\ 0 & 0 & s_{a,z} \end{bmatrix} = [\mathbf{s}_a \diagdown] \quad \text{with} \quad \mathbf{s}_a = \begin{bmatrix} s_{a,x} \\ s_{a,y} \\ s_{a,z} \end{bmatrix} .$$

We expand

$$(\mathbf{I} + \mathbf{M}_a + \mathbf{N}_a)(\mathbf{I} + \mathbf{S}_a) = \mathbf{I} + \mathbf{M}_a + \mathbf{N}_a + \mathbf{S}_a + \mathbf{M}_a\mathbf{S}_a + \mathbf{N}_a\mathbf{S}_a . \tag{5.2}$$

Since $\mathbf{M}_a$, $\mathbf{N}_a$, and $\mathbf{S}_a$ are considered to be small error terms, $\mathbf{M}_a\mathbf{S}_a$ and $\mathbf{N}_a\mathbf{S}_a$ can be neglected under first order assumptions. Here, it follows that

$$(\mathbf{I} + \mathbf{M}_a + \mathbf{N}_a)(\mathbf{I} + \mathbf{S}_a) \approx \mathbf{I} + \mathbf{M}_a + \mathbf{N}_a + \mathbf{S}_a .$$

Define

$$\boldsymbol{\Delta}_a := \mathbf{M}_a + \mathbf{N}_a + \mathbf{S}_a .$$

We can re-write Eq. 5.2 as

$$(\mathbf{I} + \mathbf{M}_a + \mathbf{N}_a)(\mathbf{I} + \mathbf{S}_a) \approx \mathbf{I} + \boldsymbol{\Delta}_a ,$$

and Eq. 5.1 reduces to

$$\mathbf{a}^c_{imu,ng,m} = (\mathbf{I} + \boldsymbol{\Delta}_a)(\mathbf{a}^c_{imu,ng} + \boldsymbol{\beta}_a + \boldsymbol{\eta}_a) . \tag{5.3}$$

Solving for $\mathbf{a}^c_{imu,ng}$ yields

$$\begin{aligned}
\mathbf{a}^c_{imu,ng} &= (\mathbf{I}+\mathbf{\Delta}_a)^{-1}\mathbf{a}^c_{imu,ng,m} - \boldsymbol{\beta}_a - \boldsymbol{\eta}_a \\
&= (\mathbf{I}-\mathbf{\Delta}_a)\mathbf{a}^c_{imu,ng,m} - \boldsymbol{\beta}_a - \boldsymbol{\eta}_a \\
&= (\mathbf{I}-\mathbf{M}_a-\mathbf{N}_a-\mathbf{S}_a)\mathbf{a}^c_{imu,ng,m} - \boldsymbol{\beta}_a - \boldsymbol{\eta}_a \\
&= \mathbf{a}^c_{imu,ng,m} - \mathbf{M}_a\mathbf{a}^c_{imu,ng,m} - \mathbf{N}_a\mathbf{a}^c_{imu,ng,m} - \mathbf{S}_a\mathbf{a}^c_{imu,ng,m} - \boldsymbol{\beta}_a - \boldsymbol{\eta}_a \\
&= \mathbf{a}^c_{imu,ng,m} + [\mathbf{m}_a\times]\mathbf{a}^c_{imu,ng,m} - [\mathbf{n}_a|\times||]\mathbf{a}^c_{imu,ng,m} - [\mathbf{s}_a\diagdown]\mathbf{a}^c_{imu,ng,m} - \boldsymbol{\beta}_a - \boldsymbol{\eta}_a\,, \qquad (5.4)
\end{aligned}$$

where we utilize the fact that Δ_a represents small angle errors and

$$(\mathbf{I}+\mathbf{\Delta}_a)^{-1} \approx \mathbf{I}-\mathbf{\Delta}_a\,.$$

Utilizing the definitions in Eqs. 2.9, 2.10, and 2.11 allows us to re-write the final form the true non-gravitational acceleration as

$$\mathbf{a}^c_{imu,ng} = \mathbf{a}^c_{imu,ng,m} - [\mathbf{a}^c_{imu,ng,m}\times]\mathbf{m}_a - [\mathbf{a}^c_{imu,ng,m}|\times||]\mathbf{n}_a - [\mathbf{a}^c_{imu,ng,m}\diagdown]\mathbf{s}_a - \boldsymbol{\beta}_a - \boldsymbol{\eta}_a\,. \qquad (5.5)$$

The estimated non-gravitational acceleration is

$$\hat{\mathbf{a}}^c_{imu,ng} = \mathbf{a}^c_{imu,ng,m} - [\mathbf{a}^c_{imu,ng,m}\times]\hat{\mathbf{m}}_a - [\mathbf{a}^c_{imu,ng,m}|\times||]\hat{\mathbf{n}}_a - [\mathbf{a}^c_{imu,ng,m}\diagdown]\hat{\mathbf{s}}_a - \hat{\boldsymbol{\beta}}_a\,. \qquad (5.6)$$

The non-gravitational acceleration estimation error, $\delta\mathbf{a}^c_{imu,ng} = \mathbf{a}^c_{imu,ng} - \hat{\mathbf{a}}^c_{imu,ng}$, is

$$\delta\mathbf{a}^c_{imu,ng} = -[\mathbf{a}^c_{imu,ng,m}\times]\delta\mathbf{m}_a - [\mathbf{a}^c_{imu,ng,m}|\times||]\delta\mathbf{n}_a - [\mathbf{a}^c_{imu,ng,m}\diagdown]\delta\mathbf{s}_a - \delta\boldsymbol{\beta}_a - \boldsymbol{\eta}_a\,. \qquad (5.7)$$

5.2 IMU - Gyroscope

The same procedure is followed for the angular velocity since the gyroscope is corrupted with similar random and systematic errors as the accelerometer. We model the measured angular rate as

$$\boldsymbol{\omega}^c_{b/i,m} = (\mathbf{I} + \mathbf{M}_g + \mathbf{N}_g)(\mathbf{I} + \mathbf{S}_g)(\boldsymbol{\omega}^c_{b/i} + \boldsymbol{\beta}_g + \boldsymbol{\eta}_g)\,, \tag{5.8}$$

where $\boldsymbol{\beta}_g$ is the gyroscope bias, $\boldsymbol{\eta}_g$ is the gyroscope noise, $\mathbf{M}_g$ is the gyroscope axes misalignment error matrix, $\mathbf{N}_g$ is the gyroscope axes non-orthogonality error matrix, and $\mathbf{S}_g$ is the gyroscope scale factor uncertainty matrix. The systematic error matrices are

$$\mathbf{M}_g = \begin{bmatrix} 0 & m_{g,z} & -m_{g,y} \\ -m_{g,z} & 0 & m_{g,x} \\ m_{g,y} & -m_{g,x} & 0 \end{bmatrix} = -[\mathbf{m}_g \times] \quad \text{with} \quad \mathbf{m}_g = \begin{bmatrix} m_{g,x} \\ m_{g,y} \\ m_{g,z} \end{bmatrix},$$

$$\mathbf{N}_g = \begin{bmatrix} 0 & n_{g,z} & n_{g,y} \\ n_{g,z} & 0 & n_{g,x} \\ n_{g,y} & n_{g,x} & 0 \end{bmatrix} = [\mathbf{n}_g | \times |] \quad \text{with} \quad \mathbf{n}_g = \begin{bmatrix} n_{g,x} \\ n_{g,y} \\ n_{g,z} \end{bmatrix},$$

$$\text{and} \quad \mathbf{S}_g = \begin{bmatrix} s_{g,x} & 0 & 0 \\ 0 & s_{g,y} & 0 \\ 0 & 0 & s_{g,z} \end{bmatrix} = [\mathbf{s}_g \diagdown] \quad \text{with} \quad \mathbf{s}_g = \begin{bmatrix} s_{g,x} \\ s_{g,y} \\ s_{g,z} \end{bmatrix}.$$

As before, we expand

$$(\mathbf{I} + \mathbf{M}_g + \mathbf{N}_g)(\mathbf{I} + \mathbf{S}_g) = \mathbf{I} + \mathbf{M}_g + \mathbf{N}_g + \mathbf{S}_g + \mathbf{M}_g\mathbf{S}_g + \mathbf{N}_g\mathbf{S}_g\,.$$

Since $\mathbf{M}_g$, $\mathbf{N}_g$, and $\mathbf{S}_g$ are considered to be small error terms, $\mathbf{M}_g\mathbf{S}_g$ and $\mathbf{N}_g\mathbf{S}_g$ can be neglected under first order assumptions. Here, it follows that

$$(\mathbf{I} + \mathbf{M}_g + \mathbf{N}_g)(\mathbf{I} + \mathbf{S}_g) \approx \mathbf{I} + \mathbf{M}_g + \mathbf{N}_g + \mathbf{S}_g\,. \tag{5.9}$$

Define

$$\boldsymbol{\Delta}_g := \mathbf{M}_g + \mathbf{N}_g + \mathbf{S}_g \; .$$

Then, we can write Eq. 5.9 as

$$(\mathbf{I} + \mathbf{M}_g + \mathbf{N}_g)(\mathbf{I} + \mathbf{S}_g) \approx \mathbf{I} + \boldsymbol{\Delta}_g \; ,$$

and Eq. 5.8 reduces to

$$\boldsymbol{\omega}^c_{b/i,m} = (\mathbf{I} + \boldsymbol{\Delta}_g)(\boldsymbol{\omega}^c_{b/i} + \boldsymbol{\beta}_a + \boldsymbol{\eta}_g) \; . \tag{5.10}$$

Solving for $\boldsymbol{\omega}^c_{b/i}$ yields

$$\begin{aligned}
\boldsymbol{\omega}^c_{b/i} &= (\mathbf{I} + \boldsymbol{\Delta}_g)^{-1}\boldsymbol{\omega}^c_{b/i,m} - \boldsymbol{\beta}_g - \boldsymbol{\eta}_g \\
&= (\mathbf{I} - \boldsymbol{\Delta}_g)\boldsymbol{\omega}^c_{b/i,m} - \boldsymbol{\beta}_g - \boldsymbol{\eta}_g \\
&= (\mathbf{I} - \mathbf{M}_g - \mathbf{N}_g - \mathbf{S}_g)\boldsymbol{\omega}^c_{b/i,m} - \boldsymbol{\beta}_g - \boldsymbol{\eta}_g \\
&= \boldsymbol{\omega}^c_{b/i,m} - \mathbf{M}_g\boldsymbol{\omega}^c_{b/i,m} - \mathbf{N}_g\boldsymbol{\omega}^c_{b/i,m} - \mathbf{S}_g\boldsymbol{\omega}^c_{b/i,m} - \boldsymbol{\beta}_g - \boldsymbol{\eta}_g \\
&= \boldsymbol{\omega}^c_{b/i,m} + [\mathbf{m}_g \times]\boldsymbol{\omega}^c_{b/i,m} - [\mathbf{n}_g|\times|]\boldsymbol{\omega}^c_{b/i,m} - [\mathbf{s}_g \diagdown]\boldsymbol{\omega}^c_{b/i,m} - \boldsymbol{\beta}_g - \boldsymbol{\eta}_g \; ,
\end{aligned}$$

where we utilize the fact that Δ_g represents small angle errors and

$$(\mathbf{I} + \boldsymbol{\Delta}_g)^{-1} \approx \mathbf{I} - \boldsymbol{\Delta}_g \; .$$

Utilizing the definitions in Eqs. 2.9, 2.10, and 2.11 allows us to write the final form the true angular velocity as

$$\boldsymbol{\omega}^c_{b/i} = \boldsymbol{\omega}^c_{b/i,m} - [\boldsymbol{\omega}^c_{b/i,m} \times]\mathbf{m}_g - [\boldsymbol{\omega}^c_{b/i,m}|\times|]\mathbf{n}_g - [\boldsymbol{\omega}^c_{b/i,m} \diagdown]\mathbf{s}_g - \boldsymbol{\beta}_g - \boldsymbol{\eta}_g \; . \tag{5.11}$$

The estimated angular velocity is

$$\hat{\boldsymbol{\omega}}^c_{b/i} = \boldsymbol{\omega}^c_{b/i,m} - [\boldsymbol{\omega}^c_{b/i,m}\times]\hat{\mathbf{m}}_g - [\boldsymbol{\omega}^c_{b/i,m}|\times|]\hat{\mathbf{n}}_g - [\boldsymbol{\omega}^c_{b/i,m}\diagdown]\hat{\mathbf{s}}_g - \hat{\boldsymbol{\beta}}_g \, . \tag{5.12}$$

The angular velocity estimation error, $\delta\boldsymbol{\omega}^c_{b/i} = \boldsymbol{\omega}^c_{c/i} - \hat{\boldsymbol{\omega}}^c_{b/i}$, is

$$\delta\boldsymbol{\omega}^c_{b/i} = -[\boldsymbol{\omega}^c_{b/i,m}\times]\delta\mathbf{m}_g - [\boldsymbol{\omega}^c_{b/i,m}|\times|]\delta\mathbf{n}_g - [\boldsymbol{\omega}^c_{b/i,m}\diagdown]\delta\mathbf{s}_g - \delta\boldsymbol{\beta}_g - \boldsymbol{\eta}_g \, . \tag{5.13}$$

5.3 GPS Position

In this section, the sensor model for the GPS position measurement is presented. The GPS reports position in the ECEF reference frame. The state to measurement mapping equation yields the GPS position measurement in terms of state variables, therefore a model for how the GPS position is related to the IMU position is needed. This model includes the error terms that are associated with the GPS measurement. In the case of the GPS, the error terms are a fixed random bias and random noise. Figure 5.1 illustrates the relationship between the IMU and GPS positions.

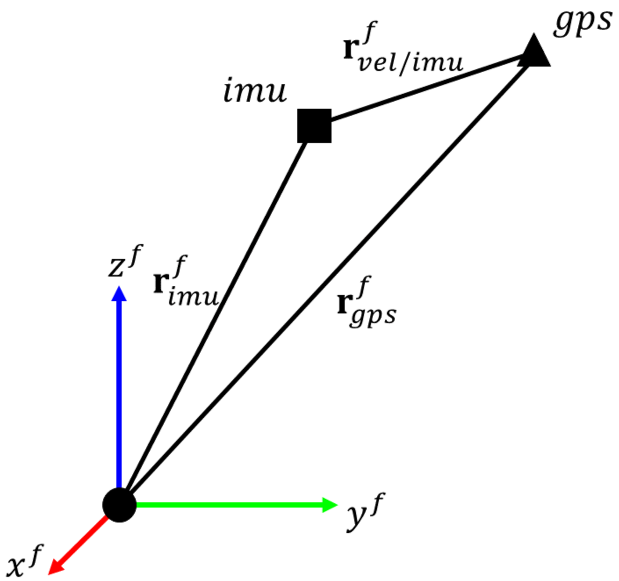


Figure 5.1: IMU and GPS in the ECEF Reference Frame

The position of the GPS with respect to the IMU in the fixed reference frame is given by

$$\mathbf{r}^f_{gps} = \mathbf{r}^f_{imu} + \mathbf{r}^f_{gps/imu} \, .$$

Since the position measurement is corrupted with a fixed random bias and random noise, the equation that represents the GPS position measurement in terms of the IMU position and GPS error terms is

$$\mathbf{r}^f_{gps} = [\mathbf{r}^f_{imu} + \mathbf{r}^f_{gps/imu}] + \boldsymbol{\beta}^f_{pos} + \boldsymbol{\eta}^f_{pos} \, .$$

The vector $\mathbf{r}_{gps/imu}$ is more naturally represented in the body reference frame, not in the ECEF reference frame. Using the body to fixed transformation,

$$\mathbf{r}^f_{gps} = [\mathbf{r}^f_{imu} + \mathbf{T}^f_b \mathbf{r}^b_{gps/imu}] + \boldsymbol{\beta}^f_{pos} + \boldsymbol{\eta}^f_{pos} \, .$$

In terms of the inertial to body reference frame transformation, the body to fixed reference frame transformation is

$$\mathbf{T}^f_b = \mathbf{T}^f_i(t)\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b) \, , \tag{5.14}$$

where $\mathbf{T}^f_i(t)$ is a known function of time. Therefore,

$$\mathbf{r}^f_{gps} = [\mathbf{r}^f_{imu} + \mathbf{T}^f_i(t)\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)\mathbf{r}^b_{gps/imu}] \, .$$

The position $\mathbf{r}_{imu}$ is estimated in the inertial reference frame, not the ECEF reference frame. Using the inertial to fixed reference frame transformation, the true GPS position measurement is

$$\mathbf{r}^f_{gps} = [\mathbf{T}^f_i(t)\mathbf{r}^i_{imu} + \mathbf{T}^f_i(t)\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)\mathbf{r}^b_{gps/imu}] + \boldsymbol{\beta}^f_{pos} + \boldsymbol{\eta}^f_{pos} \, .$$

The estimated GPS position measurement is

$$\hat{\mathbf{r}}^f_{gps} = [\mathbf{T}^f_i(t)\hat{\mathbf{r}}^i_{imu} + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)\mathbf{r}^b_{gps/imu}] + \hat{\boldsymbol{\beta}}^f_{pos}\ .$$

With the estimation error of the GPS position measurement defined as $\delta\mathbf{r}^f_{gps} = \mathbf{r}^f_{gps} - \hat{\mathbf{r}}^f_{gps}$, we have

$$\begin{aligned}
\delta\mathbf{r}^f_{gps} &= \mathbf{T}^f_i(t)\mathbf{r}^i_{imu} - \mathbf{T}^f_i(t)\hat{\mathbf{r}}^i_{imu} + \mathbf{T}^f_i(t)\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)\mathbf{r}^b_{gps/imu} - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)\mathbf{r}^b_{gps/imu} && (5.15)\\
&+ \boldsymbol{\beta}^f_{pos} - \hat{\boldsymbol{\beta}}^f_{pos} + \boldsymbol{\eta}^f_{pos}\\
&= \mathbf{T}^f_i(t)\delta\mathbf{r}^i_{imu} + \mathbf{T}^f_i(t)[\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)\mathbf{r}^b_{gps/imu} - \hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)\mathbf{r}^b_{gps/imu}] + \delta\boldsymbol{\beta}^f_{pos} + \boldsymbol{\eta}^f_{pos}\\
&= \mathbf{T}^f_i(t)\delta\mathbf{r}^i_{imu} + \mathbf{T}^f_i(t)[\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b) - \hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)]\mathbf{r}^b_{gps/imu} + \delta\boldsymbol{\beta}^f_{pos} + \boldsymbol{\eta}^f_{pos}\ . && (5.16)
\end{aligned}$$

From Eq. 4.15, the true body to inertial reference frame transformation in terms of estimated quantities is estimated as

$$\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b) = \hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b) + \hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\delta\boldsymbol{\alpha}\times]\ , \tag{5.17}$$

and substituting Eq. 5.17 into Eq. 5.15 yields

$$\begin{aligned}
\delta\mathbf{r}^f_{gps} &= \mathbf{T}^f_i(t)\delta\mathbf{r}^i_{imu} + \mathbf{T}^f_i(t)\Big[\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b) + \hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\delta\boldsymbol{\alpha}\times] - \hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)\Big]\mathbf{r}^b_{gps/imu} + \delta\boldsymbol{\beta}^f_{gps} + \boldsymbol{\eta}^f_{gps}\\
&= \mathbf{T}^f_i(t)\delta\mathbf{r}^i_{imu} + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\delta\boldsymbol{\alpha}\times]\mathbf{r}^b_{gps/imu} + \delta\boldsymbol{\beta}^f_{pos} + \boldsymbol{\eta}^f_{pos}\ . && (5.18)
\end{aligned}$$

which, using the definition in Eq. 2.9 to bring out $\delta\boldsymbol{\alpha}$ from the cross product operator, can be written in final form as

$$\delta\mathbf{r}^f_{gps} = \mathbf{T}^f_i(t)\delta\mathbf{r}^i_{imu} - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\delta\boldsymbol{\alpha} + \delta\boldsymbol{\beta}^f_{pos} + \boldsymbol{\eta}^f_{pos}\ . \tag{5.19}$$

5.4 GPS Velocity

In this section, the sensor model for the velocity measurement is presented. The GPS reports velocity in the ECEF reference frame. The state to measurement mapping equation yields the velocity measurement in terms of state variables, therefore a model for how the velocity measurement is related to the IMU velocity is needed. This model includes the error terms that are associated with the velocity measurement. In the case of the GPS, the error terms are a fixed random bias and random noise. Figure 5.1 illustrates the relationship between the IMU and GPS positions.

As illustrated in Figure 5.1, the position of the GPS with respect to the IMU in the inertial reference frame is given by

$$\mathbf{r}^f_{gps} = \mathbf{r}^f_{imu} + \mathbf{r}^f_{gps/imu} \, . \tag{5.20}$$

Taking the derivative of Eq. 5.20 yields the velocity, given by

$$\mathbf{v}^f_{gps} = \mathbf{v}^f_{imu} + \boldsymbol{\omega}^f_{b/i} \times \mathbf{r}^f_{gps/imu} \, .$$

The velocity measurement is modeled as being corrupted with a fixed random bias and random noise terms, hence we have the model

$$\mathbf{v}^f_{gps} = [\mathbf{v}^f_{imu} + \boldsymbol{\omega}^f_{b/i} \times \mathbf{r}^f_{gps/imu}] + \boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel} \, .$$

The vector $\mathbf{r}_{gps/imu}$ is more naturally represented in the body reference frame, not in the ECEF reference frame. Using the body to ECEF transformation matrix, the sensor model translates to

$$\mathbf{v}^f_{gps} = \left[\mathbf{v}^f_{imu} + \mathbf{T}^f_b[\boldsymbol{\omega}^b_{b/i} \times \mathbf{r}^b_{gps/imu}]\right] + \boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel} \, .$$

Using Eq. 5.14, we have

$$\begin{aligned}\mathbf{v}^f_{gps} &= \left[\mathbf{v}^f_{imu} + \mathbf{T}^f_i(t)\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)[\boldsymbol{\omega}^b_{b/i} \times \mathbf{r}^b_{gps/imu}]\right] + \boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel} \\ &= \mathbf{v}^f_{imu} + \mathbf{T}^f_i(t)\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)[\boldsymbol{\omega}^b_{b/i}\times]\mathbf{r}^b_{gps/imu} + \boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel}\ .\end{aligned}$$

The vector $\mathbf{v}_{imu}$ is estimated in the inertial reference frame, not the ECEF reference frame. Using the inertial to fixed reference frame transformation, we have

$$\mathbf{v}^f_{gps} = \mathbf{T}^f_i(t)\mathbf{v}^i_{imu} + \mathbf{T}^f_i(t)\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)[\boldsymbol{\omega}^b_{b/i}\times]\mathbf{r}^b_{gps/imu} + \boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel}\ ,$$

and using the definition from Eq. 2.9, the final form of the measurement model is given as

$$\mathbf{v}^f_{gps} = \mathbf{T}^f_i(t)\mathbf{v}^i_{imu} - \mathbf{T}^f_i(t)\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\boldsymbol{\omega}^b_{b/i} + \boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel}\ . \tag{5.21}$$

The estimated velocity measurement is

$$\hat{\mathbf{v}}^f_{gps} = \mathbf{T}^f_i(t)\hat{\mathbf{v}}^i_{imu} - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\hat{\boldsymbol{\omega}}^b_{b/i} + \hat{\boldsymbol{\beta}}^f_{vel}\ .$$

With the estimation error of the velocity measurement defined as $\delta\mathbf{v}^f_{gps} = \mathbf{v}^f_{gps} - \hat{\mathbf{v}}^f_{gps}$, we have

$$\begin{aligned}\delta\mathbf{v}^f_{gps} &= \mathbf{T}^f_i(t)\mathbf{v}^i_{imu} - \mathbf{T}^f_i(t)\hat{\mathbf{v}}^i_{imu} - \mathbf{T}^f_i(t)\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\boldsymbol{\omega}^b_{b/i} \\ &\quad + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\hat{\boldsymbol{\omega}}^b_{b/i} + \boldsymbol{\beta}^f_{vel} - \hat{\boldsymbol{\beta}}^f_{vel} + \boldsymbol{\eta}^f_{vel} \\ &= \mathbf{T}^f_i(t)\delta\mathbf{v}^i_{imu} - \mathbf{T}^f_i(t)\mathbf{T}^i_b(\bar{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\boldsymbol{\omega}^b_{b/i} \\ &\quad + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\hat{\boldsymbol{\omega}}^b_{b/i} + \delta\boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel}\ ,\end{aligned} \tag{5.22}$$

and using the body to inertial reference frame transformation expansion from Eq. 5.17 in Eq. 5.22 we have

$$\begin{aligned}
\delta\mathbf{v}^f_{gps} &= \mathbf{T}^f_i(t)\delta\mathbf{v}^i_{imu} - \mathbf{T}^f_i(t)\left[\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b) + \hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\delta\boldsymbol{\alpha}\times]\right][\mathbf{r}^b_{gps/imu}\times]\boldsymbol{\omega}^b_{b/i} \\
&\quad + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\hat{\boldsymbol{\omega}}^b_{b/i} + \delta\boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel} \\
&= \mathbf{T}^f_i(t)\delta\mathbf{v}^i_{imu} - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\boldsymbol{\omega}^b_{b/i} - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\delta\boldsymbol{\alpha}\times][\mathbf{r}^b_{gps/imu}\times]\boldsymbol{\omega}^b_{b/i} \\
&\quad + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\hat{\boldsymbol{\omega}}^b_{b/i} + \delta\boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel} \\
&= \mathbf{T}^f_i(t)\delta\mathbf{v}^i_{imu} - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\delta\boldsymbol{\omega}^b_{b/i} \\
&\quad - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\delta\boldsymbol{\alpha}\times][\mathbf{r}^b_{gps/imu}\times]\boldsymbol{\omega}^b_{b/i} + \delta\boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel}\,.
\end{aligned} \tag{5.23}$$

With the angular velocity estimation error defined as $\delta\boldsymbol{\omega}^b_{b/i} = \boldsymbol{\omega}^b_{b/i} - \hat{\boldsymbol{\omega}}^b_{b/i}$, we can rewrite Eq. 5.23 as

$$\begin{aligned}
\delta\mathbf{v}^f_{gps} &= \mathbf{T}^f_i(t)\delta\mathbf{v}^i_{imu} - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\delta\boldsymbol{\omega}^b_{b/i} \\
&\quad - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\delta\boldsymbol{\alpha}\times][\mathbf{r}^b_{gps/imu}\times]\left[\delta\boldsymbol{\omega}^b_{b/i} + \hat{\boldsymbol{\omega}}^b_{b/i}\right] + \delta\boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel} \\
&= \mathbf{T}^f_i(t)\delta\mathbf{v}^i_{imu} - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\delta\boldsymbol{\omega}^b_{b/i} - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\delta\boldsymbol{\alpha}\times][\mathbf{r}^b_{gps/imu}\times]\delta\boldsymbol{\omega}^b_{b/i} \\
&\quad - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\delta\boldsymbol{\alpha}\times][\mathbf{r}^b_{gps/imu}\times]\hat{\boldsymbol{\omega}}^b_{b/i} + \delta\boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel}\,.
\end{aligned} \tag{5.24}$$

Neglecting second-order terms in Eq. 5.24, the velocity measurement deviation is given as

$$\begin{aligned}
\delta\mathbf{v}^f_{gps} &= \mathbf{T}^f_i(t)\delta\mathbf{v}^i_{imu} - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\delta\boldsymbol{\omega}^b_{b/i} \\
&\quad - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\delta\boldsymbol{\alpha}\times][\mathbf{r}^b_{gps/imu}\times]\hat{\boldsymbol{\omega}}^b_{b/i} + \delta\boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel}\,,
\end{aligned}$$

and using the definition in Eq. 2.9 and Eq. 5.13 we obtain

$$\begin{aligned}
\delta\mathbf{v}^f_{gps} &= \mathbf{T}^f_i(t)\delta\mathbf{v}^i_{imu} + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{gps/imu}\times][\boldsymbol{\omega}^b_{b/i,m}\times]\delta\mathbf{m}_g \\
&+ \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{gps/imu}\times][\boldsymbol{\omega}^b_{b/i,m}|\times|]\delta\mathbf{n}_g + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{gps/imu}\times][\boldsymbol{\omega}^b_{b/i,m}\diagdown]\delta\mathbf{s}_g \\
&+ \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\delta\boldsymbol{\beta}_g + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\boldsymbol{\eta}_g \\
&+ \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)\left[\left[\mathbf{r}^b_{gps/imu}\times\hat{\boldsymbol{\omega}}^b_{b/i}\right]\times\right]\delta\boldsymbol{\alpha} + \delta\boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel}\,. \qquad (5.25)
\end{aligned}$$

5.5 Quaternion Sensor

In this section, the sensor model for the attitude quaternion measurement presented. We assume that the MEMS IMU includes a magnetometer that measures Earth's magnetic field as a vector in the magnetometer reference frame (not necessarily coincident with the IMU case reference frame). Given two specific vector pairs, we can compute a transformation matrix estimate using the TRIAD algorithm, which can then be converted to a quaternion representation. For now, it is assumed that a quaternion measurement is available for purposes of deriving the measurement deviation equation.

Since the attitude quaternion measurement is obtained from the TRIAD algorithm, the measurement will be represented in the magnetometer reference frame which will be referred to as the TRIAD reference frame, *tr*. The TRIAD algorithm uses magnetometer data as an input, hence the TRIAD reference frame is coincident with the magnetometer reference frame. The quaternion obtained from the TRIAD algorithm is

$$\bar{\mathbf{q}}^{tr}_i = \bar{\mathbf{q}}^{tr}_b \otimes \bar{\mathbf{q}}^b_i\,,$$

where $\bar{\mathbf{q}}^{tr}_b$ is assumed known. The measured quaternion is assumed to be corrupted with a fixed random bias and random noise. We incorporate the bias and noise through quaternion multiplication. The error quaternion is defined as

$$\bar{\mathbf{q}}^{tr}_{\beta,\eta} = \begin{bmatrix} \frac{\boldsymbol{\theta}_e}{\|\boldsymbol{\theta}_e\|} \sin \frac{\|\boldsymbol{\theta}_e\|}{2} \\ \cos \frac{\|\boldsymbol{\theta}_e\|}{2} \end{bmatrix} , \tag{5.26}$$

where

$$\boldsymbol{\theta}_e = \boldsymbol{\beta}^{tr}_q + \boldsymbol{\eta}^{tr}_q \, .$$

The TRIAD quaternion sensor model is then given by

$$\bar{\mathbf{q}}^{tr}_{i,m} = \bar{\mathbf{q}}^{tr}_{\beta,\eta} \otimes \bar{\mathbf{q}}^{tr}_b \otimes \bar{\mathbf{q}}^b_i \, .$$

The estimated quaternion measurement is

$$\hat{\bar{\mathbf{q}}}^{tr}_i = \hat{\bar{\mathbf{q}}}^{tr}_{\beta,\eta} \otimes \bar{\mathbf{q}}^{tr}_b \otimes \hat{\bar{\mathbf{q}}}^b_i \, , \tag{5.27}$$

where

$$\hat{\bar{\mathbf{q}}}^{tr}_{\beta,\eta} = \begin{bmatrix} \frac{\hat{\boldsymbol{\theta}}_e}{\|\hat{\boldsymbol{\theta}}_e\|} \sin \frac{\|\hat{\boldsymbol{\theta}}_e\|}{2} \\ \cos \frac{\|\hat{\boldsymbol{\theta}}_e\|}{2} \end{bmatrix} ,$$

and

$$\hat{\boldsymbol{\theta}}_e = \hat{\boldsymbol{\beta}}^{tr}_q \, .$$

With the measurement deviation defined as $\delta\bar{\mathbf{q}}^{tr}_i = \bar{\mathbf{q}}^{tr}_{i,m} \otimes [\hat{\bar{\mathbf{q}}}^{tr}_i]^{-1}$ yields

$$\begin{aligned} \delta\bar{\mathbf{q}}^{tr}_i &= \bar{\mathbf{q}}^{tr}_{\beta,\eta} \otimes \bar{\mathbf{q}}^{tr}_b \otimes \bar{\mathbf{q}}^b_i \otimes [\hat{\bar{\mathbf{q}}}^{tr}_{\beta,\eta} \otimes \bar{\mathbf{q}}^{tr}_b \otimes \hat{\bar{\mathbf{q}}}^b_i]^{-1} \\ &= \bar{\mathbf{q}}^{tr}_{\beta,\eta} \otimes \bar{\mathbf{q}}^{tr}_b \otimes \bar{\mathbf{q}}^b_i \otimes [\hat{\bar{\mathbf{q}}}^b_i]^{-1} \otimes [\bar{\mathbf{q}}^{tr}_b]^{-1} \otimes [\hat{\bar{\mathbf{q}}}^{tr}_{\beta,\eta}]^{-1} \, . \end{aligned} \tag{5.28}$$

With $\delta\bar{\mathbf{q}}^b_i = \bar{\mathbf{q}}^b_i \otimes [\hat{\bar{\mathbf{q}}}^b_i]^{-1}$, Eq. 5.28 reduces to

$$\delta\bar{\mathbf{q}}^{tr}_i = \bar{\mathbf{q}}^{tr}_{\beta,\eta} \otimes \bar{\mathbf{q}}^{tr}_b \otimes \delta\bar{\mathbf{q}}^b_i \otimes [\bar{\mathbf{q}}^{tr}_b]^{-1} \otimes [\hat{\bar{\mathbf{q}}}^{tr}_{\beta,\eta}]^{-1} \, . \tag{5.29}$$

Recalling Eq. 2.34, we can write

$$\bar{\mathbf{q}}_b^{tr} \otimes \delta\bar{\mathbf{q}}_i^b \otimes [\bar{\mathbf{q}}_b^{tr}]^{-1} = \begin{bmatrix} \mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \\ \delta q_{0_i}^b \end{bmatrix} , \tag{5.30}$$

where $\mathbf{T}_b^{tr}$ is the body to TRIAD reference frame transformation. Substituting Eq. 5.30 into Eq. 5.29 yields

$$\delta\bar{\mathbf{q}}_i^{tr} = \bar{\mathbf{q}}_{\beta,\eta}^{tr} \otimes \begin{bmatrix} \mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \\ \delta q_{0_i}^b \end{bmatrix} \otimes [\hat{\bar{\mathbf{q}}}_{\beta,\eta}^{tr}]^{-1} . \tag{5.31}$$

With the definition of the quaternion deviation $\bar{\mathbf{q}}_{\beta,\eta}^{tr}$ yields

$$\delta\bar{\mathbf{q}}_{\beta,\eta}^{tr} = \bar{\mathbf{q}}_{\beta,\eta}^{tr} \otimes [\hat{\bar{\mathbf{q}}}_{\beta,\eta}^{tr}]^{-1} .$$

we find that

$$\bar{\mathbf{q}}_{\beta,\eta}^{tr} = \delta\bar{\mathbf{q}}_{\beta,\eta}^{tr} \otimes \bar{\mathbf{q}}_{\beta,\eta}^{tr} ,$$

which, when substituted into Eq. 5.31 yields

$$\delta\bar{\mathbf{q}}_i^{tr} = \delta\bar{\mathbf{q}}_{\beta,\eta}^{tr} \otimes \hat{\bar{\mathbf{q}}}_{\beta,\eta}^{tr} \otimes \begin{bmatrix} \mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \\ \delta q_{0_i}^b \end{bmatrix} \otimes [\hat{\bar{\mathbf{q}}}_{\beta,\eta}^{tr}]^{-1} . \tag{5.32}$$

Once again, recalling Eq. 2.34, we can write

$$\hat{\bar{\mathbf{q}}}_{\beta,\eta}^{tr} \otimes \begin{bmatrix} \mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \\ \delta q_{0_i}^b \end{bmatrix} \otimes [\hat{\bar{\mathbf{q}}}_{\beta,\eta}^{tr}]^{-1} = \begin{bmatrix} \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \\ q_{0_i}^b \end{bmatrix} ,$$

and it follows that Eq. 5.32 simplifies to

$$\delta\bar{\mathbf{q}}_i^{tr} = \delta\bar{\mathbf{q}}_{\beta,\eta}^{tr} \otimes \begin{bmatrix} \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \\ \delta q_{0_i}^b \end{bmatrix} . \tag{5.33}$$

Expanding Eq. 5.33 using the quaternion product yields

$$\delta\bar{\mathbf{q}}_i^{tr} = \begin{bmatrix} \delta\mathbf{q}_{\beta,\eta}^{tr}\delta q0_i^b + \delta q0_{\beta,\eta}^{tr}\hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}_b^{tr}\delta\mathbf{q}_i^b - \delta\mathbf{q}_{\beta,\eta}^{tr} \times \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \\ \delta q0_{\beta,\eta}^{tr}\delta q0_i^b - \delta\mathbf{q}_{\beta,\eta}^{tr} \cdot \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \end{bmatrix} . \tag{5.34}$$

Neglecting higher order terms, and under the small angle quaternion assumption, the quaternion measurement deviation in Eq. 5.34 becomes

$$\delta\bar{\mathbf{q}}_i^{tr} = \begin{bmatrix} \delta\mathbf{q}_{\beta,\eta}^{tr} + \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \\ 1 \end{bmatrix} . \tag{5.35}$$

The error quaternion defined in Eq. 5.26 quantifies the bias and noise in terms of a quaternion. If this quaternion is assumed to be a small angle quaternion, then

$$\bar{\mathbf{q}}_{\beta,\eta}^{tr} \approx \begin{bmatrix} \frac{1}{2}\boldsymbol{\theta}_e \\ 1 \end{bmatrix} \quad \text{and} \quad \hat{\bar{\mathbf{q}}}_{\beta,\eta}^{tr} \approx \begin{bmatrix} \frac{1}{2}\hat{\boldsymbol{\theta}}_e \\ 1 \end{bmatrix} ,$$

therefore,

$$\delta\bar{\mathbf{q}}_{\beta,\eta}^{tr} = \begin{bmatrix} \frac{1}{2}\boldsymbol{\theta}_e \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{2}\hat{\boldsymbol{\theta}}_e \\ 1 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{2}\boldsymbol{\theta}_e \\ 1 \end{bmatrix} \otimes \begin{bmatrix} -\frac{1}{2}\hat{\boldsymbol{\theta}}_e \\ 1 \end{bmatrix} .$$

Expanding $\delta\bar{\mathbf{q}}_{\beta,\eta}^{tr}$ using the quaternion product yields

$$\delta\bar{\mathbf{q}}_{\beta,\eta}^{tr} = \begin{bmatrix} \frac{1}{2}\boldsymbol{\theta}_e - \frac{1}{2}\hat{\boldsymbol{\theta}}_e + \frac{1}{2}\boldsymbol{\theta}_e \times \frac{1}{2}\hat{\boldsymbol{\theta}}_e \\ 1 + \frac{1}{2}\boldsymbol{\theta}_e \cdot \frac{1}{2}\hat{\boldsymbol{\theta}}_e \end{bmatrix} . \tag{5.36}$$

Since $\boldsymbol{\theta}_e$ and $\hat{\boldsymbol{\theta}}_e$ are small angle vectors, Eq. 5.36 is reduced to

$$\delta\bar{\mathbf{q}}_{\beta,\eta}^{tr} = \begin{bmatrix} \frac{1}{2}(\boldsymbol{\theta}_e - \hat{\boldsymbol{\theta}}_e) \\ 1 \end{bmatrix} . \tag{5.37}$$

Substituting the simplified version of $\delta\mathbf{q}_{\beta,\eta}^{tr}$ in Eq. 5.37 into Eq. 5.35 yields

$$\begin{aligned}
\delta\bar{\mathbf{q}}_i^{tr} &= \begin{bmatrix} \frac{1}{2}(\boldsymbol{\theta}_e - \hat{\boldsymbol{\theta}}_e) + \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{2}(\boldsymbol{\beta}_q^{tr} - \hat{\boldsymbol{\beta}}_q^{tr} + \boldsymbol{\eta}_q^{tr}) + \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{2}(\delta\boldsymbol{\beta}_q^{tr} + \boldsymbol{\eta}_q^{tr}) + \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}_b^{tr}\delta\mathbf{q}_i^b \\ 1 \end{bmatrix} .
\end{aligned} \tag{5.38}$$

Applying the definition of the small angle quaternion to $\delta\bar{\mathbf{q}}_i^{tr}$ and $\delta\mathbf{q}_i^b$, namely

$$\delta\bar{\mathbf{q}}_i^{tr} \approx \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\Psi} \\ 1 \end{bmatrix} \qquad \text{and} \qquad \delta\bar{\mathbf{q}}_i^{b} \approx \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\alpha} \\ 1 \end{bmatrix} ,$$

Eq. 5.38 becomes

$$\begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\Psi} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(\delta\boldsymbol{\beta}_q^{tr} + \boldsymbol{\eta}_q^{tr}) + \frac{1}{2}\hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}_b^{tr}\delta\boldsymbol{\alpha} \\ 1 \end{bmatrix} . \tag{5.39}$$

Therefore, the final form of the quaternion measurement deviation is

$$\delta\boldsymbol{\Psi} = \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}_b^{tr}\delta\boldsymbol{\alpha} + \delta\boldsymbol{\beta}_q^{tr} + \boldsymbol{\eta}_q^{tr} . \tag{5.40}$$

5.5.1 TRIAD Algorithm

For purposes of modeling the Kalman filter equations, it was assumed that there is an onboard sensor that can provide direct quaternion measurements. However, this is not the case. This section presents the procedure for attitude determination using the TRIAD algorithm to utilize the magnetometer along with the GPS and an on-board model of Earth's magnetic field to provide a computed attitude measurement.

5.5.1.1 *TRIAD Algorithm Definition*

The TRIAD algorithm can compute an attitude estimate given two vector pairs described in an inertial and body reference frame, respectively. Typically, the inertial reference frame description is known. Examples of known inertial vectors are gravity on Earth, the direction of the sun, Earth's magnetic field, and star locations. Body reference frame descriptions are measured by sensors, and both are used to retrieve attitude information. Recall that a single pair of vectors related by

$$\mathbf{v}^b = \mathbf{T}_i^b \mathbf{v}^i \; ,$$

can yield an infinite number of solutions for $\mathbf{T}_i^b$ because there exists an infinite number of transformations that can map the inertial vector to the body vector. Even then, the true transformation cannot be determined through a single pair of vectors. With that, the TRIAD algorithm utilizes two pairs to compute the attitude of an object. Given two vectors $\mathbf{r}$ and $\mathbf{v}$ in body and inertial reference frames,

$$(\mathbf{r}^i, \mathbf{r}^b) \quad \text{and} \quad (\mathbf{v}^i, \mathbf{v}^b) \; ,$$

the inertial to body transformation matrix is then

$$\mathbf{T}_i^b = \begin{bmatrix} \mathbf{t}_{1b} & \mathbf{t}_{2b} & \mathbf{t}_{3b} \end{bmatrix} \begin{bmatrix} \mathbf{t}_{1i} & \mathbf{t}_{2i} & \mathbf{t}_{3i} \end{bmatrix}^T \; , \tag{5.41}$$

where the vectors of the two matrices can either be

$$\mathbf{t}_{1b} = \frac{\mathbf{r}^b}{\|\mathbf{r}^b\|} \; , \qquad \mathbf{t}_{2b} = \frac{\mathbf{r}^b \times \mathbf{v}^b}{\|\mathbf{r}^b \times \mathbf{v}^b\|} \; , \qquad \mathbf{t}_{3b} = \mathbf{t}_{1b} \times \mathbf{t}_{2b} \; , \tag{5.42}$$

$$\mathbf{t}_{1i} = \frac{\mathbf{r}^i}{\|\mathbf{r}^i\|} \; , \qquad \mathbf{t}_{2i} = \frac{\mathbf{r}^i \times \mathbf{v}^i}{\|\mathbf{r}^i \times \mathbf{v}^i\|} \; , \qquad \text{and} \qquad \mathbf{t}_{3i} = \mathbf{t}_{1i} \times \mathbf{t}_{2i} \; , \tag{5.43}$$

or

$$\mathbf{t}_{1b} = \frac{\mathbf{v}^b}{\|\mathbf{v}^b\|}\,, \qquad \mathbf{t}_{2b} = \frac{\mathbf{v}^b \times \mathbf{r}^b}{\|\mathbf{v}^b \times \mathbf{r}^b\|}\,, \qquad \mathbf{t}_{3b} = \mathbf{t}_{1b} \times \mathbf{t}_{2b}\,, \tag{5.44}$$

$$\mathbf{t}_{1i} = \frac{\mathbf{v}^i}{\|\mathbf{v}^i\|}\,, \qquad \mathbf{t}_{2i} = \frac{\mathbf{v}^i \times \mathbf{r}^i}{\|\mathbf{v}^i \times \mathbf{r}^i\|}\,, \qquad \text{and} \qquad \mathbf{t}_{3i} = \mathbf{t}_{1i} \times \mathbf{t}_{2i}\,. \tag{5.45}$$

Either set will yield the correct transformation matrix, assuming both vectors are exactly known with zero uncertainty. In the more realistic case that both vectors have associated uncertainties in their information, the one with the least uncertainty is utilized for $\mathbf{t}_{1b}$ and $\mathbf{t}_{1i}$. This allows the more accurate vector to influence the accuracy of the overall attitude estimate. The attitude quaternion can be found by modifying Eq. 2.29 to use the transformation matrix as an input, rather than the rotation matrix, by simply transposing the transformation matrix obtained from the TRIAD method,

$$\bar{\mathbf{q}}_i^b = \begin{bmatrix} \mathbf{q}_i^b \\ q0_i^b \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \frac{\mathbf{T}^T_{3,2} - \mathbf{T}^T_{2,3}}{\sqrt{1+\mathbf{T}^T_{1,1}+\mathbf{T}^T_{2,2}+\mathbf{T}^T_{3,3}}} \\ \frac{\mathbf{T}^T_{1,3} - \mathbf{T}^T_{3,1}}{\sqrt{1+\mathbf{T}^T_{1,1}+\mathbf{T}^T_{2,2}+\mathbf{T}^T_{3,3}}} \\ \frac{\mathbf{T}^T_{2,1} - \mathbf{T}^T_{1,2}}{\sqrt{1+\mathbf{T}^T_{1,1}+\mathbf{T}^T_{2,2}+\mathbf{T}^T_{3,3}}} \\ \sqrt{1 + \mathbf{T}^T_{1,1} + \mathbf{T}^T_{2,2} + \mathbf{T}^T_{3,3}} \end{bmatrix}.$$

5.5.1.2 TRIAD Algorithm Application

Utilizing our on-board resources, the attitude can be computed. A model of Earth's magnetic field can provide the magnetic field in the NED reference frame $\mathbf{b}^{ned}$. Given the position of the satellite and the magnetometer with respect to the satellite, we can also compute $\dot{\mathbf{b}}^{ned}$ through a finite difference approximation between two short intervals of time (in this case, between Kalman filter updates). We can also obtain both $\mathbf{b}^{tr}$ and $\dot{\mathbf{b}}^{tr}$ from the magnetometer. Therefore, we can utilize the two pairs of vectors to obtain an attitude measurement. Our TRIAD vector pairs are $(\mathbf{b}^{tr}, \mathbf{b}^{ned})$ and $(\dot{\mathbf{b}}^{tr}, \dot{\mathbf{b}}^{ned})$. The

TRIAD algorithm is then employed, using $(\mathbf{b}^{tr}, \mathbf{b}^{ned})$ as the first pair since it has a lower associated uncertainty than $(\dot{\mathbf{b}}^{tr}, \dot{\mathbf{b}}^{ned})$. See Appendix C for a detailed error analysis. We then have

$$\mathbf{T}_{ned}^{tr} = \begin{bmatrix} \mathbf{t}_{1tr} & \mathbf{t}_{2tr} & \mathbf{t}_{3tr} \end{bmatrix} \begin{bmatrix} \mathbf{t}_{1ned} & \mathbf{t}_{2ned} & \mathbf{t}_{3ned} \end{bmatrix}^T , \tag{5.46}$$

with

$$\mathbf{t}_{1tr} = \frac{\mathbf{b}^{tr}}{\left\|\mathbf{b}^{tr}\right\|} , \qquad \mathbf{t}_{2tr} = \frac{\mathbf{b}^{tr} \times \dot{\mathbf{b}}^{tr}}{\left\|\mathbf{b}^{tr} \times \dot{\mathbf{b}}^{tr}\right\|} , \qquad \mathbf{t}_{3tr} = \mathbf{t}_{1tr} \times \mathbf{t}_{2tr} , \tag{5.47}$$

$$\mathbf{t}_{1ned} = \frac{\mathbf{b}^{ned}}{\left\|\mathbf{b}^{ned}\right\|} , \qquad \mathbf{t}_{2ned} = \frac{\mathbf{b}^{ned} \times \dot{\mathbf{b}}^{ned}}{\left\|\mathbf{b}^{ned} \times \dot{\mathbf{b}}^{ned}\right\|} , \qquad \text{and} \qquad \mathbf{t}_{3ned} = \mathbf{t}_{1ned} \times \mathbf{t}_{2ned} . \tag{5.48}$$

Note that this yields the NED to TRIAD transformation, not the inertial to TRIAD transformation. To obtain the inertial to TRIAD transformation, we must first compute the inertial to NED transformation, which is given by

$$\mathbf{T}_i^{ned} = \mathbf{T}_f^{ned}(\phi, \theta)\mathbf{T}_i^f(t) ,$$

where (ϕ, θ) are the latitude and longitude, respectively. Each transformation is presented in the following subsections.

5.5.1.3 ECEF to NED

The ECEF to NED transformation is a sequence of single axis transformation matrices that are dependent on latitude ϕ and longitude θ using the spherical coordinate system convention in Figure 5.2. The transformation sequence is given as

$$\mathbf{T}_f^{ned}(\phi, \theta) = \mathbf{T}_y(-90^\circ)\mathbf{T}_x(\theta)\mathbf{T}_y(-\phi) . \tag{5.49}$$

Expanding Eq. 5.49, we have

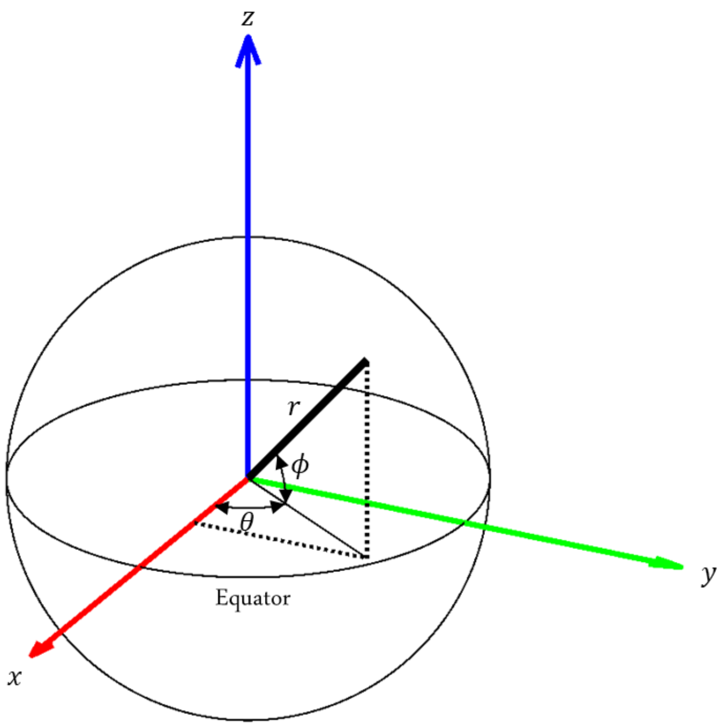


Figure 5.2: Spherical Coordinate System Convention

$$
\begin{aligned}
\mathbf{T}_f^{ned}(\phi,\theta) &= \mathbf{T}_y(-90^\circ)\mathbf{T}_x(\theta)\mathbf{T}_y(-\phi) \\
&= \begin{bmatrix} \cos(-90^\circ) & 0 & -\sin(-90^\circ) \\ 0 & 1 & 0 \\ \sin(-90^\circ) & 0 & \cos(-90^\circ) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(-\phi) & 0 & -\sin(-\phi) \\ 0 & 1 & 0 \\ \sin(-\phi) & 0 & \cos(-\phi) \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \qquad (5.50) \\
&= \begin{bmatrix} 0 & -\sin(\theta) & \cos(\theta) \\ 0 & \cos(\theta) & \sin(\theta) \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix},
\end{aligned}
$$

which yields the ECEF to NED transformation,

$$\mathbf{T}_f^{ned}(\phi,\theta) = \begin{bmatrix} -\cos\theta\sin\phi & -\sin\theta & \cos\theta\cos\phi \\ -\sin\theta\sin\phi & \cos\theta & \sin\theta\cos\phi \\ -\cos\theta & 0 & -\sin\phi \end{bmatrix}. \tag{5.51}$$

5.5.1.4 ECI to ECEF

The fixed reference frame parameters are defined for Earth as well as other planets and satellites by the IAU/IAG/COSPAR Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites. These definitions vary from planet to planet and satellite to satellite. The parameters for Earth's inertial to fixed reference frame transformation are visualized in Figure 5.3. These parameters define the location of the north pole and prime meridian of the planet with respect to the inertial reference frame and are the right ascension of the north pole α_0, the declination of the north pole

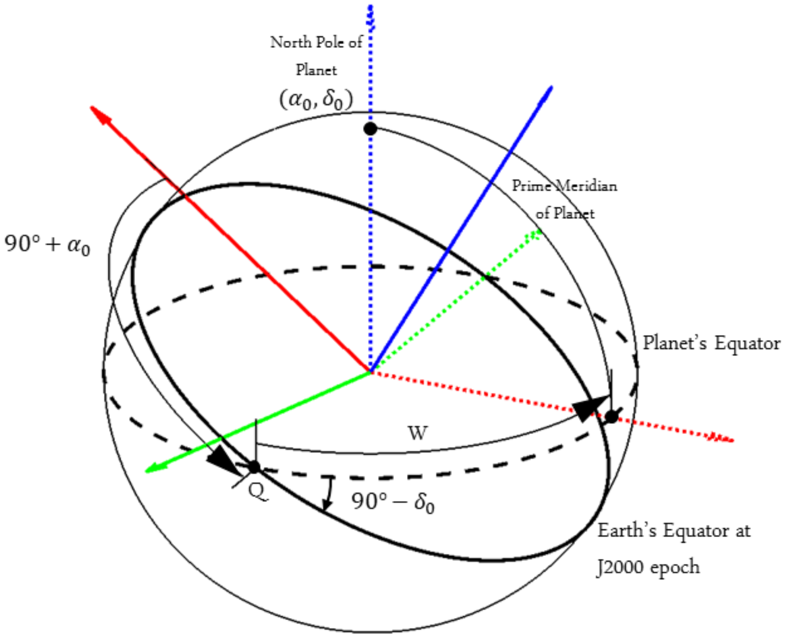


Figure 5.3: Fixed Frame Planetary Reference System

δ_0, and the twist angle W, where

$$\begin{aligned}\alpha_0 &= 0.00 - 0.641T\ ,\\ \delta_0 &= 90.00 - 0.557T\ ,\\ W &= 190.147 + 360.9856235d\ ,\end{aligned}$$

and where T is the interval in Julian centuries (of 36525 days) from the standard J2000 epoch and d is the interval in days from the standard J2000 epoch. Given these parameters, the ECI to ECEF transformation matrix $\mathbf{T}_i^f(t)$ is then given by a 3-1-3 Euler sequence,

$$\mathbf{T}_i^f(t) = \mathbf{T}_z(90^\circ + \alpha_0)\mathbf{T}_x(90^\circ - \delta_0)\mathbf{T}_z(W)\ ,$$

where we note that α_0, δ_0, and W are all functions of time.

Chapter 6: Kalman Filter Implementation

The equations derived for the state dynamics, IMU, and sensors are presented and linearized for state space representation. The position, velocity, small angle deviation, and all error terms associated with the IMU and sensors are the estimated terms that make up the state vector,

$$\mathbf{x} = \left[\begin{array}{c} \mathbf{r}^i_{imu} \\ \mathbf{v}^i_{imu} \\ \delta\boldsymbol{\alpha} \\ \boldsymbol{\beta}^f_{pos} \\ \boldsymbol{\beta}^f_{vel} \\ \boldsymbol{\beta}^{tr}_q \\ \hdashline \mathbf{m}_a \\ \mathbf{n}_a \\ \mathbf{s}_a \\ \boldsymbol{\beta}_a \\ \hdashline \mathbf{m}_g \\ \mathbf{n}_g \\ \mathbf{s}_g \\ \boldsymbol{\beta}_g \end{array}\right] = \begin{bmatrix} \mathbf{r}^i_{imu} \\ \mathbf{v}^i_{imu} \\ \delta\boldsymbol{\alpha} \\ \boldsymbol{\beta}^f_{pos} \\ \boldsymbol{\beta}^f_{vel} \\ \boldsymbol{\beta}^{tr}_q \\ \mathbf{e}_a \\ \mathbf{e}_g \end{bmatrix} \in \mathbb{R}^{42} \,,$$

where $\mathbf{e}_a$ and $\mathbf{e}_g$ are the vectors of systematic error terms for the accelerometer and gyroscope, respectively.

6.1 F Matrix

The estimation error differential equations for position, velocity, and small angle quaternion deviation were derived in Chapter 4 as

$$\begin{aligned} \delta\dot{\mathbf{r}}^i_{imu} &= \delta\mathbf{v}^i_{imu} \\ \delta\dot{\mathbf{v}}^i_{imu} &= \mathbf{G}\delta\mathbf{r}^i_{imu/i} - \mathbf{G}\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{cg/imu}\times]\delta\boldsymbol{\alpha} + \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)\mathbf{T}^b_c\delta\mathbf{a}^c_{imu,ng} - \hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)\mathbf{T}^b_c[\hat{\mathbf{a}}^c_{imu,ng}\times]\delta\boldsymbol{\alpha}\ . \\ \delta\dot{\boldsymbol{\alpha}} &= \delta\boldsymbol{\omega}^b_{b/i} - [\hat{\boldsymbol{\omega}}^b_{b/i}\times]\delta\boldsymbol{\alpha}\ . \end{aligned} \tag{6.1}$$

The accelerometer and gyroscope estimation error equations are written in terms of their systematic errors in Chapter 5 are

$$\begin{aligned} \delta\mathbf{a}^c_{imu,ng} &= -[\mathbf{a}^c_{imu,ng,m}\times]\delta\mathbf{m}_a - [\mathbf{a}^c_{imu,ng,m}|\times|]\delta\mathbf{n}_a - [\mathbf{a}^c_{imu,ng,m}\diagdown]\delta\mathbf{s}_a - \delta\boldsymbol{\beta}_a - \boldsymbol{\eta}_a \\ \delta\boldsymbol{\omega}^c_{b/i} &= -[\boldsymbol{\omega}^c_{b/i,m}\times]\delta\mathbf{m}_g - [\boldsymbol{\omega}^c_{b/i,m}|\times|]\delta\mathbf{n}_g - [\boldsymbol{\omega}^c_{b/i,m}\diagdown]\delta\mathbf{s}_g - \delta\boldsymbol{\beta}_g - \boldsymbol{\eta}_g\ . \end{aligned} \tag{6.2}$$

Substituting $\delta\mathbf{a}^c_{imu,ng}$ and $\delta\boldsymbol{\omega}^c_{b/i}$ in Eq. 6.2 into the state differential equations in Eq. 6.1 yields the final form of the position, velocity, and small angle quaternion deviation differential equations in terms of all the IMU systematic error terms. However, the angular velocity estimation error equation is derived in the IMU case reference frame, therefore, it is transformed into the body reference frame to be substituted into the small angle quaternion deviation differential equation. Applying the transformation to the angular velocity estimation error yields

$$\begin{aligned} \delta\boldsymbol{\omega}^b_{b/i} &= \mathbf{T}^b_c\delta\boldsymbol{\omega}^c_{b/i} \\ &= -\mathbf{T}^b_c[\boldsymbol{\omega}^c_{b/i,m}\times]\delta\mathbf{m}_g - \mathbf{T}^b_c[\boldsymbol{\omega}^c_{b/i,m}|\times|]\delta\mathbf{n}_g - \mathbf{T}^b_c[\boldsymbol{\omega}^c_{b/i,m}\diagdown]\delta\mathbf{s}_g - \mathbf{T}^b_c\delta\boldsymbol{\beta}_g - \mathbf{T}^b_c\boldsymbol{\eta}_g\ , \end{aligned}$$

and

$$\hat{\omega}^{b}_{b/i} = \mathbf{T}^{b}_{c}\hat{\omega}^{c}_{b/i} \,.$$

Making the substitution of the accelerometer and gyroscope estimation error equations into the position, velocity, and small angle quaternion deviation differential equations yields the final form of the differential equation set,

$$\begin{aligned}
\delta\dot{\mathbf{r}}^{i}_{imu} &= \delta\mathbf{v}^{i}_{imu} \\
\delta\dot{\mathbf{v}}^{i}_{imu} &= \mathbf{G}\delta\mathbf{r}^{i}_{imu/i} - \mathbf{G}\hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})[\mathbf{r}^{b}_{cg/imu}\times]\delta\boldsymbol{\alpha} - \hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})\mathbf{T}^{b}_{c}[\mathbf{a}^{c}_{imu,ng,m}\times]\delta\mathbf{m}_{a} \\
&\quad - \hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})\mathbf{T}^{b}_{c}[\mathbf{a}^{c}_{imu,ng,m}|\times|]\delta\mathbf{n}_{a} \; - \hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})\mathbf{T}^{b}_{c}[\mathbf{a}^{c}_{imu,ng,m}\diagdown]\delta\mathbf{s}_{a} \\
&\quad - \hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})\mathbf{T}^{b}_{c}\delta\boldsymbol{\beta}_{a} - \hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})\mathbf{T}^{b}_{c}\boldsymbol{\eta}_{a} - \hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})\mathbf{T}^{b}_{c}[\hat{\mathbf{a}}^{c}_{imu,ng}\times]\delta\boldsymbol{\alpha} \qquad (6.3) \\
\delta\dot{\boldsymbol{\alpha}} &= -\mathbf{T}^{b}_{c}[\boldsymbol{\omega}^{c}_{b/i,m}\times]\delta\mathbf{m}_{g} - \mathbf{T}^{b}_{c}[\boldsymbol{\omega}^{c}_{b/i,m}|\times|]\delta\mathbf{n}_{g} - \mathbf{T}^{b}_{c}[\boldsymbol{\omega}^{c}_{b/i,m}\diagdown]\delta\mathbf{s}_{g} \\
&\quad - \mathbf{T}^{b}_{c}\delta\boldsymbol{\beta}_{g} - \mathbf{T}^{b}_{c}\boldsymbol{\eta}_{g} - \mathbf{T}^{b}_{c}[\hat{\omega}^{c}_{b/i}\times]\delta\boldsymbol{\alpha} \,.
\end{aligned}$$

The biases are modeled as random constants, therefore

$$\delta\dot{\boldsymbol{\beta}}^{f}_{pos} = \mathbf{0} \,,\; \delta\dot{\boldsymbol{\beta}}^{f}_{vel} = \mathbf{0} \,,\; \delta\dot{\boldsymbol{\beta}}^{tr}_{q} = \mathbf{0} \,, \dot{\boldsymbol{\beta}}_{a} = \mathbf{0} \,, \quad \text{and} \quad \dot{\boldsymbol{\beta}}_{g} = \mathbf{0} \,. \qquad (6.4)$$

The IMU systematic errors are also modeled as random constants, therefore

$$\delta\dot{\mathbf{m}}_{a} = \mathbf{0} \,,\; \delta\dot{\mathbf{n}}_{a} = \mathbf{0} \,,\; \delta\dot{\mathbf{s}}_{a} = \mathbf{0} \,,\; \delta\dot{\mathbf{m}}_{g} = \mathbf{0} \,,\; \delta\dot{\mathbf{n}}_{g} = \mathbf{0} \,, \quad \text{and} \quad \delta\dot{\mathbf{s}}_{g} = \mathbf{0} \,. \qquad (6.5)$$

Note that once the state estimates are available, we can correct the estimated non-gravitational acceleration and angular velocity terms via

$$\begin{aligned}
\hat{\mathbf{a}}^{c}_{imu,ng} &= -\mathbf{a}^{c}_{imu,ng,m} + [\mathbf{a}^{c}_{imu,ng,m}\times]\hat{\mathbf{m}}_{a} - [\mathbf{a}^{c}_{imu,ng,m}|\times|]\hat{\mathbf{n}}_{a} - [\mathbf{a}^{c}_{imu,ng,m}\diagdown]\hat{\mathbf{s}}_{a} - \hat{\boldsymbol{\beta}}_{a} \\
\hat{\omega}^{c}_{b/i} &= -\boldsymbol{\omega}^{c}_{b/i,m} + [\boldsymbol{\omega}^{c}_{b/i,m}\times]\hat{\mathbf{m}}_{g} - [\boldsymbol{\omega}^{c}_{b/i,m}|\times|]\hat{\mathbf{n}}_{g} - [\boldsymbol{\omega}^{c}_{b/i,m}\diagdown]\hat{\mathbf{s}}_{g} - \hat{\boldsymbol{\beta}}_{g} \,.
\end{aligned}$$

6.1.1 Jacobian Matrix - Dynamics

The results from Eqs. 6.3, 6.4, and 6.5 are used to determine the Jacobian, **F**. Specifically,

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_{3x3} & \frac{\partial \delta \dot{\mathbf{r}}^i_{imu}}{\partial \mathbf{v}^i_{imu}} & \mathbf{0}_{3x3} & \mathbf{0}_{3x9} & \mathbf{0}_{3x12} & \mathbf{0}_{3x12} \\ \frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \mathbf{r}^i_{imu}} & \mathbf{0}_{3x3} & \frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \delta \boldsymbol{\alpha}} & \mathbf{0}_{3x9} & \frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \mathbf{e}_a} & \mathbf{0}_{3x12} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \frac{\partial \delta \dot{\boldsymbol{\alpha}}}{\partial \delta \boldsymbol{\alpha}} & \mathbf{0}_{3x9} & \mathbf{0}_{3x12} & \frac{\partial \delta \dot{\boldsymbol{\alpha}}}{\partial \mathbf{e}_g} \\ \mathbf{0}_{33x3} & \mathbf{0}_{33x3} & \mathbf{0}_{33x3} & \mathbf{0}_{33x9} & \mathbf{0}_{33x12} & \mathbf{0}_{33x12} \end{bmatrix},$$

where

$$\begin{aligned}
\frac{\partial \delta \dot{\mathbf{r}}^i_{imu}}{\partial \mathbf{v}^i_{imu}} &= \mathbf{I}_{3x3} \\
\frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \mathbf{r}^i_{imu}} &= \mathbf{G} \\
\frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \delta \boldsymbol{\alpha}} &= -\mathbf{G}\hat{\mathbf{T}}^i_b(\hat{\hat{\mathbf{q}}}^i_b)[\mathbf{r}^b_{cg/imu}\times] - \hat{\mathbf{T}}^i_b(\hat{\hat{\mathbf{q}}}^i_b)\mathbf{T}^b_c[\hat{\mathbf{a}}^c_{imu,ng}\times] \\
\frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \mathbf{e}_a} &= \Big[-\hat{\mathbf{T}}^i_b(\hat{\hat{\mathbf{q}}}^i_b)\mathbf{T}^b_c[\mathbf{a}^c_{imu,ng,m}\times] \quad -\hat{\mathbf{T}}^i_b(\hat{\hat{\mathbf{q}}}^i_b)\mathbf{T}^b_c[\mathbf{a}^c_{imu,ng,m}|\times|] \ldots \\
&\quad \ldots -\hat{\mathbf{T}}^i_b(\hat{\hat{\mathbf{q}}}^i_b)\mathbf{T}^b_c[\mathbf{a}^c_{imu,ng,m}\diagdown] \quad -\hat{\mathbf{T}}^i_b(\hat{\hat{\mathbf{q}}}^i_b)\mathbf{T}^b_c \Big] \\
\frac{\partial \delta \dot{\boldsymbol{\alpha}}}{\partial \delta \boldsymbol{\alpha}} &= -\mathbf{T}^b_c[\hat{\omega}^c_{b/i}\times] \\
\frac{\partial \delta \dot{\boldsymbol{\alpha}}}{\partial \mathbf{e}_g} &= \begin{bmatrix} -\mathbf{T}^b_c[\boldsymbol{\omega}^b_{b/i,m}\times] & -\mathbf{T}^b_c[\boldsymbol{\omega}^b_{b/i,m}|\times|] & -\mathbf{T}^b_c[\boldsymbol{\omega}^b_{b/i,m}\diagdown] & -\mathbf{T}^b_c \end{bmatrix}.
\end{aligned}$$

6.2 H Matrix

The measurement deviation equations for the position, velocity, and quaternion were derived in Chapter 5 as

$$
\begin{aligned}
\delta\mathbf{r}^f_{gps} &= \mathbf{T}^f_i(t)\delta\mathbf{r}^i_{imu} - \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\delta\boldsymbol{\alpha} + \delta\boldsymbol{\beta}^f_{pos} + \boldsymbol{\eta}^f_{pos} \\
\delta\mathbf{v}^f_{gps} &= \mathbf{T}^f_i(t)\delta\mathbf{v}^i_{imu} + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times][\boldsymbol{\omega}^b_{b/i,m}\times]\delta\mathbf{m}_g \\
&\quad + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times][\boldsymbol{\omega}^b_{b/i,m}|\times|]\delta\mathbf{n}_g + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times][\boldsymbol{\omega}^b_{b/i,m}\diagdown]\delta\mathbf{s}_g \\
&\quad + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\delta\boldsymbol{\beta}_g + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\boldsymbol{\eta}_g \\
&\quad + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)\left[\left[\mathbf{r}^b_{gps/imu}\times\hat{\boldsymbol{\omega}}^b_{b/i}\right]\times\right]\delta\boldsymbol{\alpha} + \delta\boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel} \\
\delta\boldsymbol{\Psi} &= \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}^{tr}_b\delta\boldsymbol{\alpha} + \delta\boldsymbol{\beta}^{tr}_q + \boldsymbol{\eta}^{tr}_q \,.
\end{aligned}
\tag{6.6}
$$

The velocity measurement deviation is written in terns of the body angular velocity in the body reference frame, however, the body angular velocity equation is written in the IMU case reference frame. Applying the case to body transformation $\mathbf{T}^b_c$ yields the final form of the velocity measurement deviation,

$$
\begin{aligned}
\delta\mathbf{v}^f_{gps} &= \mathbf{T}^f_i(t)\delta\mathbf{v}^i_{imu} + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\mathbf{T}^b_c[\boldsymbol{\omega}^c_{b/i,m}\times]\delta\mathbf{m}_g \\
&\quad + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\mathbf{T}^b_c[\boldsymbol{\omega}^c_{b/i,m}|\times|]\delta\mathbf{n}_g \\
&\quad + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\mathbf{T}^b_c[\boldsymbol{\omega}^c_{b/i,m}\diagdown]\delta\mathbf{s}_g \\
&\quad + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\mathbf{T}^b_c\delta\boldsymbol{\beta}_g + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\mathbf{T}^b_c\boldsymbol{\eta}_g \\
&\quad + \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\mathbf{q}}^i_b)\left[\left[\mathbf{r}^b_{gps/imu}\times\mathbf{T}^b_c\hat{\boldsymbol{\omega}}^c_{b/i}\right]\times\right]\delta\boldsymbol{\alpha} + \delta\boldsymbol{\beta}^f_{vel} + \boldsymbol{\eta}^f_{vel} \,.
\end{aligned}
\tag{6.7}
$$

The estimated angular velocity is corrected via

$$
\hat{\boldsymbol{\omega}}^c_{b/i} = \boldsymbol{\omega}^c_{b/i,m} + [\boldsymbol{\omega}^c_{b/i,m}\times]\hat{\mathbf{m}}_g - [\boldsymbol{\omega}^c_{b/i,m}|\times|]\hat{\mathbf{n}}_g - [\boldsymbol{\omega}^c_{b/i,m}\diagdown]\hat{\mathbf{s}}_g - \hat{\boldsymbol{\beta}}_g \,.
$$

The measurement matrix, **H**, is formed using Eq. 6.6 and Eq. 6.7 as

$$\mathbf{H} = \begin{bmatrix} \frac{\partial \delta \mathbf{r}^f_{gps}}{\partial \mathbf{r}^i_{imu}} & \mathbf{0}_{3x3} & \frac{\partial \delta \mathbf{r}^f_{gps}}{\partial \delta \boldsymbol{\alpha}} & \frac{\partial \delta \mathbf{r}^f_{gps}}{\partial \boldsymbol{\beta}^f_{pos}} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x12} & \mathbf{0}_{3x12} \\ \mathbf{0}_{3x3} & \frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \mathbf{v}^i_{imu}} & \frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \delta \boldsymbol{\alpha}} & \mathbf{0}_{3x3} & \frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \boldsymbol{\beta}^f_{vel}} & \mathbf{0}_{3x3} & \frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \mathbf{e}_a} & \mathbf{0}_{3x12} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \frac{\partial \delta \boldsymbol{\Psi}}{\partial \delta \boldsymbol{\alpha}} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \frac{\partial \delta \boldsymbol{\Psi}}{\partial \boldsymbol{\beta}^{tr}_q} & \mathbf{0}_{3x12} & \mathbf{0}_{3x12} \end{bmatrix},$$

where

$$\frac{\partial \delta \mathbf{r}^f_{gps}}{\partial \mathbf{r}^i_{imu}} = \mathbf{T}^f_i(t)$$

$$\frac{\partial \delta \mathbf{r}^f_{gps}}{\partial \delta \boldsymbol{\alpha}} = -\mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{gps/imu}\times]$$

$$\frac{\partial \delta \mathbf{r}^f_{gps}}{\partial \boldsymbol{\beta}^f_{pos}} = \mathbf{I}_{3x3}$$

$$\frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \mathbf{v}^i_{imu}} = \mathbf{T}^f_i(t)$$

$$\frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \delta \boldsymbol{\alpha}} = \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)\left[\left[\mathbf{r}^b_{gps/imu} \times \mathbf{T}^b_c \hat{\boldsymbol{\omega}}^c_{b/i}\right] \times \right]$$

$$\frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \boldsymbol{\beta}^f_{vel}} = \mathbf{I}_{3x3}$$

$$\frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \mathbf{e}_a} = \Big[\mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\mathbf{T}^b_c[\boldsymbol{\omega}^c_{b/i,m}\times] \quad \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\mathbf{T}^b_c[\boldsymbol{\omega}^c_{b/i,m}|\times|] \ldots$$

$$\ldots \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\mathbf{T}^b_c[\boldsymbol{\omega}^c_{b/i,m}\diagdown] \quad \mathbf{T}^f_i(t)\hat{\mathbf{T}}^i_b(\hat{\bar{\mathbf{q}}}^i_b)[\mathbf{r}^b_{gps/imu}\times]\mathbf{T}^b_c\Big]$$

$$\frac{\partial \delta \boldsymbol{\Psi}}{\partial \delta \boldsymbol{\alpha}} = \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}^{tr}_b$$

$$\frac{\partial \delta \boldsymbol{\Psi}}{\partial \boldsymbol{\beta}^{tr}_q} = \mathbf{I}_{3x3}\,.$$

6.3 Propagation

The state vector and state estimation error covariance are propagated between updates at t_k and t_{k-1}, which is assumed constant for our simulations. The update rate is limited by the sample rates of the sensors while the propagation rate is set by the clock source of the embedded processor and limited by computation time. For our simulation, the time between computations $\Delta t_j = t_j - t_{j-1}$ is assumed constant and set by the propagation frequency. Figure 6.1 depicts the Kalman filter timeline with a focus on propagation between measurements.

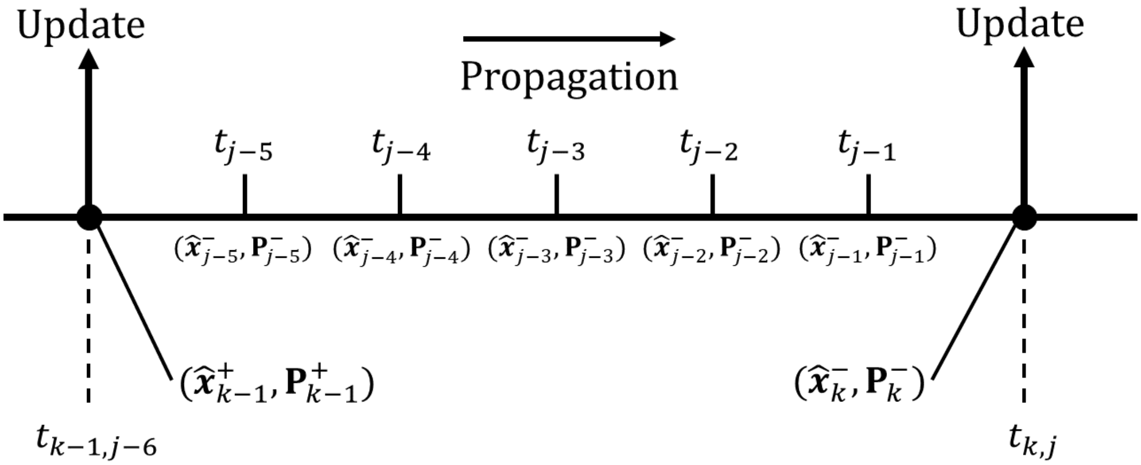

Update
Propagation
Update
t_{j-5}
t_{j-4}
t_{j-3}
t_{j-2}
t_{j-1}
$(\hat{x}^-_{j-5}, \mathbf{P}^-_{j-5})$
$(\hat{x}^-_{j-4}, \mathbf{P}^-_{j-4})$
$(\hat{x}^-_{j-3}, \mathbf{P}^-_{j-3})$
$(\hat{x}^-_{j-2}, \mathbf{P}^-_{j-2})$
$(\hat{x}^-_{j-1}, \mathbf{P}^-_{j-1})$
$(\hat{x}^+_{k-1}, \mathbf{P}^+_{k-1})$
$(\hat{x}^-_{k}, \mathbf{P}^-_{k})$
$t_{k-1,j-6}$
$t_{k,j}$

Figure 6.1: MEKF Propagation Timeline

6.3.1 State Propagation

A good choice for solving differential equations numerically is the RK4 tool. Ideally, this would be a good way to propagate the states between measurements. However, recall that attitude quaternions can't be added or subtracted with each other to produce another attitude quaternion. Instead, they have a special multiplication process that describes a sequence of positive or negative rotations. Because of this, the RK4 tool can't be used to propagate the attitude quaternion. Instead, the attitude will have a special propagation method.

It can be assumed because of the small time step between t_{j-1} and t_j that the angular velocity and non-gravitational acceleration are constant over the small time step. This leads to the following equations:

$$\mathbf{v}_j = \mathbf{a}_j \Delta t_j \tag{6.8}$$

$$\boldsymbol{\theta}_j = \boldsymbol{\omega}_j \Delta t_j \,, \tag{6.9}$$

where

$$\mathbf{a}_j = \mathbf{a}_{j-1} \quad \text{and} \quad \boldsymbol{\omega}_j = \boldsymbol{\omega}_{j-1} \,.$$

6.3.1.1 Attitude Propagation

Recall the quaternion multiplicative error and applying it to represent the difference between a quaternion at time t_j and time t_{j-1} ,

$$\Delta\bar{\mathbf{q}} = \hat{\bar{\mathbf{q}}}_j^- \otimes [\hat{\bar{\mathbf{q}}}_{j-1}]^{-1} \,.$$

Re-arranging for $\hat{\bar{\mathbf{q}}}_j^-$,

$$\hat{\bar{\mathbf{q}}}_j^- = \Delta\bar{\mathbf{q}} \otimes \hat{\bar{\mathbf{q}}}_{j-1} \,.$$

$\Delta\bar{\mathbf{q}}$ is the small angle quaternion that propagates the attitude quaternion from time t_{j-1} to time t_j. The small angle is the angular displacement during the small time step Δt_j from the angular velocity of the spacecraft. Therefore, with the small angle assumption applied due to the small time step,

$$\Delta\bar{\mathbf{q}} = \bar{\mathbf{q}}(\hat{\boldsymbol{\theta}}_j) \,,$$

where

$$\hat{\boldsymbol{\theta}}_j = \hat{\boldsymbol{\omega}}_j \Delta t_j \,.$$

Therefore, the *a priori* attitude quaternion estimate is

$$\hat{\bar{\mathbf{q}}}_j^- = \bar{\mathbf{q}}(\hat{\boldsymbol{\theta}}_j) \otimes \hat{\bar{\mathbf{q}}}_{j-1} \, , \tag{6.10}$$

where, based on Eq. 5.26,

$$\bar{\mathbf{q}}(\hat{\boldsymbol{\theta}}_j) = \begin{bmatrix} \frac{\hat{\boldsymbol{\theta}}_j}{||\hat{\boldsymbol{\theta}}_j||} \sin \frac{||\hat{\boldsymbol{\theta}}_j||}{2} \\ \cos \frac{||\hat{\boldsymbol{\theta}}_j||}{2} \end{bmatrix} \, .$$

6.3.1.2 Position and Velocity Propagation

From Eq. 4.6, the velocity differential equation is given by

$$\dot{\mathbf{v}}(t) = \mathbf{g} + \mathbf{T}^T \mathbf{a} \, , \tag{6.11}$$

where $\mathbf{g}$ is the gravitational acceleration, $\mathbf{T}^T$ is the case to inertial transformation matrix $\mathbf{T}_b^i(\bar{\mathbf{q}}_b^i)\mathbf{T}_c^b$, and $\mathbf{a}$ is the non-gravitational acceleration. Integrating Eq. 6.11 from t_0 to t ,

$$\begin{aligned} \int_{t_0}^{t} \dot{\mathbf{v}}(t) \, dt &= \int_{t_0}^{t} [\mathbf{g} + \mathbf{T}^T \mathbf{a}] \, dt \\ \mathbf{v}(t) - \mathbf{v}(t_0) &= \mathbf{g}[t - t_0] + \mathbf{T}^T \mathbf{a}[t - t_0] \, . \end{aligned}$$

The velocity in continuous time is

$$\mathbf{v}(t) = \mathbf{g}[t - t_0] + \mathbf{T}^T \mathbf{a}[t - t_0] + \mathbf{v}(t_0) \, . \tag{6.12}$$

Integrating Eq. 6.12, where $\mathbf{v}(t) = \dot{\mathbf{r}}(t)$,

$$\begin{aligned} \int_{t_0}^{t} \dot{\mathbf{r}}(t) \, dt &= \int_{t_0}^{t} \Big[\mathbf{g}[t - t_0] + \mathbf{T}^T \mathbf{a}[t - t_0] + \mathbf{v}(t_0) \Big] \, dt \\ \mathbf{r}(t) - \mathbf{r}(t_0) &= \frac{1}{2}\mathbf{g}[t - t_0]^2 + \frac{1}{2}\mathbf{T}^T \mathbf{a}[t - t_0]^2 + \mathbf{v}(t_0)[t - t_0] \, . \end{aligned}$$

The position in continuous time over a small time step is

$$\mathbf{r}(t) = \frac{1}{2}\mathbf{g}[t - t_0]^2 + \frac{1}{2}\mathbf{T}^T\mathbf{a}[t - t_0]^2 + \mathbf{v}(t_0)[t - t_0] + \mathbf{r}(t_0)\ . \tag{6.13}$$

Eqs. 6.13 and 6.12 are formed on the assumption that the behavior of the spacecraft is linear. Since this is only valid over a small time step Δt_j where $t = t_j$ and $t_0 = t_{j-1}$, the discrete *a priori* position and velocity estimates are

$$\mathbf{r}_j = \frac{1}{2}\mathbf{g}\Delta t_j^2 + \frac{1}{2}\mathbf{T}^T\mathbf{a}\Delta t_j^2 + \mathbf{v}_{j-1}\Delta t_k + \mathbf{r}_{j-1} \tag{6.14}$$

$$\mathbf{v}_j = \mathbf{g}\Delta t_j + \mathbf{T}^T\mathbf{a}\Delta t_j + \mathbf{v}_{j-1}\ . \tag{6.15}$$

6.3.1.3 *Bias and Error Term Propagation*

Since the error terms are modeled as constants,

$$\begin{aligned}
\mathbf{m}_j &= \mathbf{m}_{j-1} \\
\mathbf{n}_j &= \mathbf{n}_{j-1} \\
\mathbf{s}_j &= \mathbf{s}_{j-1} \\
\boldsymbol{\beta}_j &= \boldsymbol{\beta}_{j-1}\ .
\end{aligned}$$

6.3.2 Covariance Propagation

For the MEKF, the state estimation error covariance differential equation is

$$\dot{\mathbf{P}}(t) = \mathbf{F}(\hat{\mathbf{x}}, t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(\hat{\mathbf{x}}, t) + \mathbf{Q}\ , \tag{6.16}$$

where we propagate the covariance between updates according to Eq. 3.16 using numerical integration with $\mathbf{P}(t_{k-1}) = P_{k-1}^+$.

6.4 Update

The update process for the MEKF differs from that of a typical extended Kalman filter. Due to the unit constraint of attitude quaternions that invalidates addition and subtraction, a special procedure is required for the attitude quaternion update. This procedure, in some sense, is separate from the rest of the state elements but is still implemented in the same equations as the other state elements.

6.4.1 Position and Velocity Residual

The state residuals for position and velocity are given by Eq. 3.8 ,

$$\mathbf{r}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k \ ,$$

where

$$\hat{\mathbf{y}}_k = \mathbf{h}_k(\hat{\mathbf{x}}_k^-) \ .$$

$\mathbf{y}_k$ is the measured output from the sensor and $\hat{\mathbf{y}}_k$ is the estimated output of the sensor. However, for the case of the quaternion measurement, this is invalid since quaternions cannot be subtracted. In order to compute the residual of the estimated quaternion, the quaternion product is be used.

6.4.2 Quaternion Residual

From Eq. 5.27, the estimated quaternion measurement is given by

$$\hat{\bar{\mathbf{q}}}_i^{tr} = \hat{\bar{\mathbf{q}}}_{\beta,\eta}^{tr} \otimes \bar{\mathbf{q}}_b^{tr} \otimes \hat{\bar{\mathbf{q}}}_i^{b} \ .$$

The quaternion residual of the sensor quaternion measurement and the estimated quaternion measurement is

$$\delta\bar{\mathbf{q}}_i^{tr} = \bar{\mathbf{q}}_i^{tr} \otimes [\hat{\bar{\mathbf{q}}}_i^{tr}]^{-1} \ .$$

When computing the residual for the quaternion, the result is a small angle quaternion of the form

$$\delta \bar{\mathbf{q}}_i^{tr} = \begin{bmatrix} \frac{1}{2}\mathbf{r}_k \\ 1 \end{bmatrix} ,$$

where $\mathbf{r}_k$ is the quaternion measurement residual vector that is used in Eq. 3.8. Therefore, for the attitude quaternion, the measurement residual is

$$\mathbf{r}_k = 2\delta \mathbf{q}_i^{tr} \ . \tag{6.17}$$

6.5 Position and Velocity Update

The state update equation for position, velocity and all the error terms are given by Eq. 3.10,

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{r}_k \ .$$

6.5.1 Quaternion Update

For updating the quaternion using Eq. 3.10, the same issue arises where addition or subtraction is invalid. The *a priori* quaternion estimate cannot be added to the quaternion residual, otherwise the the addition produces a quaternion with $||\bar{\mathbf{q}}|| > 1$. For this reason, $\delta\boldsymbol{\alpha}$ is estimated, rather than the quaternion itself. This can be done because it is assumed that the error between the true and estimated quaternion is small. However, $\delta\boldsymbol{\alpha}$ is not propagated like the quaternion is. Applying Eq. 3.10 to $\delta\boldsymbol{\alpha}$ yields

$$\delta\hat{\boldsymbol{\alpha}}_k^+ = \delta\hat{\boldsymbol{\alpha}}_k^- + \mathbf{K}_k \mathbf{r}_k \ .$$

$\delta\hat{\boldsymbol{\alpha}}_k^-$ is the predicted or *a priori* deviation or error of the predicted quaternion $\hat{\bar{\mathbf{q}}}_k^-$. However, the estimation error cannot be predicted before the update because that is the purpose of the update. Therefore, for the quaternion small angle deviation $\delta\boldsymbol{\alpha}$,

$$\delta\hat{\boldsymbol{\alpha}}_k^- = 0\,,$$

and the estimated quaternion error is

$$\delta\hat{\boldsymbol{\alpha}}_k^+ = \mathbf{K}_k\mathbf{r}_k\,. \tag{6.18}$$

The attitude quaternion is updated through

$$\hat{\mathbf{q}}_k^+ = \begin{bmatrix} \frac{1}{2}\delta\hat{\boldsymbol{\alpha}}_k^+ \\ 1 \end{bmatrix} \otimes \hat{\mathbf{q}}_k^-\,. \tag{6.19}$$

6.5.2 Covariance Update

The state estimation error covariance update remains the same and is given in Eq. 3.11,

$$\hat{\mathbf{P}}_k^+ = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k(\hat{\mathbf{x}}_k^-)]\hat{\mathbf{P}}_k^-\,.$$

Note that for the quaternion, the covariance being estimated is actually the covariance of $\delta\hat{\boldsymbol{\alpha}}$. This corresponds to the estimation error in each axis of the attitude estimate.

6.6 Stochastic Modeling of Error Terms

The stochastic models for the IMU systematic errors, sensor errors, and process noise are presented. From Eq. 3.15, sensor noise is modeled as a white noise sequence. Therefore, for our attitude, position, and velocity sensor noise terms we have

$$E\{\boldsymbol{\eta}_q^{tr}\} = \mathbf{0}\,, \qquad E\{\boldsymbol{\eta}_q^{tr} \cdot \boldsymbol{\eta}_q^{tr}\} = \mathbf{R}_q\,,$$

$$E\{\boldsymbol{\eta}^i_{gps/i}\} = \mathbf{0}\,, \qquad E\{\boldsymbol{\eta}^i_{gps/i} \cdot \boldsymbol{\eta}^i_{gps/i}\} = \mathbf{R}_r\,,$$

$$E\{\boldsymbol{\eta}^i_{gps}\} = \mathbf{0}\,, \qquad E\{\boldsymbol{\eta}^i_{gps} \cdot \boldsymbol{\eta}^i_{gps}\} = \mathbf{R}_v\,.$$

From Eq. 3.14, the process noise is modeled as a white noise sequence. For our model, uncertainty comes from accelerometer and gyroscope noise. Therefore, for the velocity and attitude process noise terms we have

$$E\{\boldsymbol{\eta}_a\} = \mathbf{0}\,, \qquad E\{\boldsymbol{\eta}_a \cdot \boldsymbol{\eta}_a\} = \mathbf{Q}_a\delta(t-\tau)\,,$$

$$E\{\boldsymbol{\eta}_g\} = \mathbf{0}\,, \qquad E\{\boldsymbol{\eta}_g \cdot \boldsymbol{\eta}_g\} = \mathbf{Q}_g\delta(t-\tau)\,.$$

The IMU systematic errors are modeled as random constants, as well as the GPS and attitude biases. Therefore, for the accelerometer error terms we have

$$E\{\mathbf{m}_a\} = \mathbf{0}\,, \qquad E\{\mathbf{m}_a \cdot \mathbf{m}_a\} = \mathbf{P}_{\mathbf{m}_a}\,,$$

$$E\{\mathbf{n}_a\} = \mathbf{0}\,, \qquad E\{\mathbf{n}_a \cdot \mathbf{n}_a\} = \mathbf{P}_{\mathbf{n}_a}\,,$$

$$E\{\mathbf{s}_a\} = \mathbf{0}\,, \qquad E\{\mathbf{s}_a \cdot \mathbf{s}_a\} = \mathbf{P}_{\mathbf{s}_a}\,,$$

$$E\{\boldsymbol{\beta}_a\} = \mathbf{0}\,, \qquad E\{\boldsymbol{\beta}_a \cdot \boldsymbol{\beta}_a\} = \mathbf{P}_{\boldsymbol{\beta}_a}\,.$$

For the gyroscope error terms we have

$$E\{\mathbf{m}_g\} = \mathbf{0}\,, \qquad E\{\mathbf{m}_g \cdot \mathbf{m}_g\} = \mathbf{P}_{\mathbf{m}_g}\,,$$

$$E\{\mathbf{n}_g\} = \mathbf{0}\,, \qquad E\{\mathbf{n}_g \cdot \mathbf{n}_g\} = \mathbf{P}_{\mathbf{n}_g}\,,$$

$$E\{\mathbf{s}_g\} = \mathbf{0}\,, \qquad E\{\mathbf{s}_g \cdot \mathbf{s}_g\} = \mathbf{P}_{\mathbf{s}_g}\,,$$

$$E\{\boldsymbol{\beta}_g\} = \mathbf{0}\,, \qquad E\{\boldsymbol{\beta}_g \cdot \boldsymbol{\beta}_g\} = \mathbf{P}_{\boldsymbol{\beta}_g}\,.$$

For the GPS position, GPS velocity, and attitude bias terms we have

$$E\{\boldsymbol{\beta}^{f}_{pos}\} = \mathbf{0}\,, \qquad E\{\boldsymbol{\beta}^{f}_{pos} \cdot \boldsymbol{\beta}^{f}_{pos}\} = \mathbf{P}_{\boldsymbol{\beta}^{f}_{pos}}\,,$$

$$E\{\boldsymbol{\beta}^{f}_{vel}\} = \mathbf{0}\,, \qquad E\{\boldsymbol{\beta}^{f}_{vel} \cdot \boldsymbol{\beta}^{f}_{vel}\} = \mathbf{P}_{\boldsymbol{\beta}^{f}_{vel}}\,,$$

$$E\{\boldsymbol{\beta}^{tr}_{q}\} = \mathbf{0}\,, \qquad E\{\boldsymbol{\beta}^{tr}_{q} \cdot \boldsymbol{\beta}^{tr}_{q}\} = \mathbf{P}_{\boldsymbol{\beta}^{tr}_{q}}\,.$$

Chapter 7: Kalman Filter Tuning and Results

The MEKF results are presented here. First we evaluate the results from a single run. This provides a glimpse of how well the MEKF is estimating the state vector and state estimation error covariance. It will also highlight the unobservable states, if any. The unobservable states are indicated by non-converging state estimation error covariances. Furthermore, we employ more analyses tools to properly scrutinize the MEKF performance and ensure that the filter is performing in a realistic manner.

Following a single simulation is the Monte-Carlo analysis of the Kalman filter. This involves simulating multiple runs and taking the average of the state error history and sample error variance. The average state error and square root of variance over hundreds of runs are plotted with a variance plot from a single run. The sample error variance and the square root of estimate variance should closely match for a well-tuned filter. In mathematical terms, we want to confirm that

$$\frac{\sum_{n=1}^{m}(\mathbf{x}_n - \hat{\mathbf{x}}_n^-)(\mathbf{x}_n - \hat{\mathbf{x}}_n^-)^T}{m-1} \approx \mathbf{P}_n^- \quad \forall \quad n$$

and

$$\frac{\sum_{n=1}^{m}(\mathbf{x}_n - \hat{\mathbf{x}}_n^+)(\mathbf{x}_n - \hat{\mathbf{x}}_n^+)^T}{m-1} \approx \mathbf{P}_n^+ \quad \forall \quad n\,,$$

where m is the sample size. This is confirmation that the filter is working properly and that the variance estimate is accurately representing the true error bounds, i.e., the results represent reality.

The next analysis is the error budget. This is a detailed investigation of each error contribution to the state estimation error. Through the error budget we can see which sources of uncertainty (such as noise, bias, or systematic errors) contribute the most to the overall uncertainty of the position, velocity, and attitude estimates. If certain error sources are found to contribute a negligible amount of error to the 9 main state elements, then they may be considered for removal from our system model to further reduce complexity while achieving similar estimation accuracy.

Consideration of error group removal is dependent on a sensitivity analysis which follows from the error budget and examines the effect of increasing or decreasing the scale of certain error groups on the overall estimation accuracy. This is useful in case an error group is much larger than expected. If the state estimate accuracy is affected by scaling a group, then the error group is sensitive. If scaling does not affect the state estimation error then the error group is negligible. Following from the sensitivity analysis is the sub-optimal filter design. If any error groups are found to have insignificant effects on the overall uncertainty from the error budget and sensitivity analysis then we may remove them from the model. However, removing information from the model will affect the uncertainty in the model. To make up for the reduced amount of information in the model we may increase the process noise matrix and tune the filter.

7.1 Simulation Trajectory

The attitude trajectory used in this simulation captures the behavior of a cubesat deployed in a 28.5° orbit. The cubesat will experience an initial angular rate of around $10^\circ/s$ caused by the spring-loaded deployment. The onboard controller will stabilize the rate of the spacecraft and proceed with sun-normal solar panel pointing. This simulation covers the time it takes to de-tumble (estimated at 30 seconds) and 170 seconds of stable flying.

7.2 Single Run Results

The filter uncertainty parameters are given in Table 7.1 and are defined for all runs. These parameters reflect performance of the typical off-the-shelf low-cost MEMS sensors that are commercially available and similar to the sensors utilized aboard our satellite [16,17].

Table 7.1: Kalman Filter Uncertainty Parameters

	Variable	**Description**	**Value(STD)**
Initial covariance	$\mathbf{P_r}$	Position uncertainty	$4(m)$
	$\mathbf{P_v}$	Velocity uncertainty	$0.11(m/s)$
	$\mathbf{P_q}$	Attitude uncertainty	$0.225(rad)$
	$\mathbf{P}_{\beta_r}$	GPS position bias uncertainty	$1(m)$
	$\mathbf{P}_{\beta_v}$	GPS velocity bias uncertainty	$0.01(m/s)$
	$\mathbf{P}_{\beta_q}$	Derived quaternion bias uncertainty	$0.05(rad)$
	$\mathbf{P}_{\mathbf{m}_a}$	IMU accelerometer mislaignment uncertainty	$0.0032(m/s^2)$
	$\mathbf{P}_{\mathbf{n}_a}$	IMU accelerometer non-orthogonality uncertainty	$0.0032(m/s^2)$
	$\mathbf{P}_{\mathbf{s}_a}$	IMU accelerometer scale factor uncertainty	$0.0032(m/s^2)$
	$\mathbf{P}_{\beta_a}$	IMU accelerometer bias uncertainty	$0.08(m/s^2)$
	$\mathbf{P}_{\mathbf{m}_g}$	IMU gyroscope mislaignment uncertainty	$0.0032(rad/s)$
	$\mathbf{P}_{\mathbf{n}_g}$	IMU gyroscope non-orthogonality uncertainty	$0.0032(rad/s)$
	$\mathbf{P}_{\mathbf{s}_g}$	IMU gyroscope scale factor uncertainty	$0.0032(rad/s)$
	$\mathbf{P}_{\beta_g}$	IMU gyroscope bias uncertainty	$0.08(rad/s)$
Measurement Noise	$\mathbf{R_r}$	GPS position	$3(m)$
	$\mathbf{R_v}$	GPS velocity	$0.1(m/s)$
	$\mathbf{R_q}$	Derived quaternion	$0.175(rad)$
Process Noise	$\mathbf{Q}_a\delta(t-\tau)$	IMU accelerometer	$0.0175(m/s^2)$
	$\mathbf{Q}_g\delta(t-\tau)$	IMU gyroscope	$0.0035(rad/s)$

Figures 7.1 - 7.29 depict single run results for all states. For attitude, comparison plots between the true, measured, and estimated quaternion elements are presented. For position, velocity, and attitude, the estimation error and state estimation error covariance are presented. For the rest of the state elements, estimation error and state estimation error covariance plots are presented.

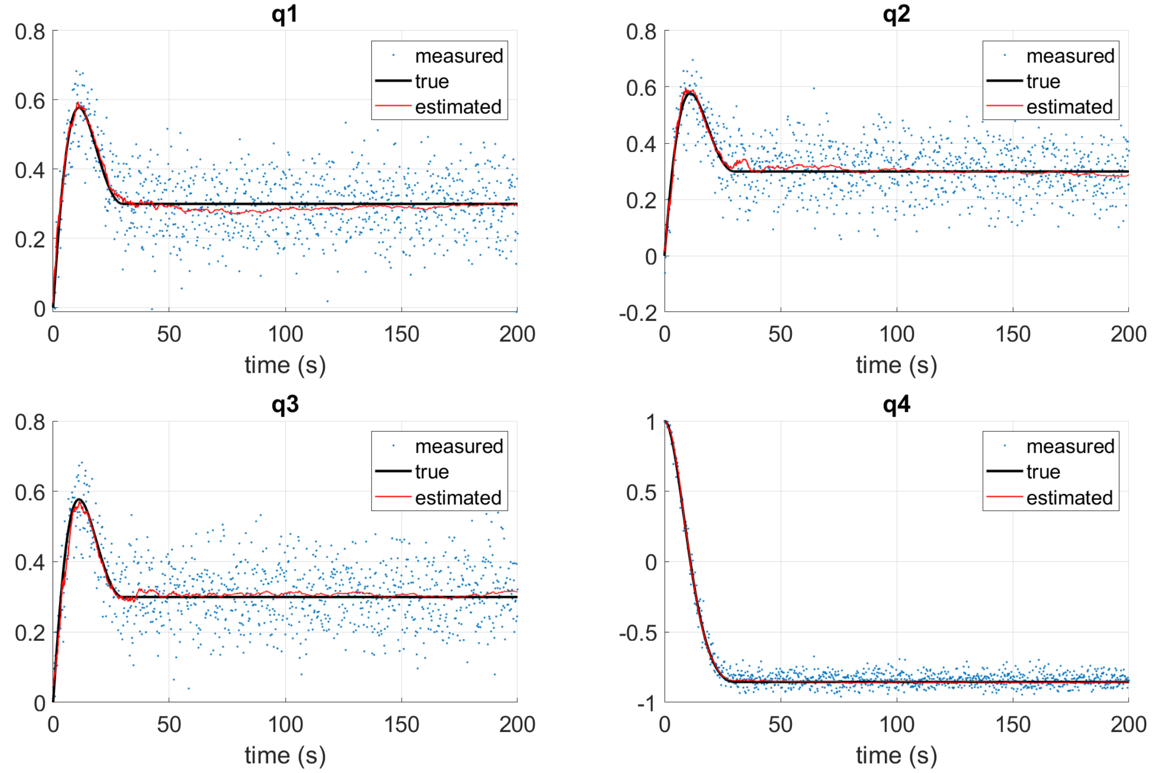


Figure 7.1: Quaternion Vector Elements

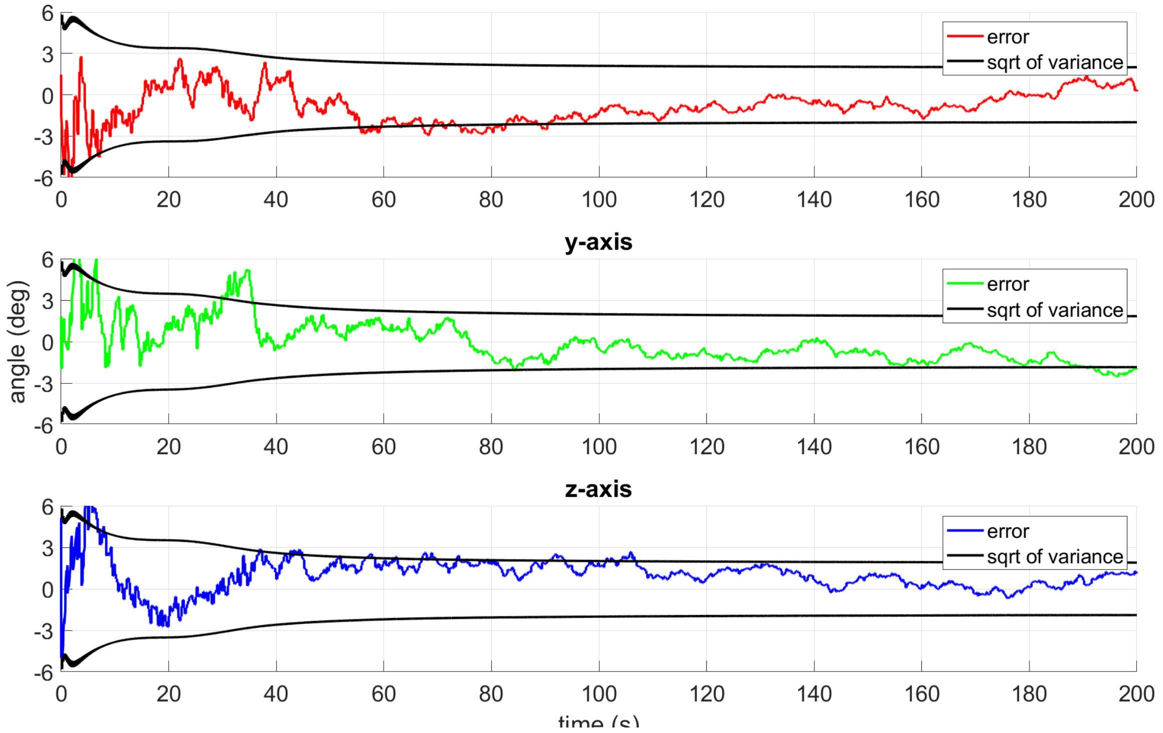


Figure 7.2: Attitude Error and Covariance

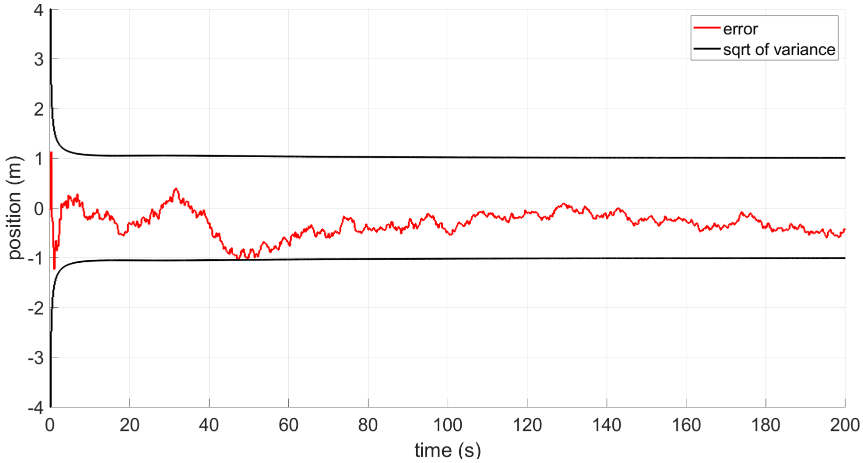


Figure 7.3: GPS Position Error and Error Covariance (x-axis)

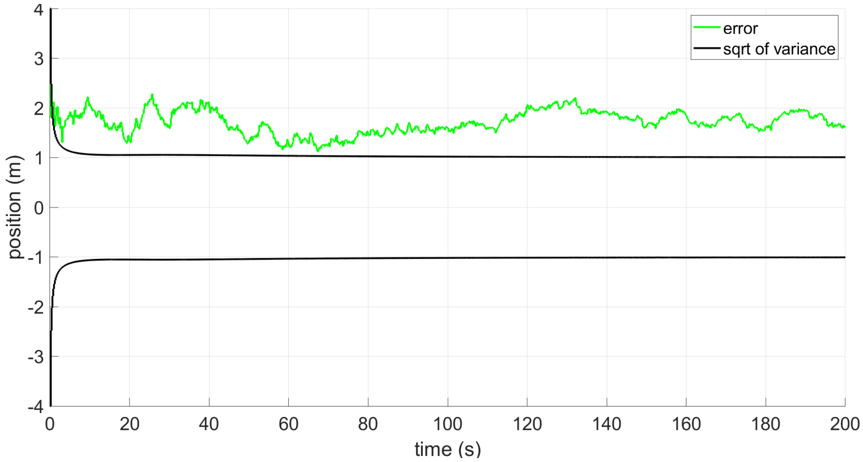


Figure 7.4: GPS Position Error and Error Covariance (y-axis)

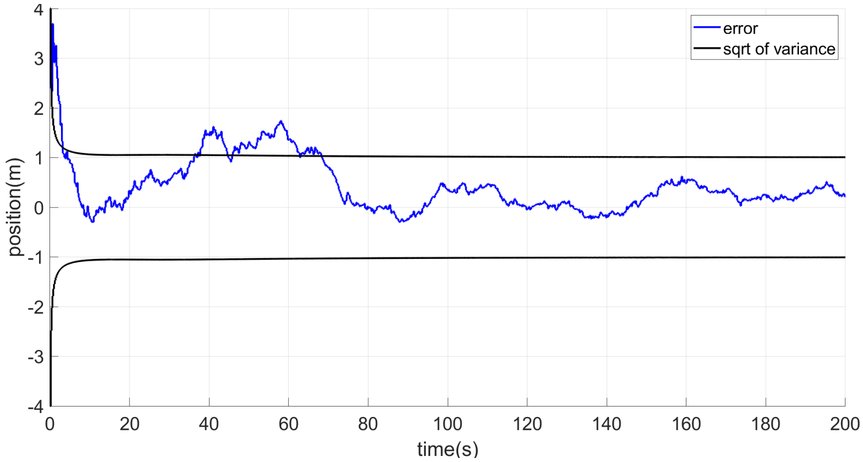


Figure 7.5: GPS Position Error and Error Covariance (z-axis)

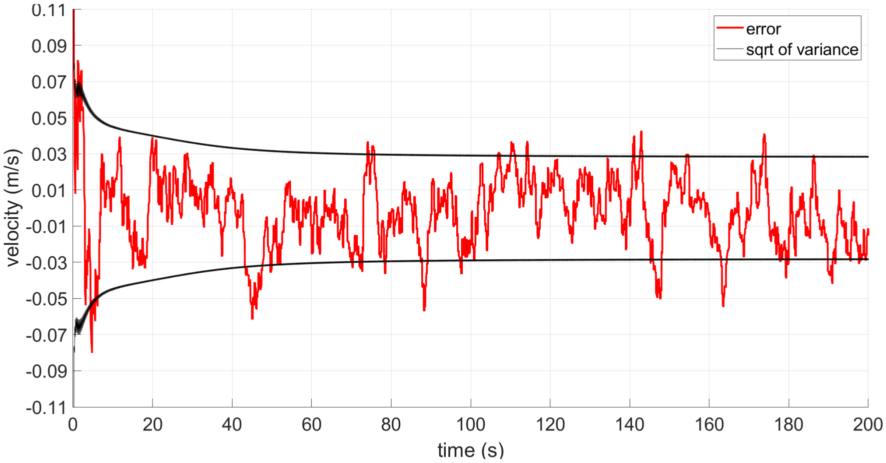


Figure 7.6: GPS Velocity Error and Error Covariance (x-axis)

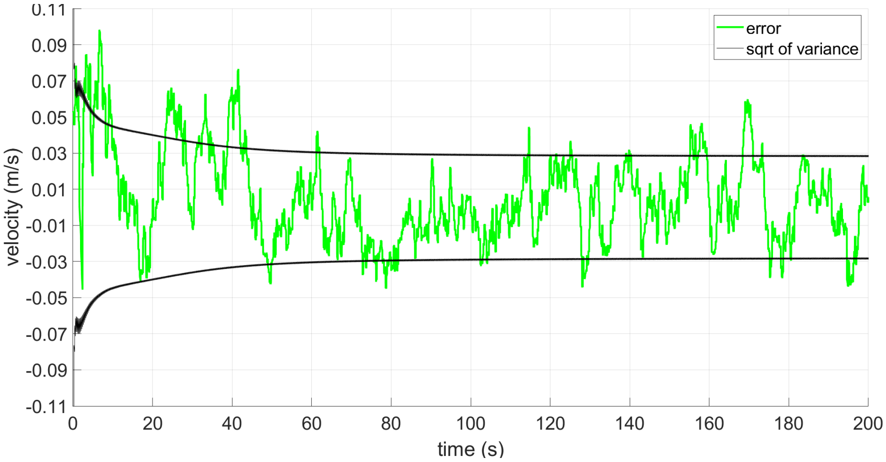


Figure 7.7: GPS Velocity Error and Error Covariance (y-axis)

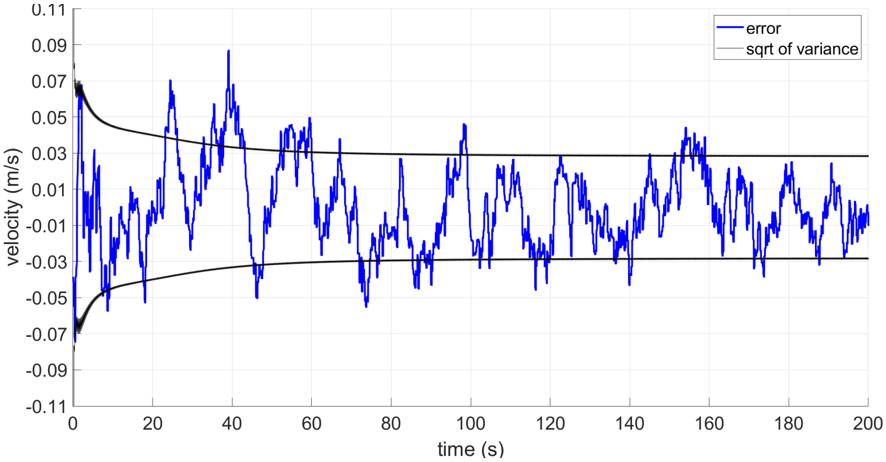


Figure 7.8: GPS Velocity Error and Error Covariance (z-axis)

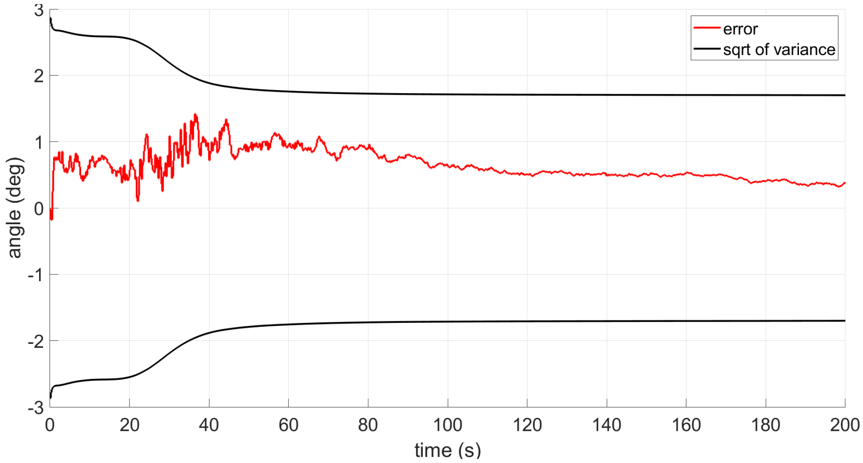


Figure 7.9: Quaternion Bias Error and Error Covariance (x-axis)

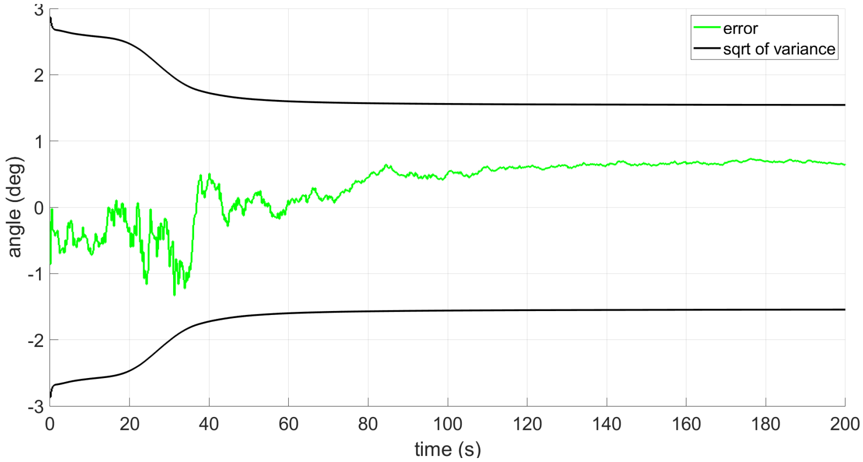


Figure 7.10: Quaternion Bias Error and Error Covariance (y-axis)

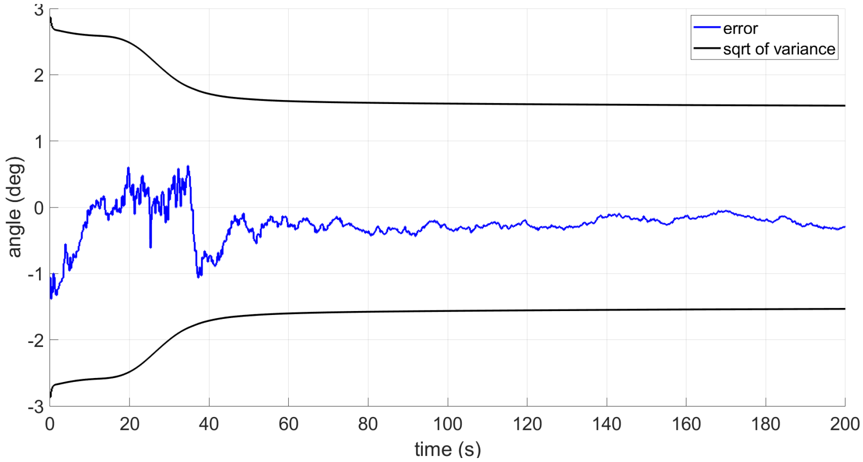


Figure 7.11: Quaternion Bias Error and Error Covariance (z-axis)

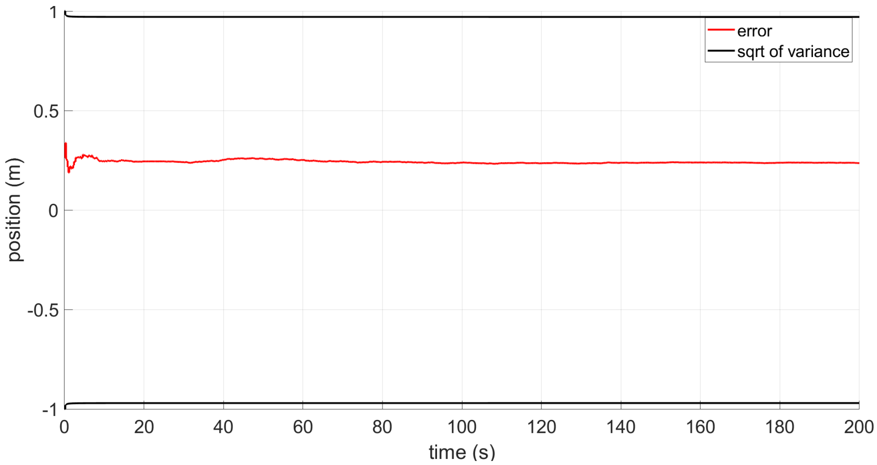


Figure 7.12: GPS Position Bias Error and Error Covariance (x-axis)

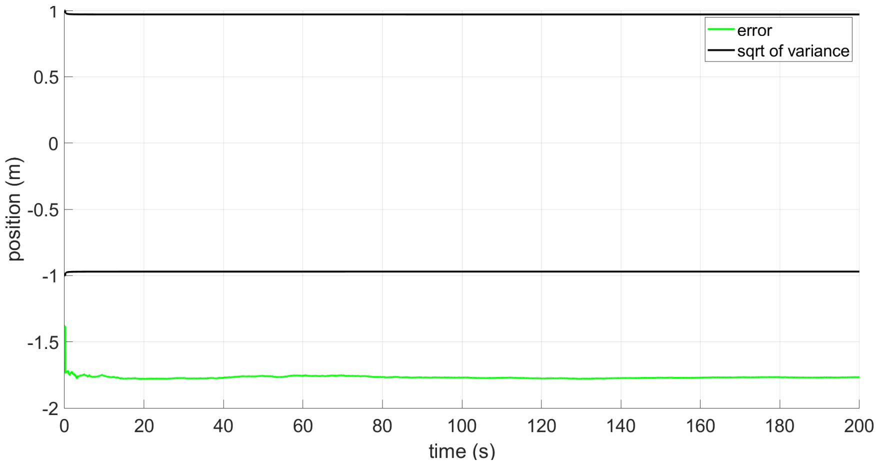


Figure 7.13: GPS Position Bias Error and Error Covariance (y-axis)

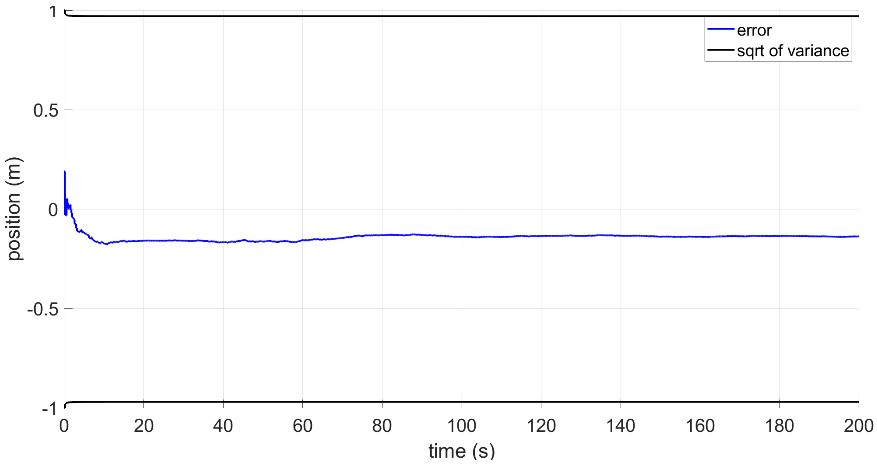


Figure 7.14: GPS Position Bias Error and Error Covariance (z-axis)

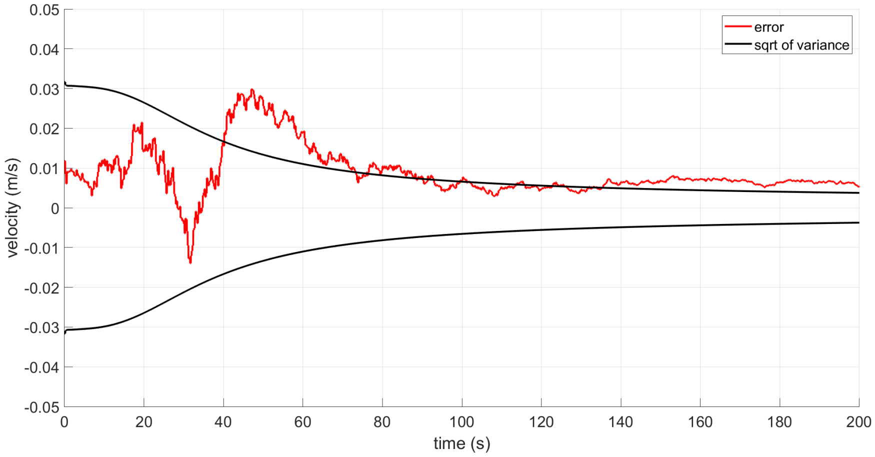


Figure 7.15: GPS Velocity Bias Error and Error Covariance (x-axis)

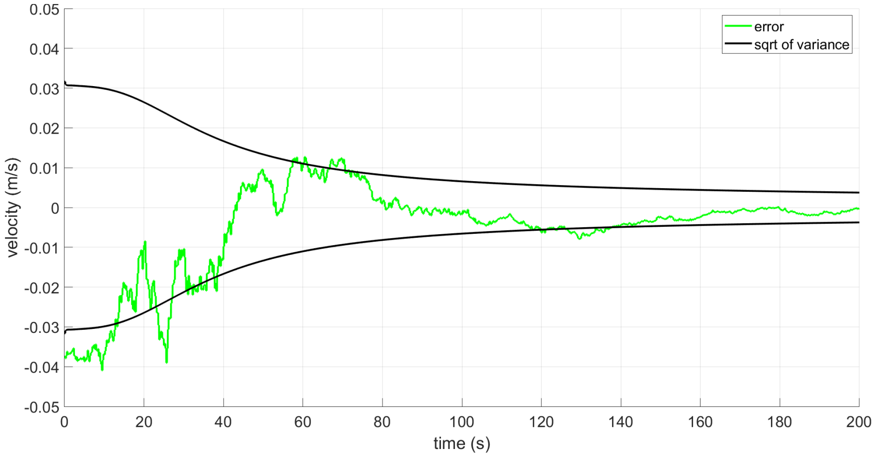


Figure 7.16: GPS Velocity Bias Error and Error Covariance (y-axis)

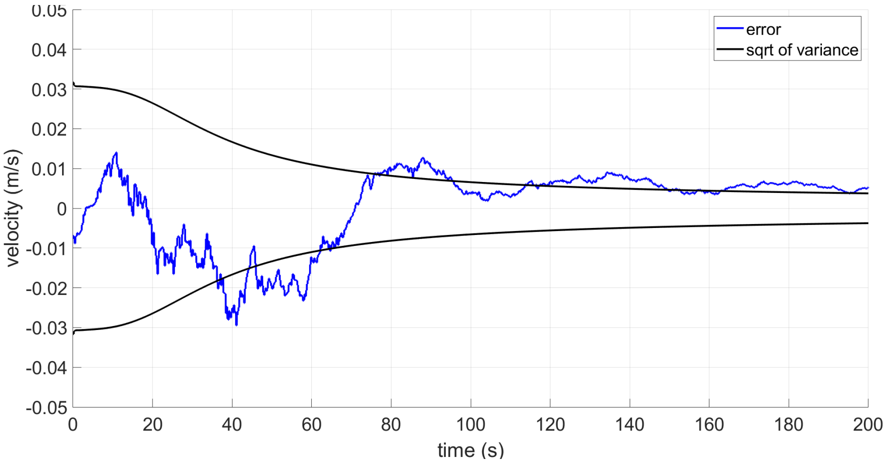


Figure 7.17: GPS Velocity Bias Error and Error Covariance (z-axis)

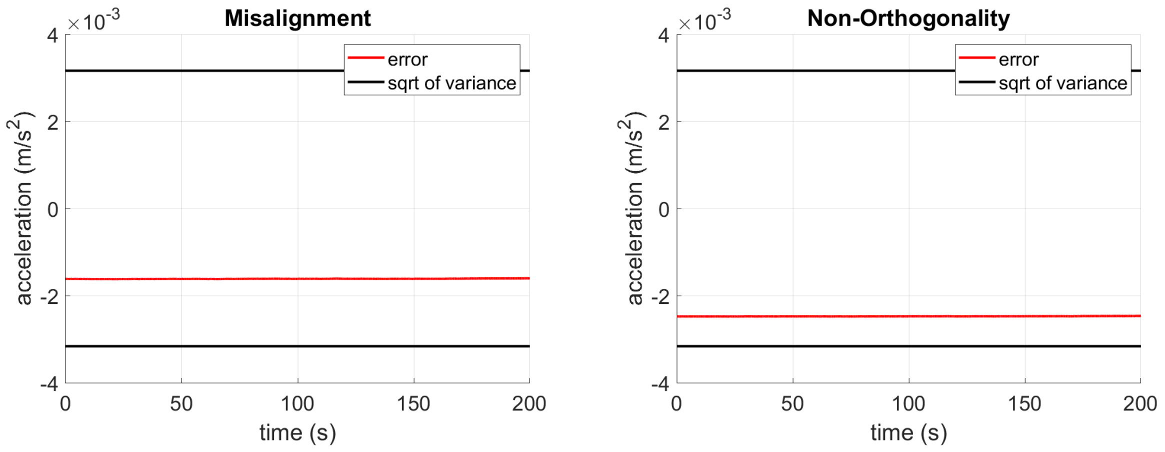


Figure 7.18: Accelerometer Misalignment and Non-Orthogonality (x-axis)

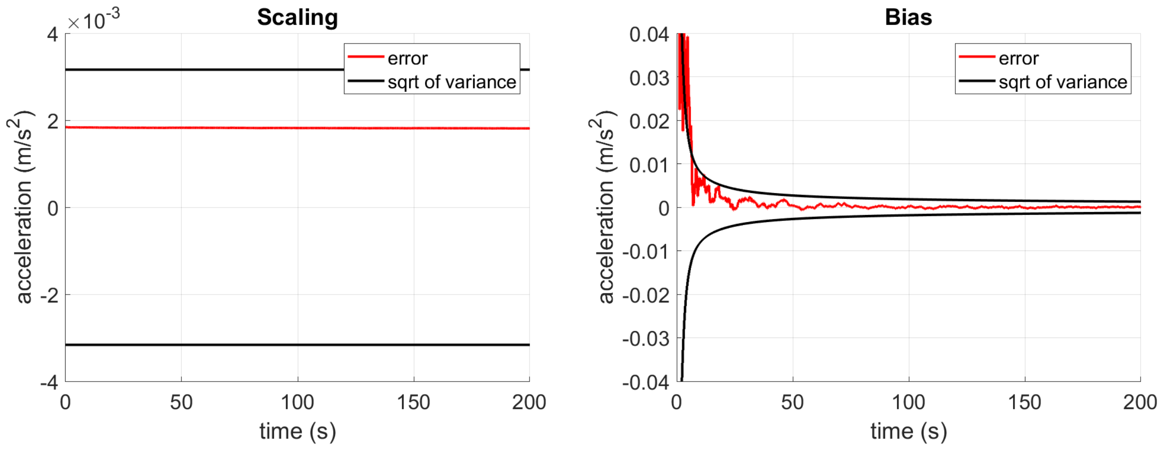


Figure 7.19: Accelerometer Scale Factor and Bias (x-axis)

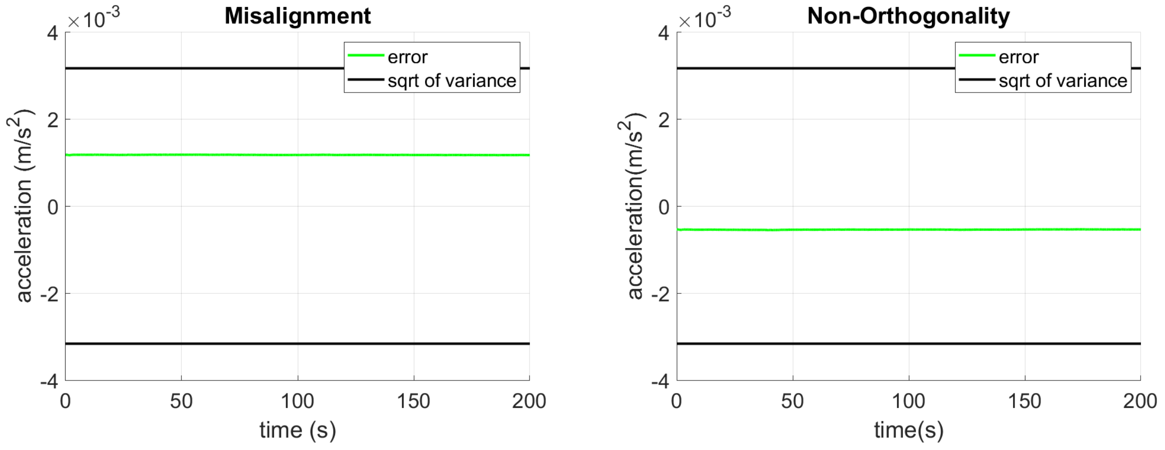


Figure 7.20: Accelerometer Misalignment and Non-Orthogonality (y-axis)

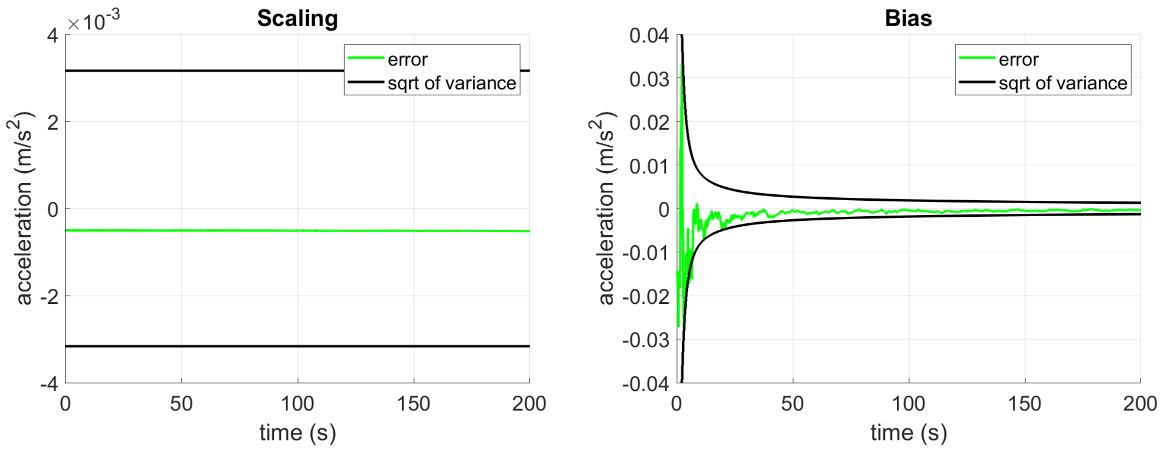


Figure 7.21: Accelerometer Scale Factor and Bias (y-axis)

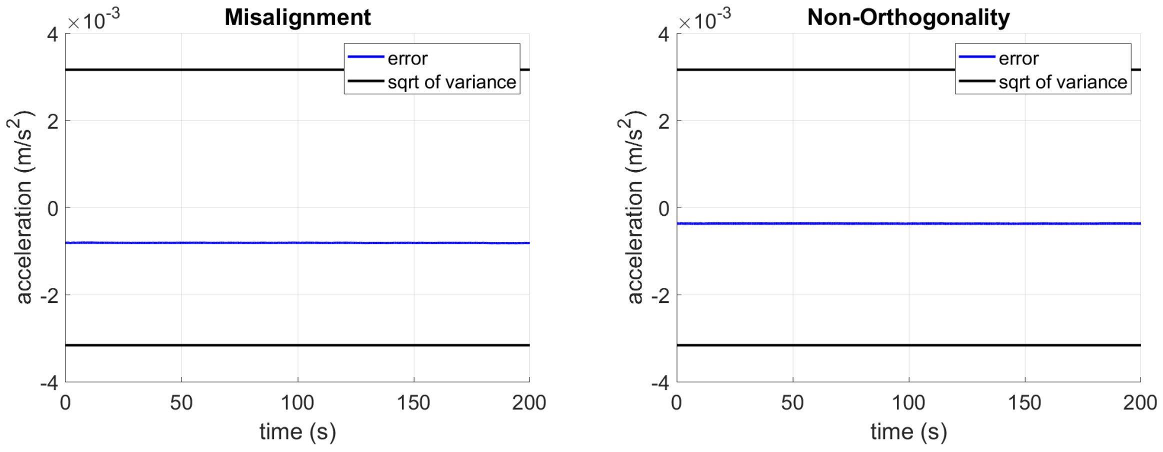


Figure 7.22: Accelerometer Misalignment and Non-Orthogonality (z-axis)

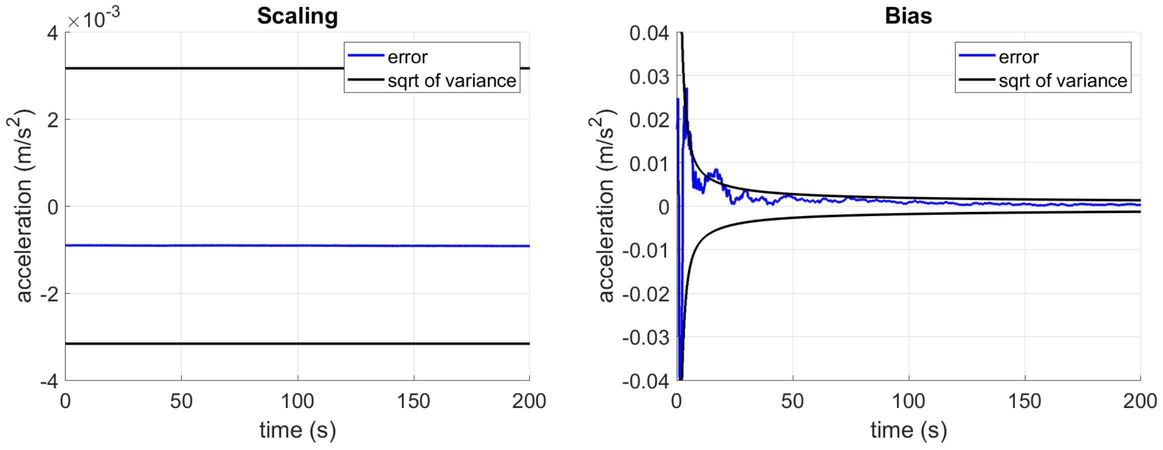


Figure 7.23: Accelerometer Scale Factor and Bias (z-axis)

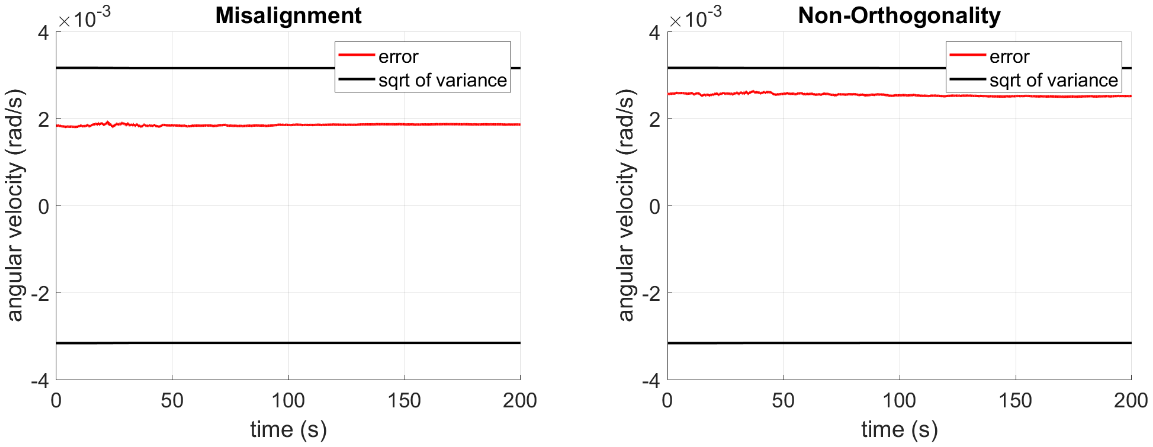


Figure 7.24: Gyroscope Misalignment and Non-Orthogonality (x-axis)

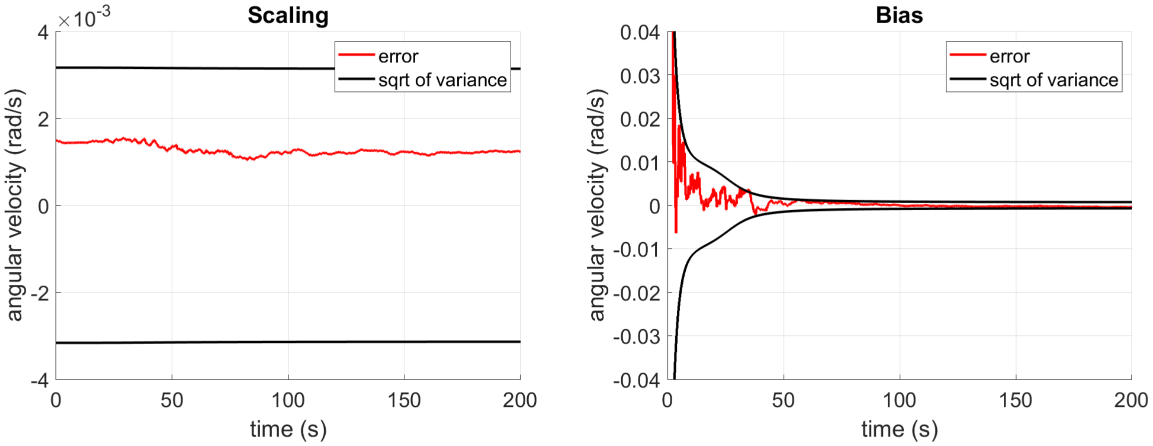


Figure 7.25: Gyroscope Scale Factor and Bias (x-axis)

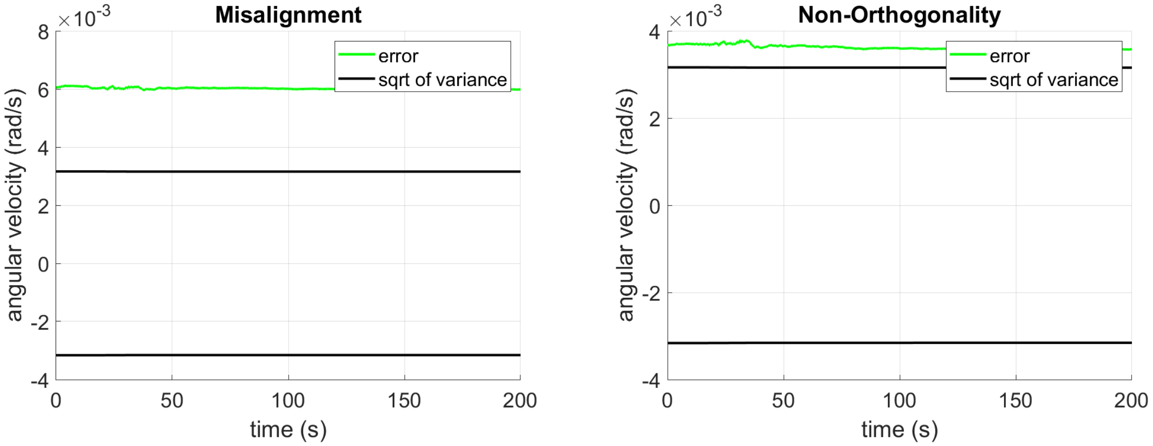


Figure 7.26: Gyroscope Misalignment and Non-Orthogonality (y-axis)

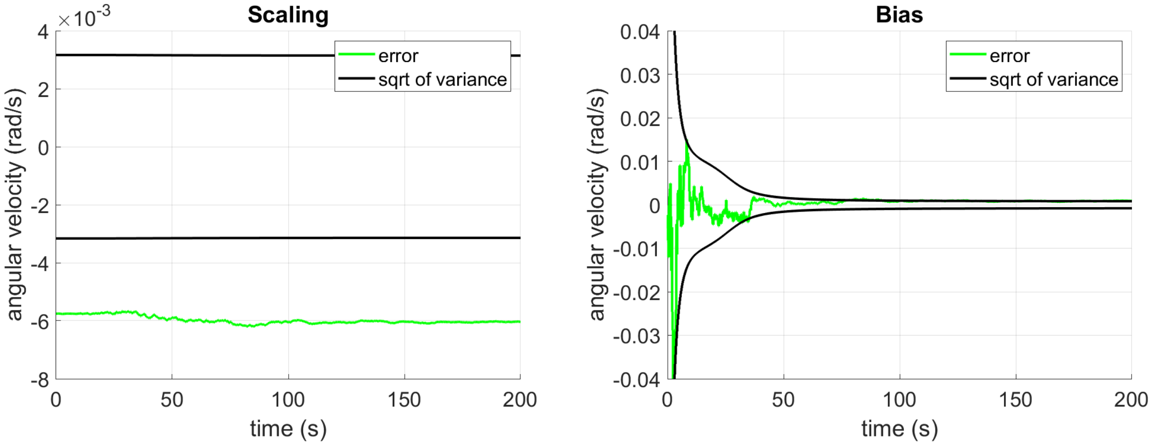


Figure 7.27: Gyroscope Scale Factor and Bias (y-axis)

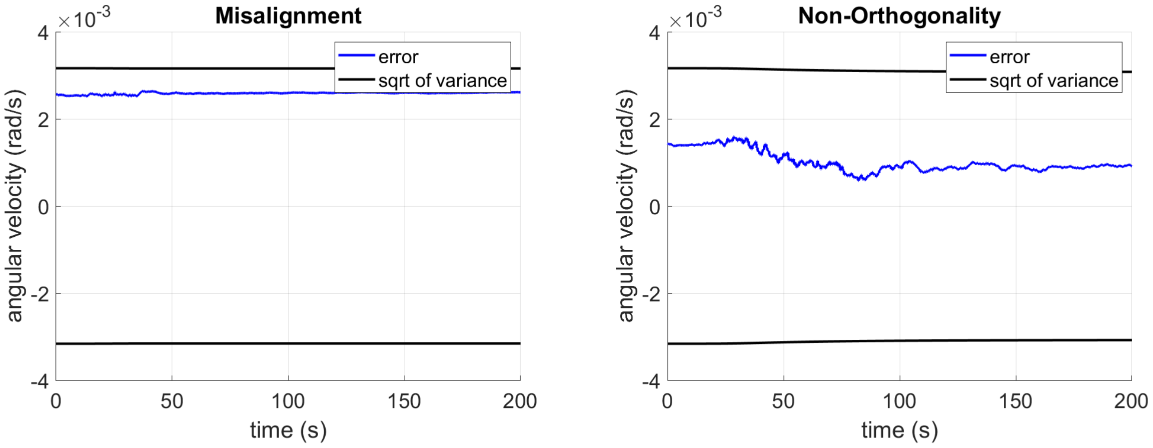


Figure 7.28: Gyroscope Misalignment and Non-Orthogonality (z-axis)

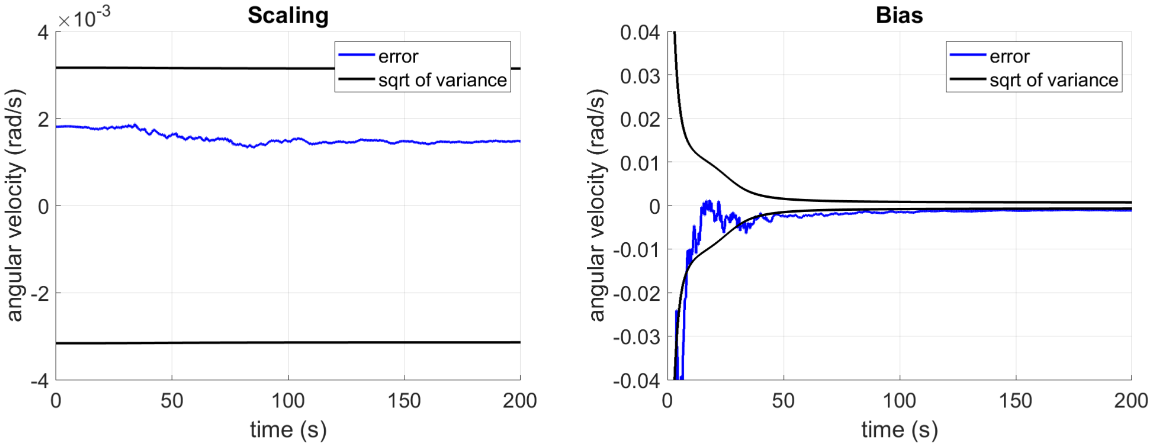


Figure 7.29: Gyroscope Scale Factor and Bias (z-axis)

7.3 Monte-Carlo Results

The Monte-Carlo results are depicted in Figures 7.30 - 7.43 for 500 runs.

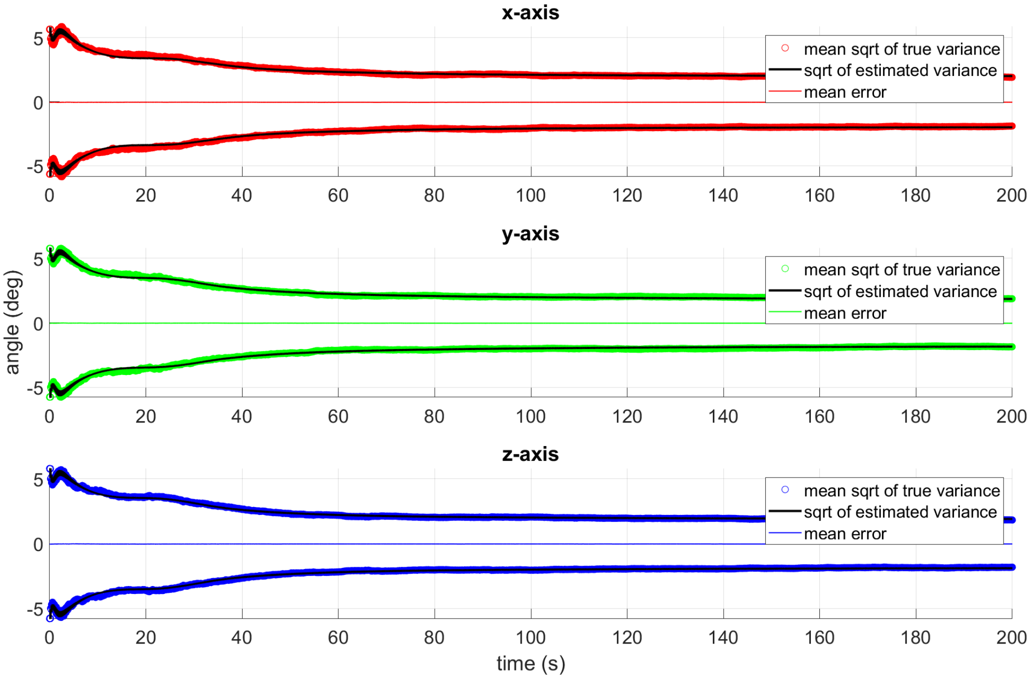


Figure 7.30: Monte-Carlo Attitude

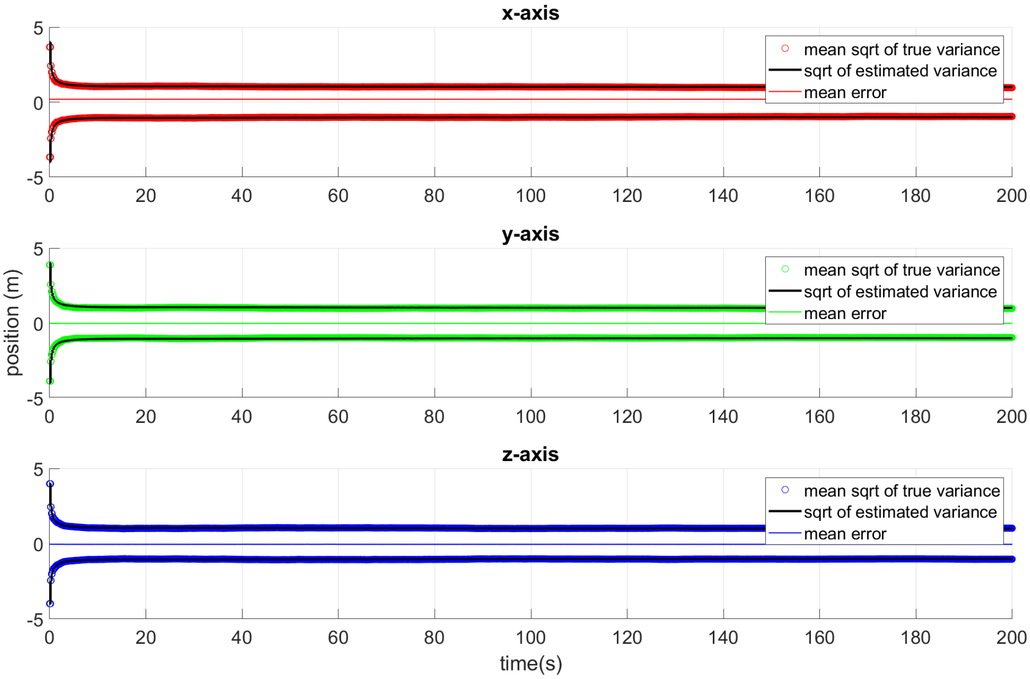


Figure 7.31: Monte-Carlo Position

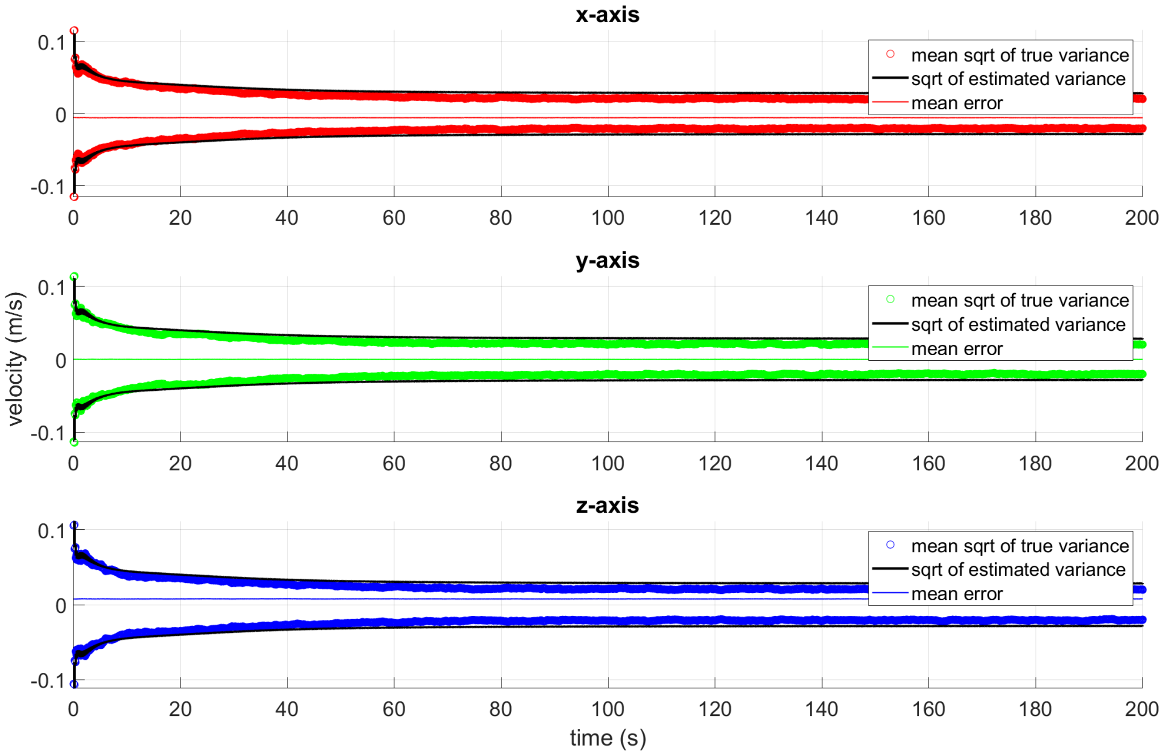


Figure 7.32: Monte-Carlo Velocity

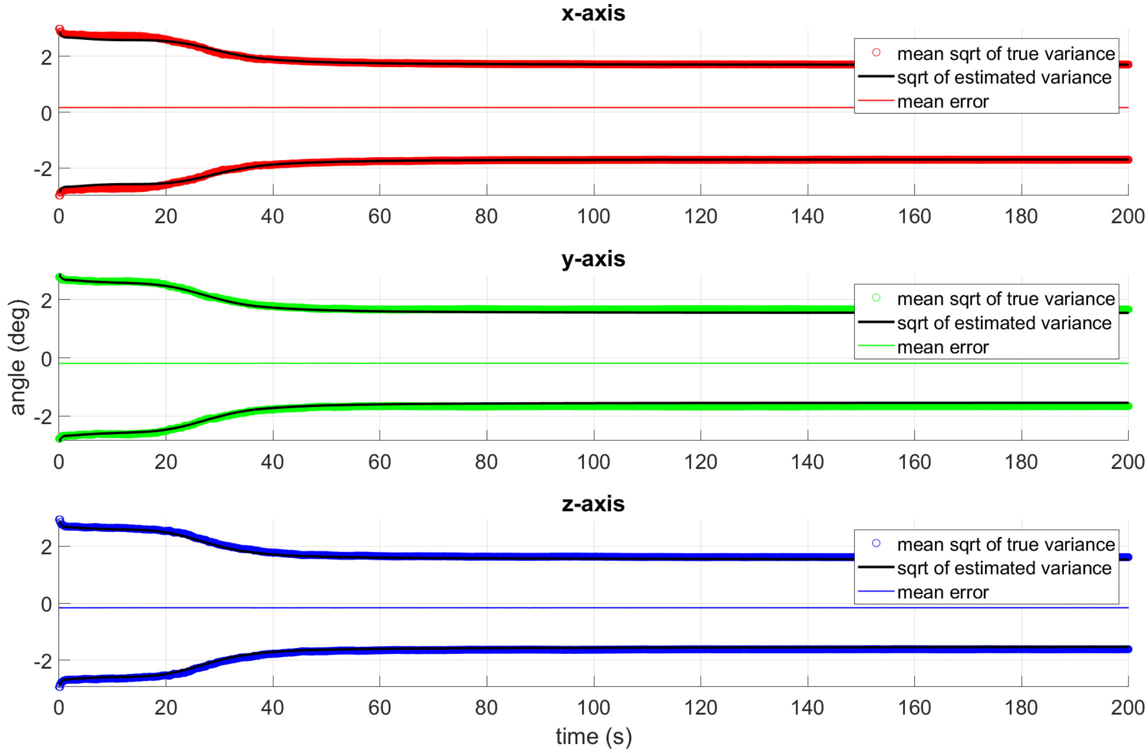


Figure 7.33: Monte-Carlo Quaternion Bias

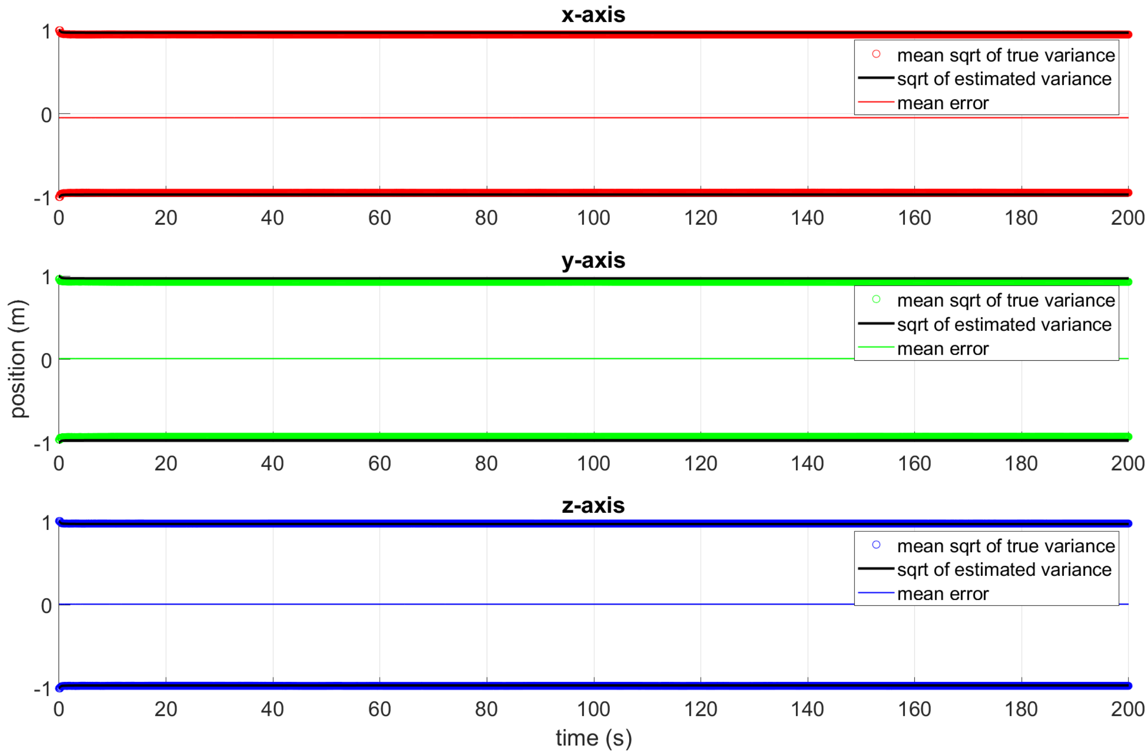


Figure 7.34: Monte-Carlo GPS Position Bias

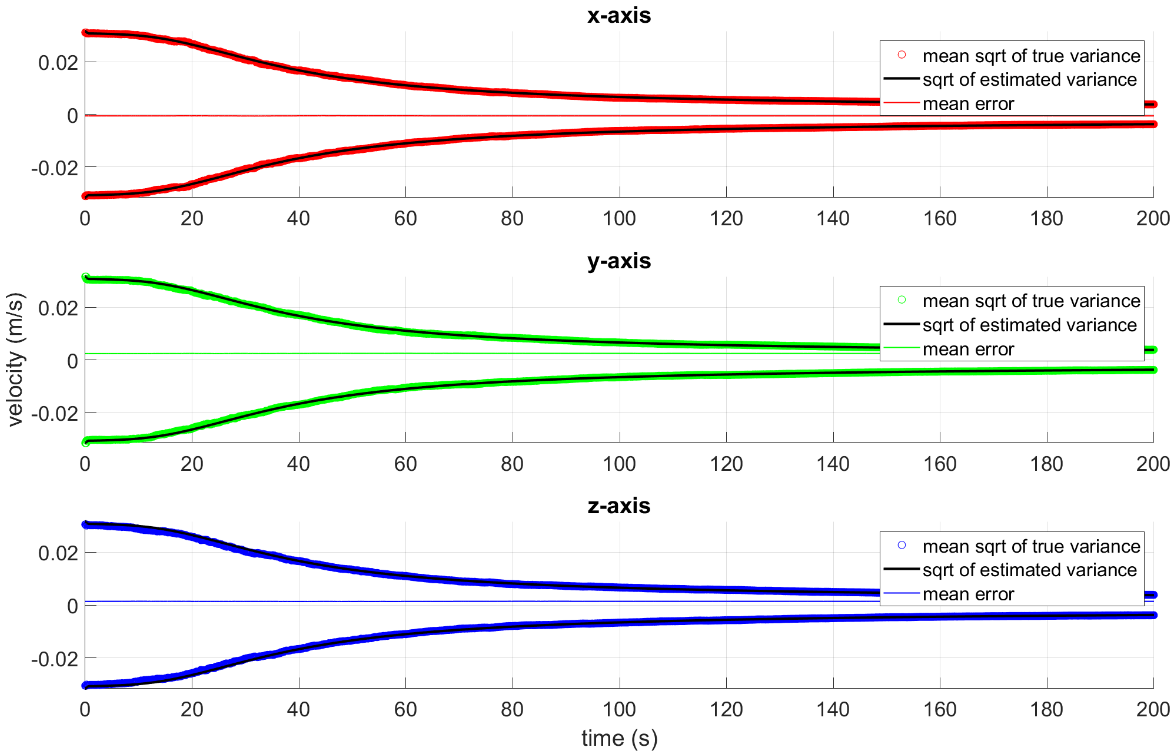


Figure 7.35: Monte-Carlo GPS Velocity Bias

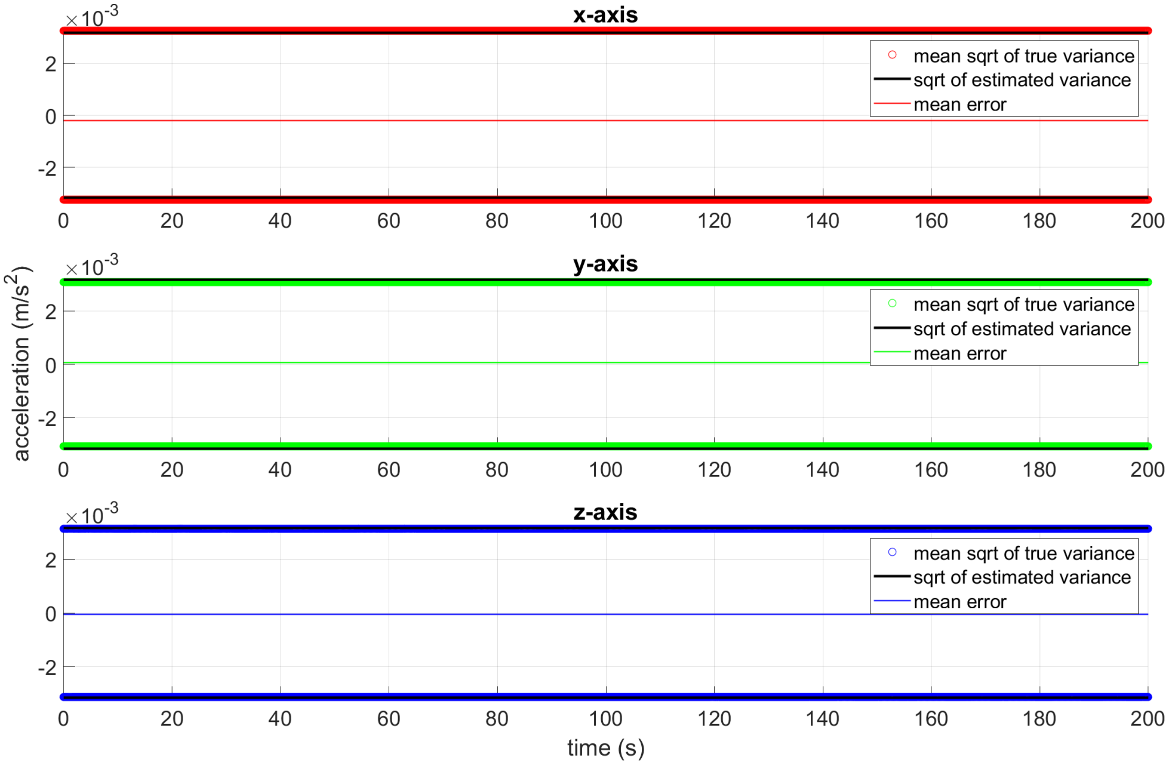


Figure 7.36: Monte-Carlo Accelerometer Misalignment Error

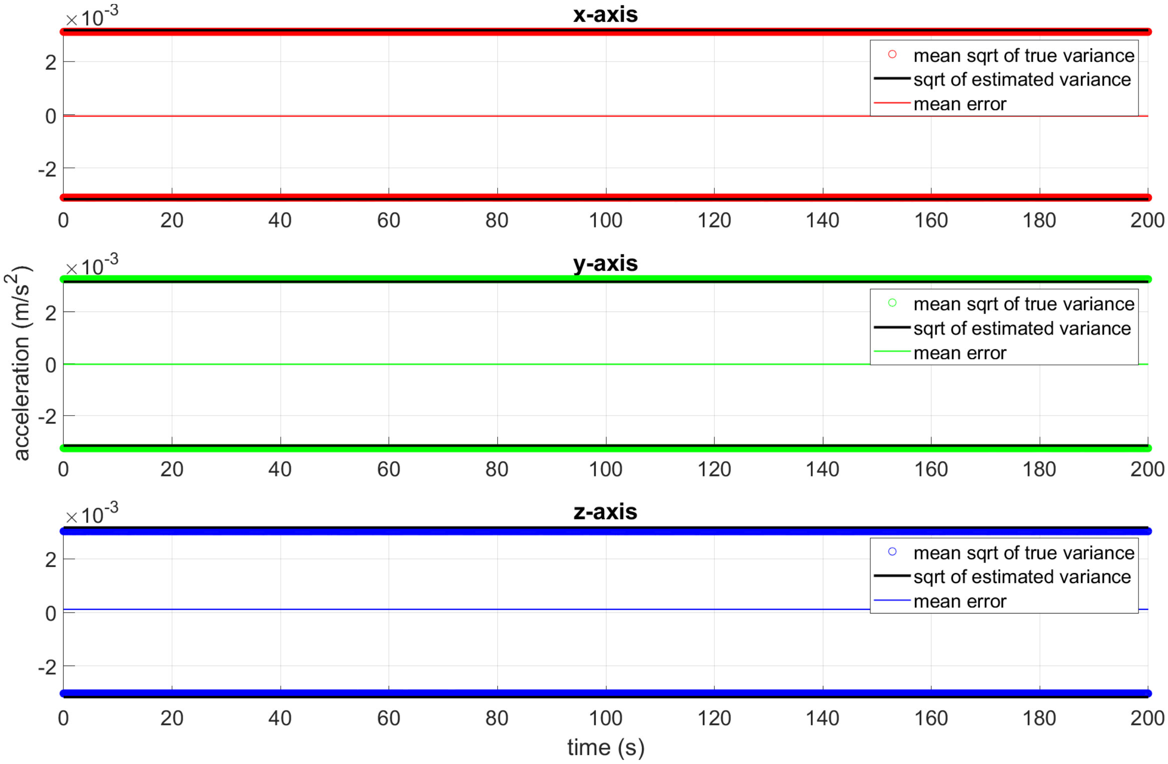


Figure 7.37: Monte-Carlo Accelerometer Non-Orthogonality Error

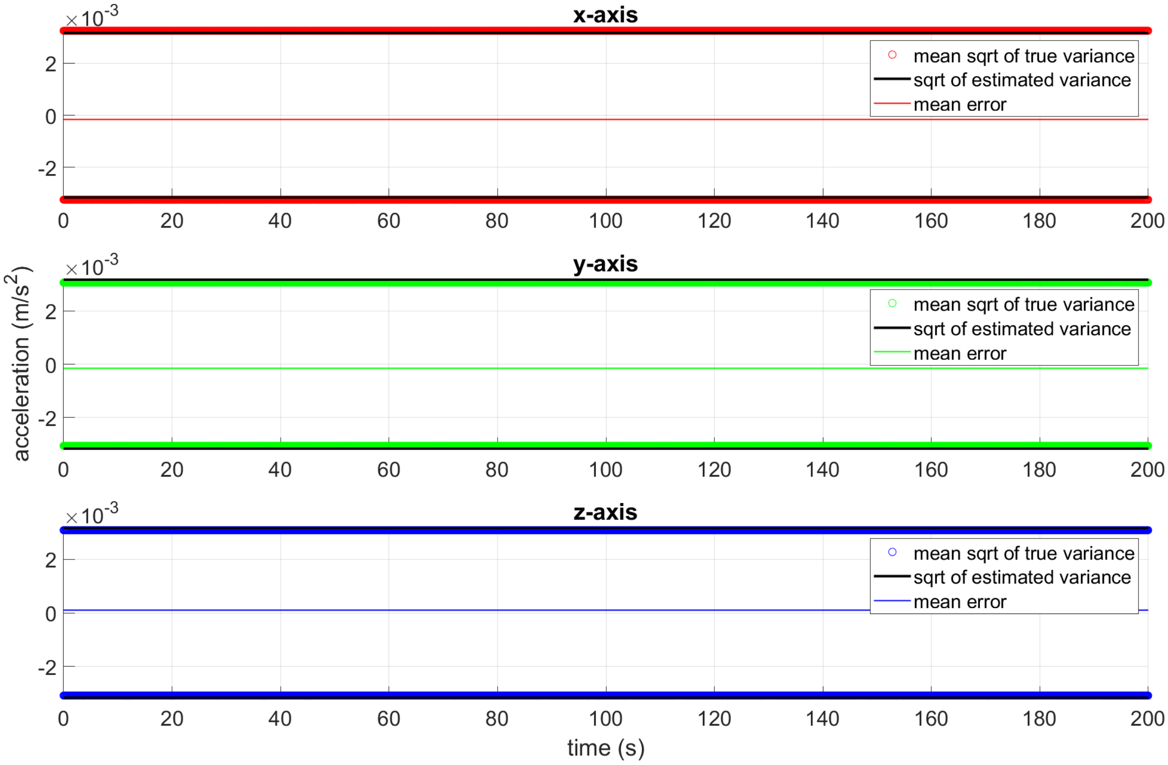


Figure 7.38: Monte-Carlo Accelerometer Scaling Error

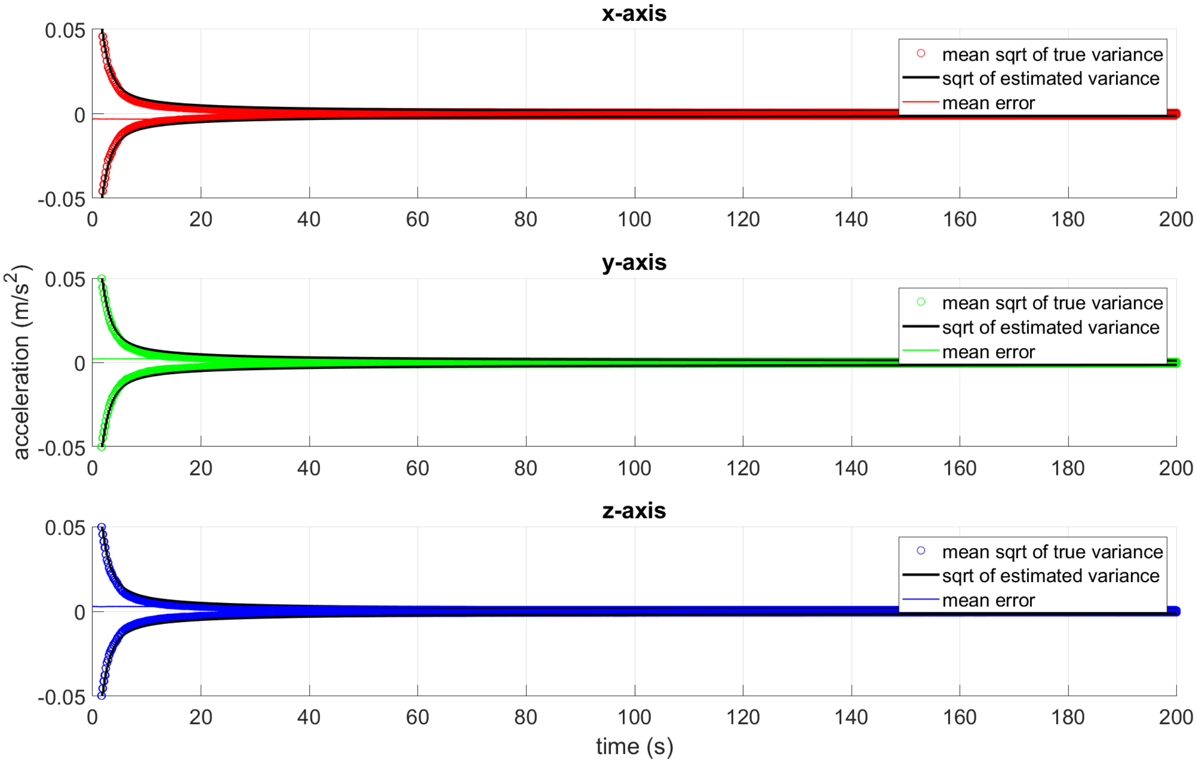


Figure 7.39: Monte-Carlo Accelerometer Bias

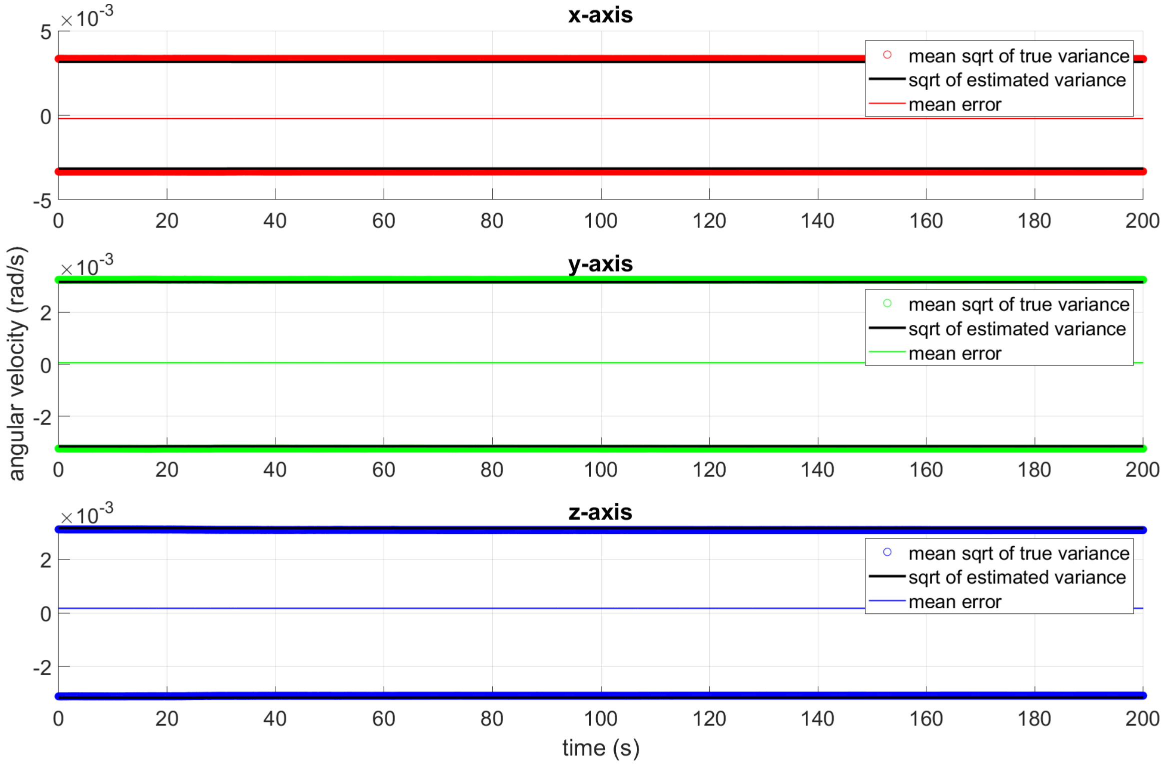


Figure 7.40: Monte-Carlo Gyroscope Misalignment Error

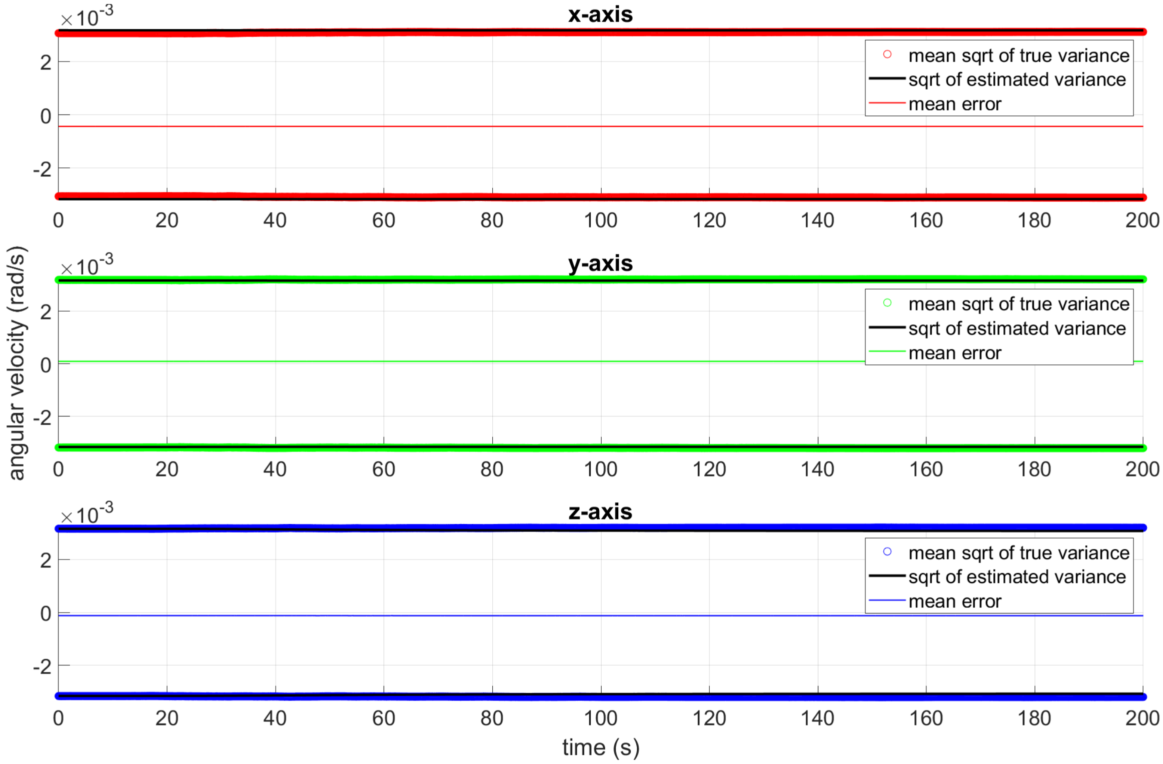


Figure 7.41: Monte-Carlo Gyroscope Non-Orthogonality Error

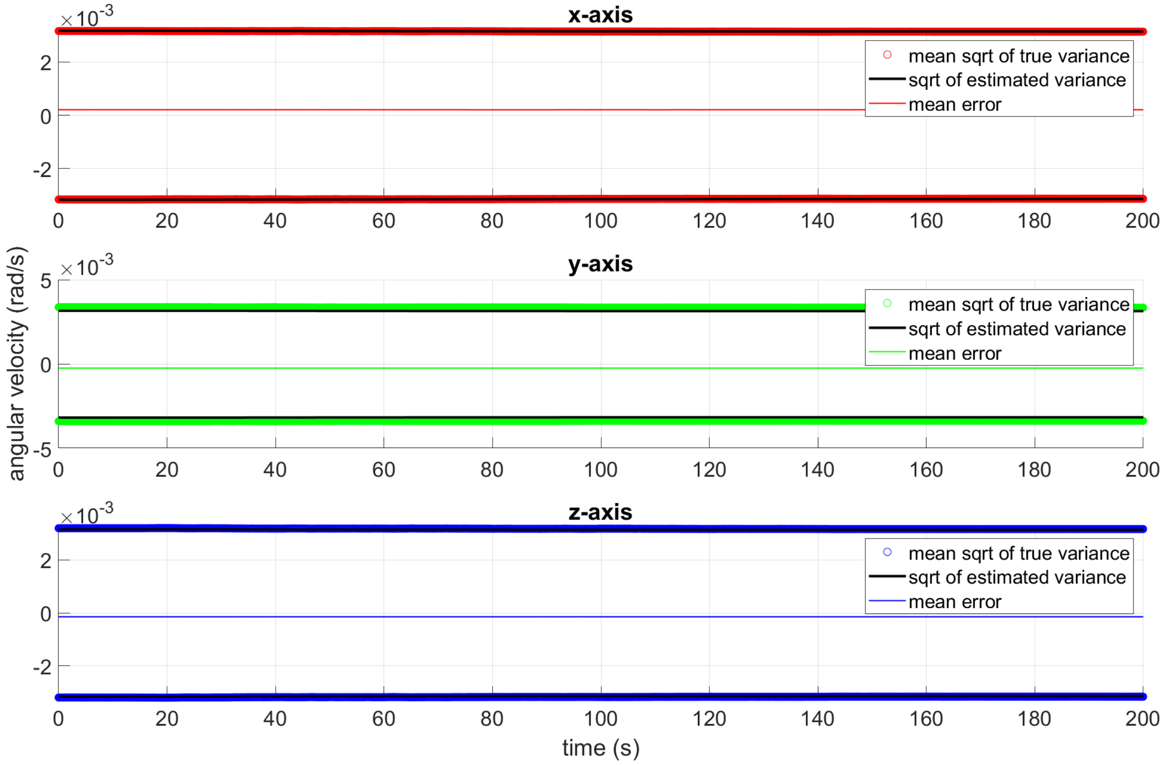


Figure 7.42: Monte-Carlo Gyroscope Scaling Error

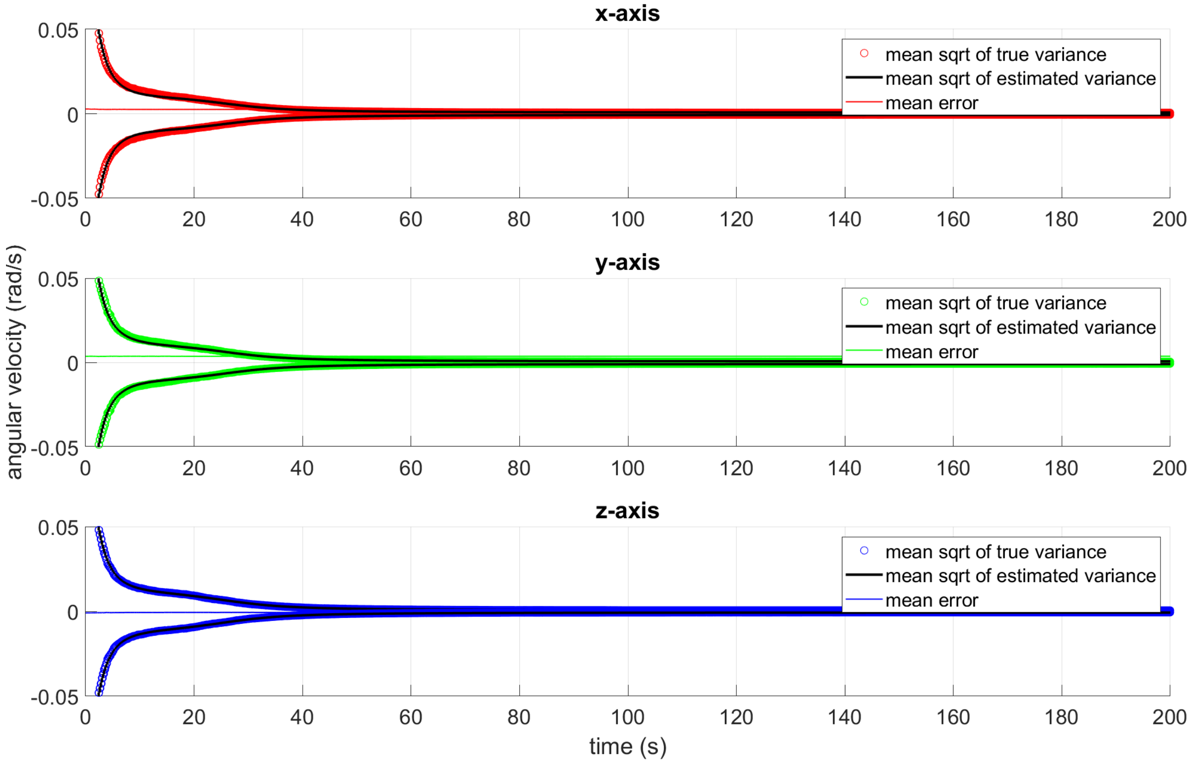


Figure 7.43: Monte-Carlo Gyroscope Bias

7.4 Error Budget

The error budget illuminates sources of error which contribute the most towards the overall uncertainty of the Kalman filter estimation error. The steps for producing the error budget are

1. Run the Kalman filter simulation with all error sources active and store the Kalman gain for every point in time. Also record the square root of variance for position, velocity, and attitude at a single point in time where the variance has converged and reached a steady state value.

2. Using the stored Kalman gain, re-run the Kalman filter simulation but with all error sources turned off except one. Use the stored Kalman gain for the update stage instead of computing the gain again. Also record the steady state square root of variance value at the same point in time as the previous run.

3. Repeat the previous step for all error sources, alternating through each source while all the others are off and storing the steady state square root of variance at the same point in time.

4. Take the RSS of the steady state square root of variance from each error group run (excluding the first run with all sources active) and compare it to the square root of variance of the first run where all error sources were active. Both values should be nearly identical.

The variance from each error group run is the contribution of that error group to the overall uncertainty of the position, velocity, and attitude estimates. If the contribution of a certain group is so small that it does not affect the uncertainty the position, velocity, and attitude, then the terms can be considered for removal from the model to further simplify the equations and reduce computational load. Further justification beyond the direct contribution from a single error budget is required to constitute removal of an error term from the model. This is done in a sensitivity analysis, where the effect of increasing or decreasing the magnitude of the error term uncertainty on the overall estimate accuracy is analyzed. In the real world, these errors may be larger than expected, therefore, we need insight into how the filter will respond to larger than expected errors. The error sources are listed in Table 7.2.

Table 7.2: Error Group Designations

	Group #	**Name**
Initial covariance	1	Position uncertainty
	2	Velocity uncertainty
	3	Attitude uncertainty
	4	GPS position bias uncertainty
	5	GPS velocity bias uncertainty
	6	Derived quaternion bias uncertainty
	7	IMU accelerometer mislaignment uncertainty
	8	IMU accelerometer non-orthogonality uncertainty
	9	IMU accelerometer scale factor uncertainty
	10	IMU accelerometer bias uncertainty
	11	IMU gyroscope mislaignment uncertainty
	12	IMU gyroscope non-orthogonality uncertainty
	13	IMU gyroscope scale factor uncertainty
	14	IMU gyroscope bias uncertainty
Measurement Noise	15	GPS position
	16	GPS velocity
	17	Derived quaternion
Process Noise	18	IMU accelerometer
	19	IMU gyroscope

The error budget is then computed with results shown in Table 7.3. The percent contribution of each group is also computed from Table 7.3 and shown in Table 7.4.

Table 7.3: Error Budget

Error Group	Position(m)			Velocity(m/s)			Attitude(deg)		
1	0.2249	0.2249	0.2249	0.001432	0.001412	0.001432	0.0008236	0.001114	0.0003989
2	0.05667	0.05463	0.05649	0.004136	0.002935	0.004125	0.03077	0.08587	0.0734
3	0.0002273	0.00147	0.0008115	$9.319e-5$	$2.457e-5$	$4.932e-5$	0.7596	0.8927	0.7627
4	0.9438	0.9438	0.9438	0.0003579	0.0003531	0.0003579	0.0002059	0.0002785	$9.972e-5$
5	0.2077	0.2078	0.2077	0.01385	0.01384	0.01384	0.002277	0.01052	0.009635
6	0.001107	0.000466	0.0004178	$1.957e-5$	0.0002381	$2.581e-5$	2.205	2.445	2.145
7	$1.821e-7$	$9.456e-6$	$4.79e-8$	$3.997e-8$	$3.65e-6$	$3.529e-9$	$1.677e-6$	$1.79e-6$	$4.998e-6$
8	$1.821e-7$	$9.456e-6$	$4.79e-8$	$3.997e-8$	$3.65e-6$	$3.529e-9$	$1.677e-6$	$1.79e-6$	$4.998e-6$
9	$2.037e-5$	$8.824e-8$	$2.355e-6$	$1.773e-7$	$3.497e-8$	$4.216e-6$	$4.778e-6$	$2.504e-5$	$6.541e-7$
10	0.004821	0.003695	0.0002371	0.0007167	0.0004286	0.0003683	0.01932	0.0301	0.03675
11	$2.467e-5$	$5.237e-5$	$1.551e-5$	$1.011e-6$	$1.257e-5$	$2.875e-6$	0.06769	0.009986	0.06762
12	$2.472e-5$	$5.24e-5$	$1.557e-5$	$1.065e-6$	$1.256e-5$	$2.943e-6$	0.06764	0.01018	0.06765
13	$3.368e-5$	$1.004e-5$	$4.529e-5$	$1.485e-5$	$1.448e-6$	$5.586e-6$	0.01658	0.06956	0.0005914
14	0.001567	0.001212	0.001752	0.0003165	0.0002445	0.000263	0.1765	0.09138	0.1725
15	0.3371	0.3374	0.3371	0.01477	0.01507	0.01477	0.008984	0.01688	0.01434
16	0.08933	0.09108	0.08933	0.02163	0.02138	0.02163	0.1117	0.192	0.1692
17	0.001387	0.001311	0.001096	0.0004112	0.0004178	0.0002366	2.168	1.794	2.185
18	0.01656	0.01323	0.01651	0.01889	0.01909	0.01888	0.08023	0.1597	0.1594
19	0.0001887	0.0002499	0.0001895	$6.505e-5$	$5.274e-5$	$3.287e-5$	0.3556	0.3757	0.35
All	1.053	1.053	1.053	0.03541	0.03535	0.03541	3.141	3.115	3.162
RSS	1.053	1.053	1.053	0.03542	0.03538	0.03541	3.214	3.197	3.19
Difference	$-1.464e-5$	$-9.318e-6$	$-3.542e-6$	$-1.646e-5$	$-2.027e-5$	$-6.763e-7$	-0.07274	-0.08259	-0.02817

Table 7.4: Error Group % Contribution

Error Group	Position(m)			Velocity(m/s)			Attitude(deg)		
1	4.558	4.559	4.558	0.1633	0.1594	0.1635	6.567e − 6	1.214e − 5	1.563e − 6
2	0.2894	0.2689	0.2876	1.363	0.6885	1.357	0.009165	0.07214	0.05294
3	4.655e − 6	0.0001946	5.935e − 5	0.0006921	4.824e − 5	0.000194	5.586	7.797	5.716
4	80.28	80.26	80.28	0.01021	0.009963	0.01022	4.104e − 7	7.586e − 7	9.771e − 8
5	3.887	3.89	3.889	15.28	15.3	15.29	5.019e − 5	0.001082	0.0009122
6	0.0001104	1.956e − 5	1.574e − 5	3.053e − 5	0.004531	5.312e − 5	47.08	58.5	45.19
7	2.989e − 12	8.056e − 9	2.068e − 13	1.273e − 10	1.064e − 6	9.933e − 13	2.722e − 11	3.136e − 11	2.454e − 10
8	2.989e − 12	8.056e − 9	2.068e − 13	1.273e − 10	1.064e − 6	9.933e − 13	2.722e − 11	3.136e − 11	2.454e − 10
9	3.739e − 8	7.016e − 13	4.998e − 10	2.504e − 9	9.775e − 11	1.418e − 6	2.21e − 10	6.133e − 9	4.204e − 12
10	0.002094	0.001231	5.067e − 6	0.04093	0.01468	0.01082	0.003615	0.008862	0.01327
11	5.486e − 8	2.471e − 7	2.168e − 8	8.144e − 8	1.262e − 5	6.593e − 7	0.04435	0.0009755	0.04493
12	5.506e − 8	2.474e − 7	2.185e − 8	9.04e − 8	1.261e − 5	6.906e − 7	0.0443	0.001015	0.04497
13	1.022e − 7	9.083e − 9	1.848e − 7	1.756e − 5	1.675e − 7	2.489e − 6	0.00266	0.04733	3.437e − 6
14	0.0002212	0.0001323	0.0002766	0.007984	0.004777	0.005516	0.3016	0.08169	0.2923
15	10.24	10.26	10.24	17.38	18.15	17.4	0.0007813	0.002788	0.002019
16	0.7192	0.7475	0.7193	37.29	36.53	37.32	0.1208	0.3608	0.2813
17	0.0001735	0.0001548	0.0001083	0.01347	0.01395	0.004464	45.52	31.5	46.9
18	0.02472	0.01577	0.02456	28.45	29.12	28.44	0.06232	0.2495	0.2497
19	3.208e − 6	5.626e − 6	3.236e − 6	0.0003372	0.0002222	8.616e − 5	1.224	1.381	1.204

7.5 Sensitivity Analysis

The sensitivity analysis follows directly from the results of the error budget. The error budget was computed with static uncertainty and error values. While the error budget shows which states contribute the most towards the overall uncertainty, those contributions come from static values. Therefore, this only represents a single scenario of uncertainty. To properly analyze the effect each error group on the overall uncertainty, we must look at a range of uncertainty for each error group rather than a single point. By varying or scaling the uncertainty of each error group, we can see how it affects the Kalman filter overall estimation uncertainty. If certain error groups are larger than expected, the sensitivity analysis can illustrate how much of an effect a larger than expected error term has on the performance. If scaling the error group magnitude does not change the overall uncertainty much, then the estimate is not sensitive to the error group. However, if the overall uncertainty increases significantly with the scaling of the error group, then the estimate is very sensitive to that error term. This is the final step in determining which states or error terms can be neglected to simplify the filter. If an error group is seen to have little to no impact on all three main states (position, velocity, and attitude), then we may remove it from our filter. The sensitivity analysis procedure is as follows:

1. Following the error budget, select the first error group from the first column of the error budget table and scale it from 0.1 to 10.

2. Following the scaling, re-compute the root rum square of the error terms for the chosen column and do so for the entire range of scale factors chosen.

3. Store and plot the range of RSS values for that column (in our case, $\mathbf{r}_x$).

4. Move to the next error group and repeat the first two steps for the rest of the error groups.

5. Repeat the first four steps for the rest of the columns in the error budget table.

The sensitivity analysis results are presented in Figures 7.44 - 7.52. For convenience, position, velocity, and attitude have plots for each axis. Each plot illustrates the scaling effects of all error groups on the corresponding state element, allowing for a direct comparison between the effect each error group on the estimation error. All error groups are labeled in the legend and the most sensitive groups are called out by arrows with group designations attached to the arrow base. This makes it easy to see the groups that exhibit the most sensitivity, which are the most important groups to consider in this analysis. For more detailed sensitivity plots, the scaling effect of each error group is plotted individually for each state element in appendix B.

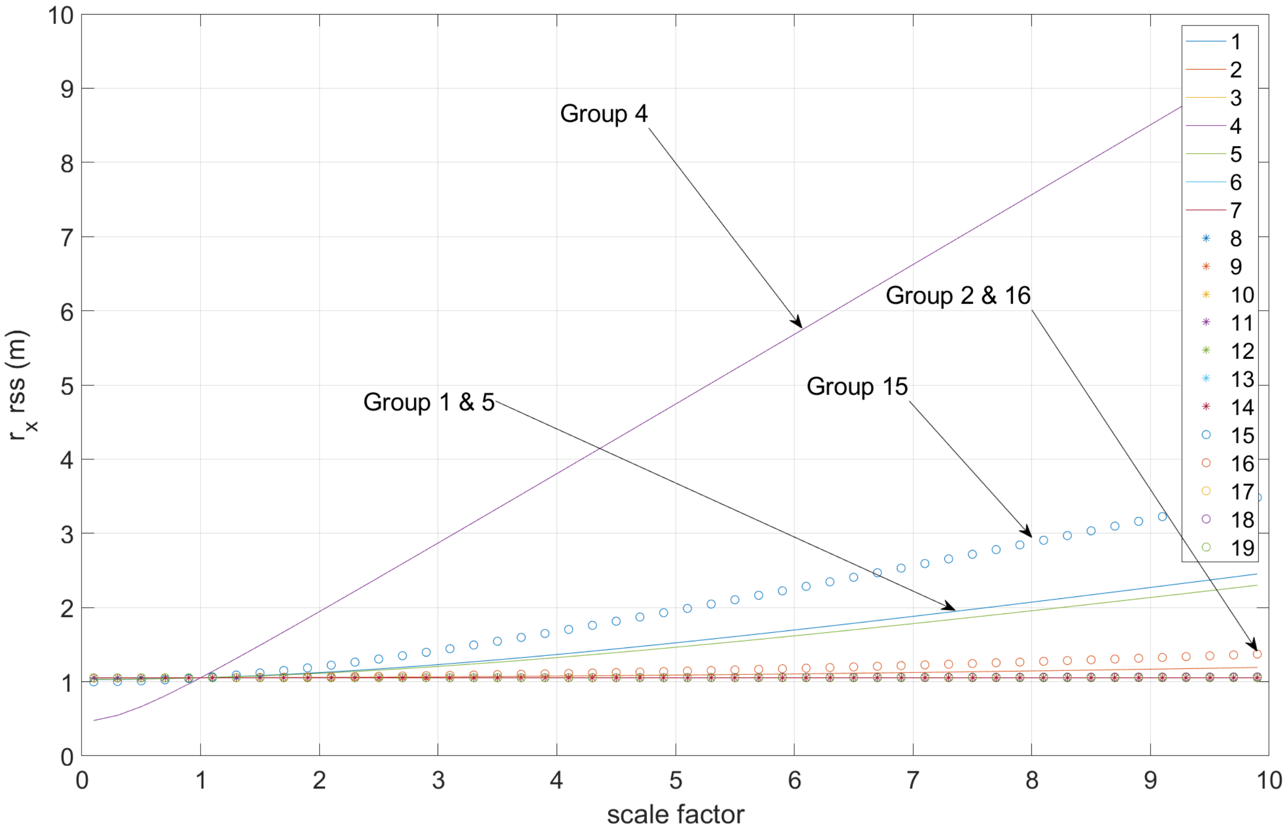


Figure 7.44: x-axis Position Sensitivity

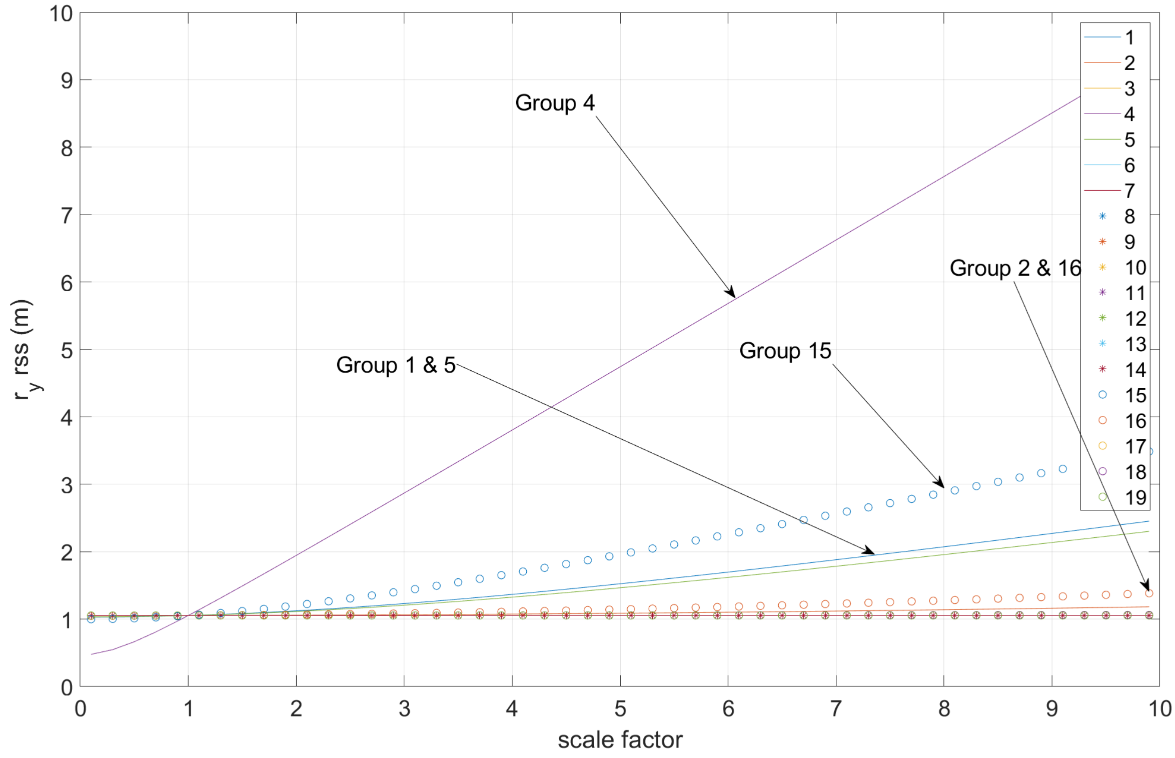


Figure 7.45: y-axis Position Sensitivity

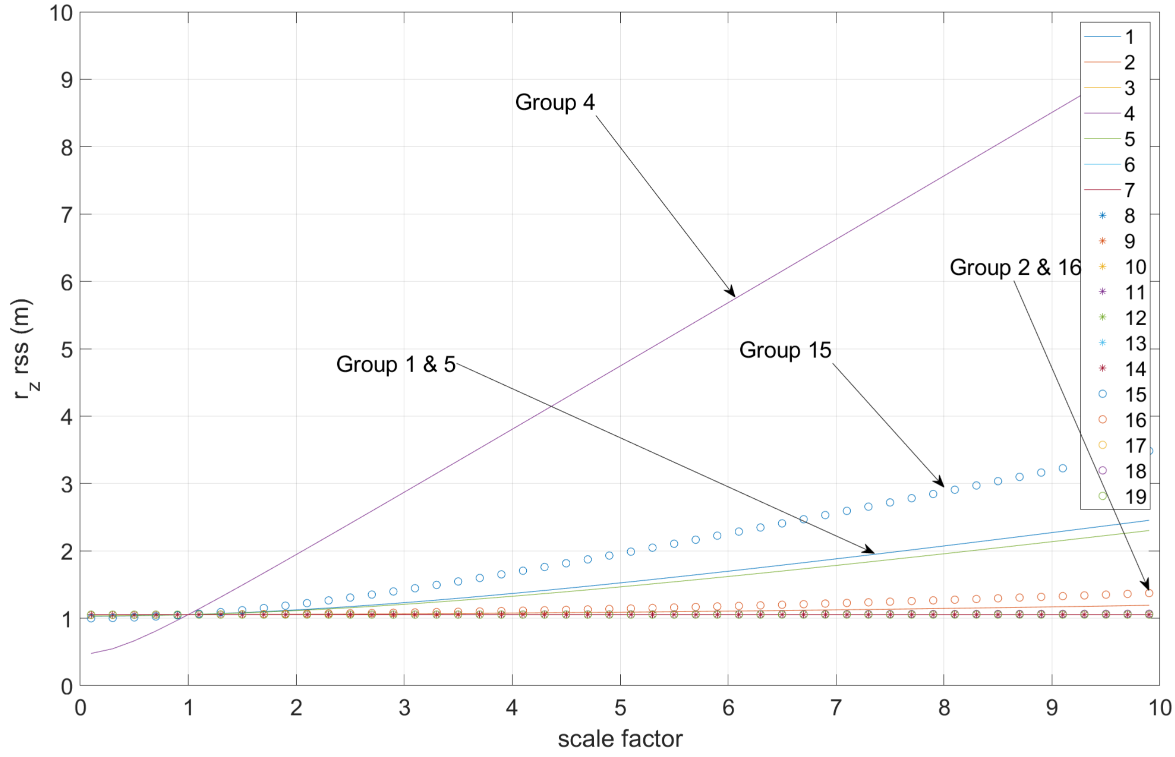


Figure 7.46: z-axis Position Sensitivity

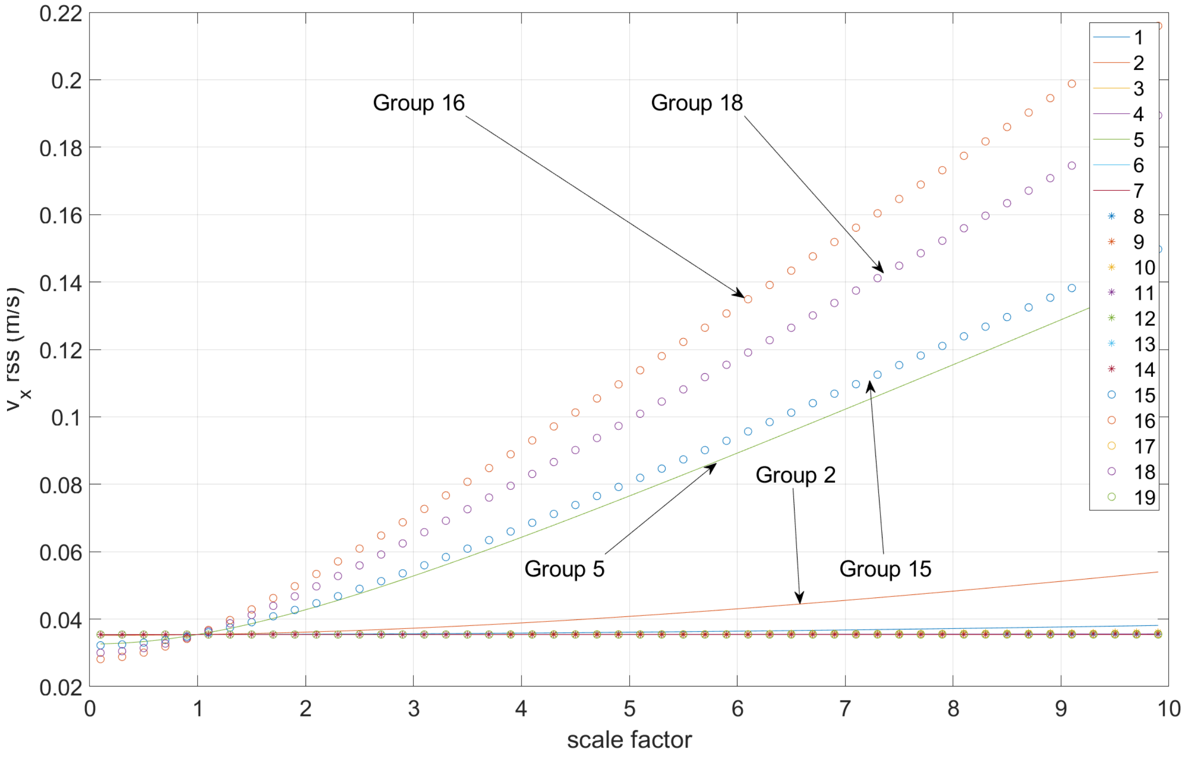


Figure 7.47: x-axis Velocity Sensitivity

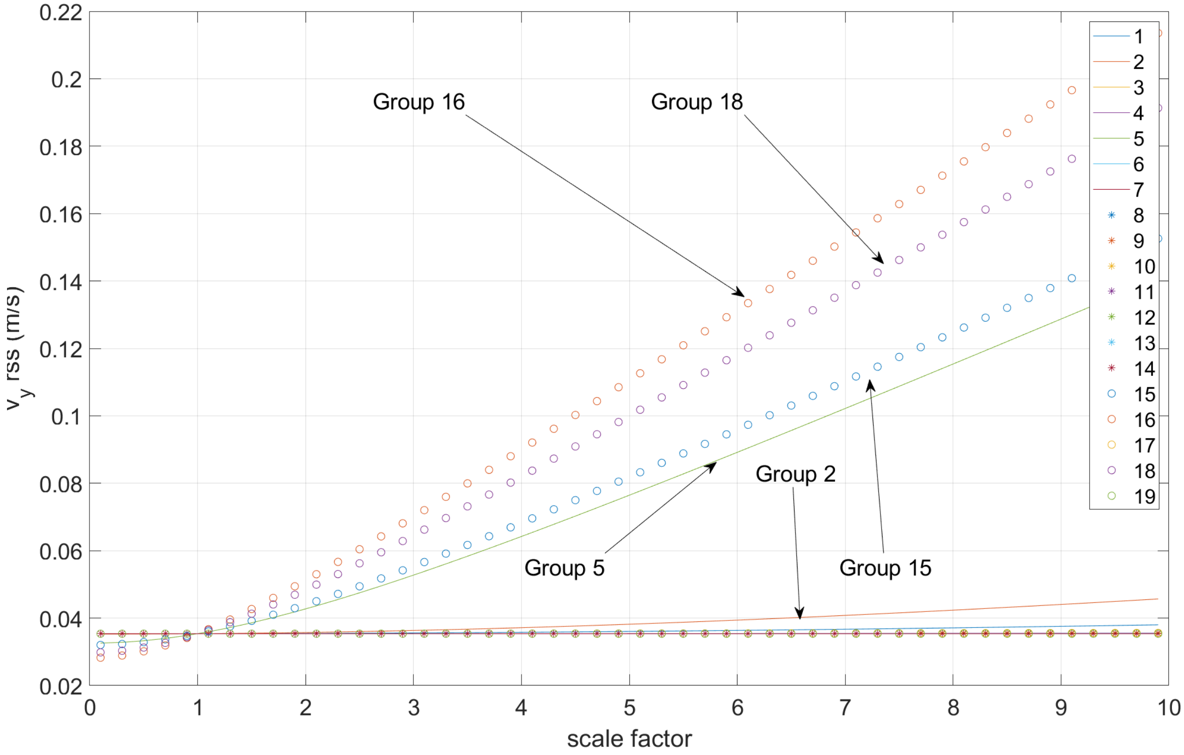


Figure 7.48: y-axis Velocity Sensitivity

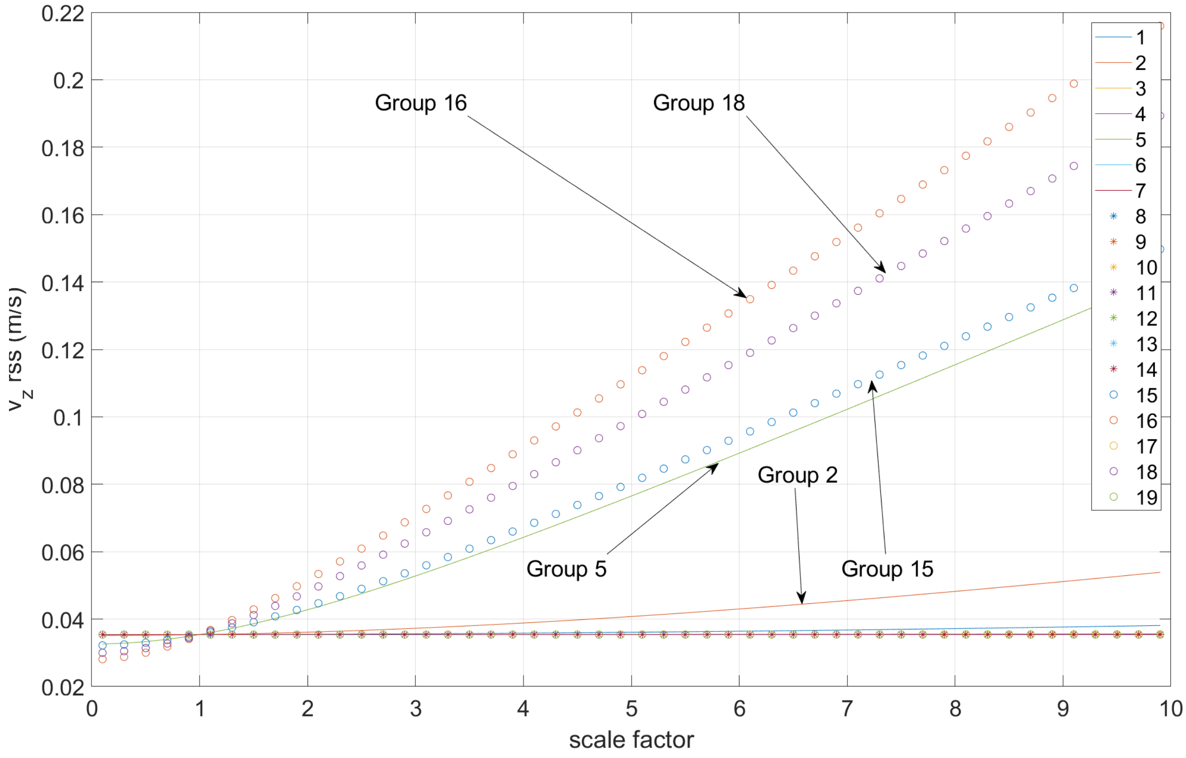


Figure 7.49: z-axis Velocity Sensitivity

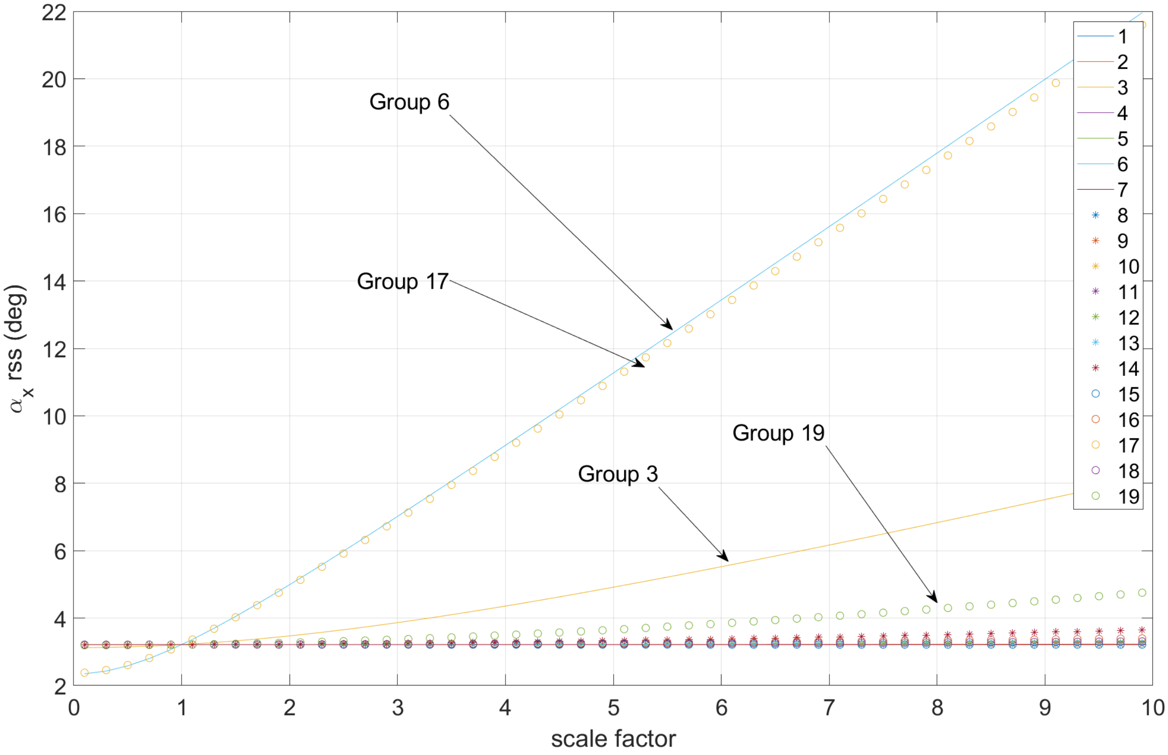


Figure 7.50: x-axis Attitude Sensitivity

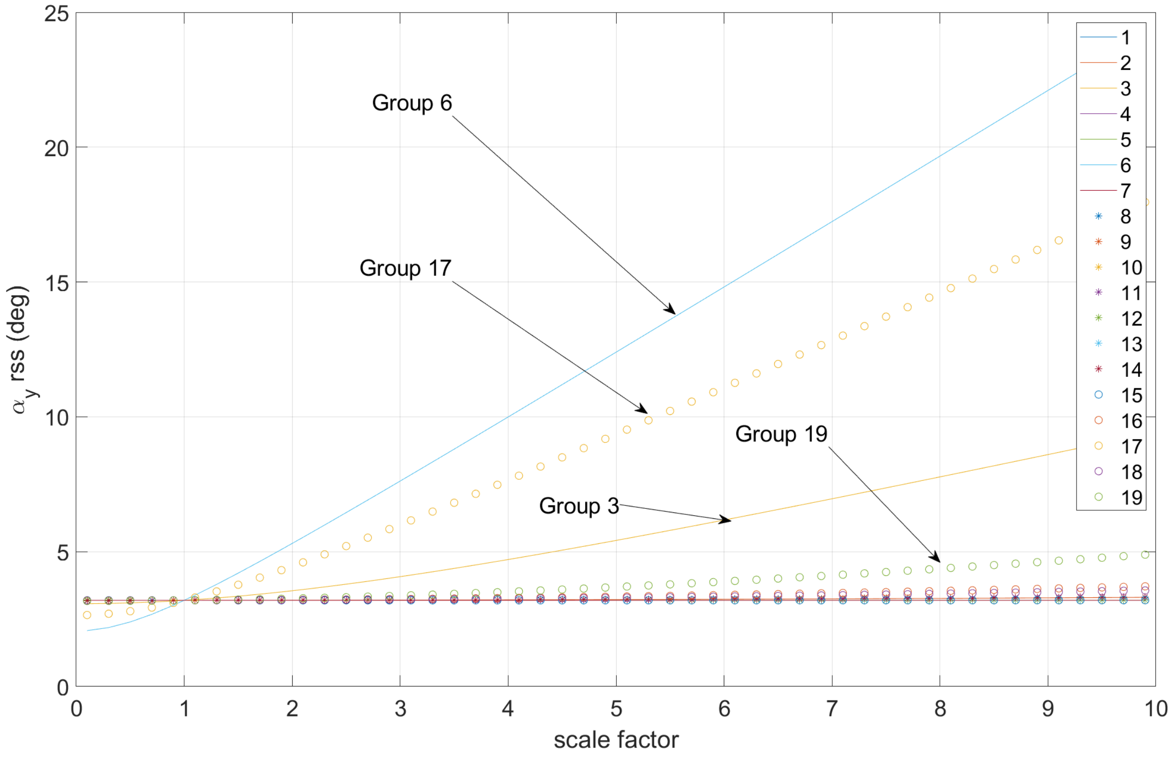


Figure 7.51: y-axis Attitude Sensitivity

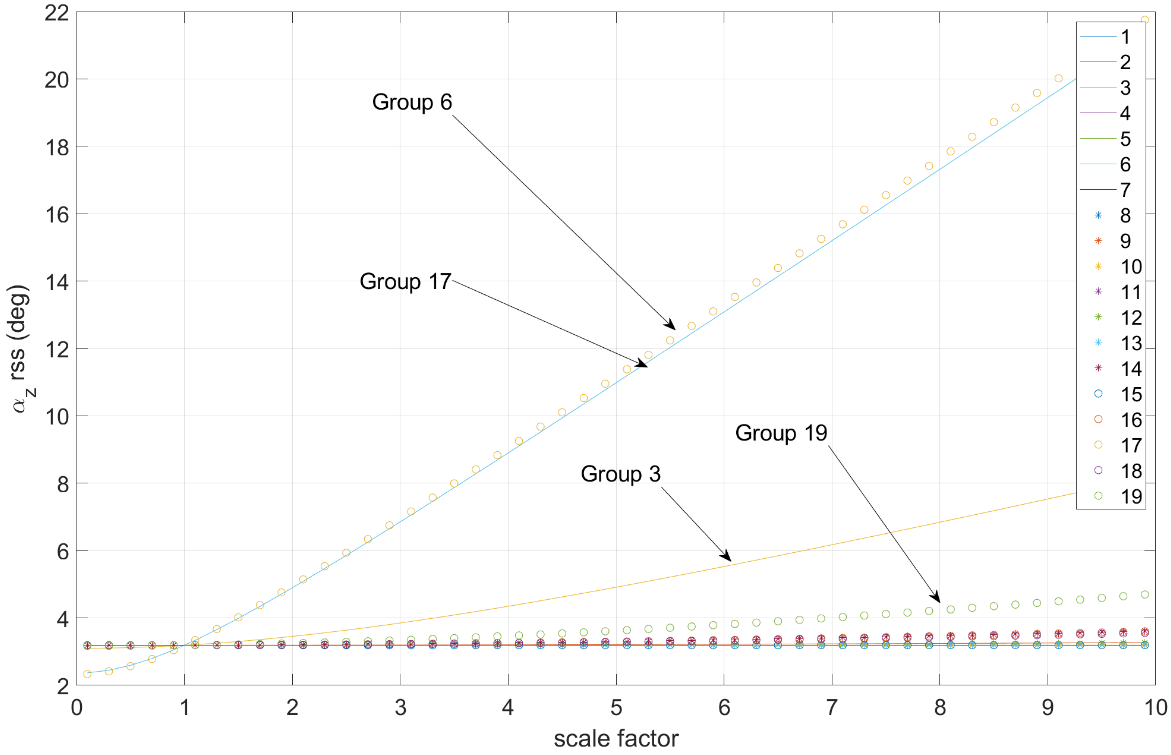


Figure 7.52: z-axis Attitude Sensitivity

From the sensitivity analysis, the Kalman filter uncertainty is most sensitive to groups 1, 2, 3, 4, 5, 6, 15, 16, 17, and 19. Therefore, groups 7, 8, 9, 10, 11, 12, 13, and 14 may be considered for removal due to their low sensitivity effects.

7.6 Sub-Optimal Filter Design

Given the computational limitations of a computer onboard a nanosatellite, an optimal filter such as the one derived in this thesis may be too computationally expensive. To increase performance while maintaining similar accuracy, we consider a sub-optimal filter which is derived from our optimal filter through removal of error terms that don't have any significant impact on the overall estimation uncertainty.

We now consider the results from both the error budget and sensitivity analysis. The error budget illuminated error groups that contributed the most to the overall state uncertainty, specifically, Table 7.4 showed us which error groups had the most percent contribution to the overall uncertainty. If a certain error group contributed very little, then it could be considered for removal following a sensitivity analysis. We now consider those terms. If those terms show little to no effect on the overall uncertainty when scaled throughout a range of scale factors, then they can be removed from the filter. From the error budget, the terms that contributed very little to the uncertainty of all states were groups 7, 8, 9, 11, 12, and 13. These are the accelerometer and gyroscope systematic errors, respectively. Since these error groups also intersect with the terms that cause the least effect on estimation sensitivity, these terms may be removed. Removing these error terms, our sub-optimal state then becomes

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}^i_{imu/i} & \mathbf{v}^i_{imu/i} & \delta\boldsymbol{\alpha} & \boldsymbol{\beta}^i_{gps/i} & \boldsymbol{\beta}^i_{gps} & \boldsymbol{\beta}^{tr}_q & \boldsymbol{\beta}_a & \boldsymbol{\beta}_g \end{bmatrix}^T .$$

With this new state defined, we proceed with removing the IMU systematic error terms from Eq. 6.3. Doing so will yield a new set of state estimation error differential equations for the sub-optimal filter,

$$
\begin{aligned}
\delta\dot{\mathbf{r}}^{i}_{imu/i} &= \delta\mathbf{v}^{i}_{imu/i} && (7.1)\\
\delta\dot{\mathbf{v}}^{i}_{imu} &= \mathbf{G}\delta\mathbf{r}^{i}_{imu/i} - \mathbf{G}\hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})[\mathbf{r}^{b}_{cg/imu}\times]\delta\boldsymbol{\alpha} - \hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})\mathbf{T}^{b}_{c}\delta\boldsymbol{\beta}_{a} - \hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})\mathbf{T}^{b}_{c}\boldsymbol{\eta}_{a}\\
&\quad - \hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})\mathbf{T}^{b}_{c}[\hat{\mathbf{a}}^{c}_{imu,ng}\times]\delta\boldsymbol{\alpha} && (7.2)\\
\delta\dot{\boldsymbol{\alpha}} &= -\mathbf{T}^{b}_{c}\delta\boldsymbol{\beta}_{g} - \mathbf{T}^{b}_{c}\boldsymbol{\eta}_{g} - \mathbf{T}^{b}_{c}[\hat{\boldsymbol{\omega}}^{c}_{b/i}\times]\delta\boldsymbol{\alpha} && (7.3)\\
\delta\dot{\boldsymbol{\beta}}^{i}_{gps/i} &= \mathbf{0} && (7.4)\\
\delta\dot{\boldsymbol{\beta}}^{i}_{gps} &= \mathbf{0} && (7.5)\\
\delta\dot{\boldsymbol{\beta}}^{tr}_{q} &= \mathbf{0} && (7.6)\\
\delta\dot{\boldsymbol{\beta}}_{a} &= \mathbf{0} && (7.7)\\
\delta\dot{\boldsymbol{\beta}}_{g} &= \mathbf{0}\,. && (7.8)
\end{aligned}
$$

Removing the IMU systematic error terms from Eq. 6.7 will yield a new set of measurement deviation equations for the sub-optimal filter,

$$
\begin{aligned}
\delta\mathbf{r}^{f}_{gps} &= \mathbf{T}^{f}_{i}(t)\delta\mathbf{r}^{i}_{imu} - \mathbf{T}^{f}_{i}(t)\hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})[\mathbf{r}^{b}_{gps/imu}\times]\delta\boldsymbol{\alpha} + \delta\boldsymbol{\beta}^{f}_{pos} + \boldsymbol{\eta}^{f}_{pos} && (7.9)\\
\delta\mathbf{v}^{f}_{gps} &= \mathbf{T}^{f}_{i}(t)\delta\mathbf{v}^{i}_{imu} + \mathbf{T}^{f}_{i}(t)\hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})[\mathbf{r}^{b}_{gps/imu}\times]\mathbf{T}^{b}_{c}\delta\boldsymbol{\beta}_{g} + \mathbf{T}^{f}_{i}(t)\hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})[\mathbf{r}^{b}_{gps/imu}\times]\mathbf{T}^{b}_{c}\boldsymbol{\eta}_{g}\\
&\quad + \mathbf{T}^{f}_{i}(t)\hat{\mathbf{T}}^{i}_{b}(\hat{\bar{\mathbf{q}}}^{i}_{b})\left[\left[\mathbf{r}^{b}_{gps/imu}\times\mathbf{T}^{b}_{c}\hat{\boldsymbol{\omega}}^{c}_{b/i}\right]\times\right]\delta\boldsymbol{\alpha} + \delta\boldsymbol{\beta}^{f}_{vel} + \boldsymbol{\eta}^{f}_{vel} && (7.10)\\
\delta\boldsymbol{\Psi} &= \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}^{tr}_{b}\delta\boldsymbol{\alpha} + \delta\boldsymbol{\beta}_{tr} + \boldsymbol{\eta}_{tr}\,. && (7.11)
\end{aligned}
$$

This is a significant reduction in complexity compared to the original state estimation error differential equations and measurement deviation equations.

There are 24 states in the state vector. Therefore, the sub-optimal $\mathbf{F} \in \mathbb{R}^{24x24}$ is

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_{3x3} & \frac{\partial \delta \dot{\mathbf{r}}^i_{imu}}{\partial \mathbf{v}^i_{imu}} & \mathbf{0}_{3x3} & \mathbf{0}_{3x9} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \mathbf{r}^i_{imu}} & \mathbf{0}_{3x3} & \frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \delta \boldsymbol{\alpha}} & \mathbf{0}_{3x9} & \frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \boldsymbol{\beta}_a} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \frac{\partial \delta \dot{\boldsymbol{\alpha}}}{\partial \delta \boldsymbol{\alpha}} & \mathbf{0}_{3x9} & \mathbf{0}_{3x3} & \frac{\partial \delta \dot{\boldsymbol{\alpha}}}{\partial \boldsymbol{\beta}_g} \\ \mathbf{0}_{15x3} & \mathbf{0}_{15x3} & \mathbf{0}_{15x3} & \mathbf{0}_{15x9} & \mathbf{0}_{15x3} & \mathbf{0}_{15x3} \end{bmatrix},$$

where

$$\begin{aligned} \frac{\partial \delta \dot{\mathbf{r}}^i_{imu}}{\partial \mathbf{v}^i_{imu}} &= \mathbf{I}_{3x3} \\ \frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \mathbf{r}^i_{imu}} &= \mathbf{G}\mathbf{I}_{3x3} \\ \frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \delta \boldsymbol{\alpha}} &= -\mathbf{G}\hat{\mathbf{T}}^i_b(\hat{\hat{\mathbf{q}}}^i_b)[\mathbf{r}^b_{cg/imu}\times] - \hat{\mathbf{T}}^i_b(\hat{\hat{\mathbf{q}}}^i_b)\mathbf{T}^b_c[\hat{\mathbf{a}}^c_{imu,ng}\times] \\ \frac{\partial \delta \dot{\mathbf{v}}^i_{imu}}{\partial \boldsymbol{\beta}_a} &= -\hat{\mathbf{T}}^i_b(\hat{\hat{\mathbf{q}}}^i_b)\mathbf{T}^b_c \\ \frac{\partial \delta \dot{\boldsymbol{\alpha}}}{\partial \delta \boldsymbol{\alpha}} &= -\mathbf{T}^b_c[\hat{\boldsymbol{\omega}}^c_{b/i}\times] \\ \frac{\partial \delta \dot{\boldsymbol{\alpha}}}{\partial \boldsymbol{\beta}_g} &= -\mathbf{T}^b_c\,. \end{aligned}$$

The sub-optimal $\mathbf{H} \in \mathbb{R}^{9x24}$ is

$$\mathbf{H} = \begin{bmatrix} \frac{\partial \delta \mathbf{r}^f_{gps}}{\partial \mathbf{r}^i_{imu}} & \mathbf{0}_{3x3} & \frac{\partial \delta \mathbf{r}^f_{gps}}{\partial \delta \boldsymbol{\alpha}} & \frac{\partial \delta \mathbf{r}^f_{gps}}{\partial \boldsymbol{\beta}^f_{pos}} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \mathbf{v}^i_{imu}} & \frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \delta \boldsymbol{\alpha}} & \mathbf{0}_{3x3} & \frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \boldsymbol{\beta}^f_{vel}} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \frac{\partial \delta \mathbf{v}^f_{gps}}{\partial \boldsymbol{\beta}_g} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \frac{\partial \delta \boldsymbol{\Psi}}{\partial \delta \boldsymbol{\alpha}} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \frac{\partial \delta \boldsymbol{\Psi}}{\partial \boldsymbol{\beta}^{tr}_q} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{bmatrix}.$$

where

$$\begin{aligned}
\frac{\partial \delta \mathbf{r}^{f}_{gps}}{\partial \mathbf{r}^{i}_{imu}} &= \mathbf{T}^{f}_{i}(t) \\
\frac{\partial \delta \mathbf{r}^{i}_{gps}}{\partial \delta \boldsymbol{\alpha}} &= -\mathbf{T}^{f}_{i}(t)\hat{\mathbf{T}}^{i}_{b}(\hat{\mathbf{q}}^{i}_{b})[\mathbf{r}^{b}_{gps/imu}\times] \\
\frac{\partial \delta \mathbf{r}^{i}_{gps}}{\partial \boldsymbol{\beta}^{f}_{pos}} &= \mathbf{I}_{3x3} \\
\frac{\partial \delta \mathbf{v}^{i}_{gps}}{\partial \mathbf{v}^{i}_{imu}} &= \mathbf{T}^{f}_{i}(t) \\
\frac{\partial \delta \mathbf{v}^{i}_{gps}}{\partial \delta \boldsymbol{\alpha}} &= \mathbf{T}^{f}_{i}(t)\hat{\mathbf{T}}^{i}_{b}(\hat{\mathbf{q}}^{i}_{b})\left[\left[\mathbf{r}^{b}_{gps/imu} \times \mathbf{T}^{b}_{c}\hat{\boldsymbol{\omega}}^{c}_{b/i}\right] \times\right] \\
\frac{\partial \delta \mathbf{v}^{f}_{gps}}{\partial \boldsymbol{\beta}^{f}_{vel}} &= \mathbf{I}_{3x3} \\
\frac{\partial \delta \mathbf{v}^{f}_{gps}}{\partial \boldsymbol{\beta}_{g}} &= \mathbf{T}^{f}_{i}(t)\hat{\mathbf{T}}^{i}_{b}(\hat{\mathbf{q}}^{i}_{b})[\mathbf{r}^{b}_{gps/imu}\times]\mathbf{T}^{b}_{c} \\
\frac{\partial \delta \boldsymbol{\Psi}}{\partial \delta \boldsymbol{\alpha}} &= \hat{\mathbf{T}}_{\beta,\eta}\mathbf{T}^{tr}_{b} \\
\frac{\partial \delta \boldsymbol{\Psi}}{\partial \boldsymbol{\beta}^{tr}_{q}} &= \mathbf{I}_{3x3}\ .
\end{aligned}$$

Since this sub-optimal filter is created by removing some of the error terms, we may have to increase our model uncertainty by modifying the process noise matrix. This is dependent on simulation results. We are removing error terms from the filter model, making the model less accurate which directly affects the estimation error. To make up for this in the case that the accuracy exhibits a noticeable change, the model uncertainty is increased to accommodate for these changes. In the case of our simulations, there was no significant or noticeable change in estimation error, therefore, further significant tuning of the process noise was not necessary.

7.6.1 Sub-Optimal Filter Single Run Results

Figures 7.53 - 7.75 depict single run results for all states. For the states that contain measurements, comparison plots are shown. For the rest of the states, covariance plots are shown.

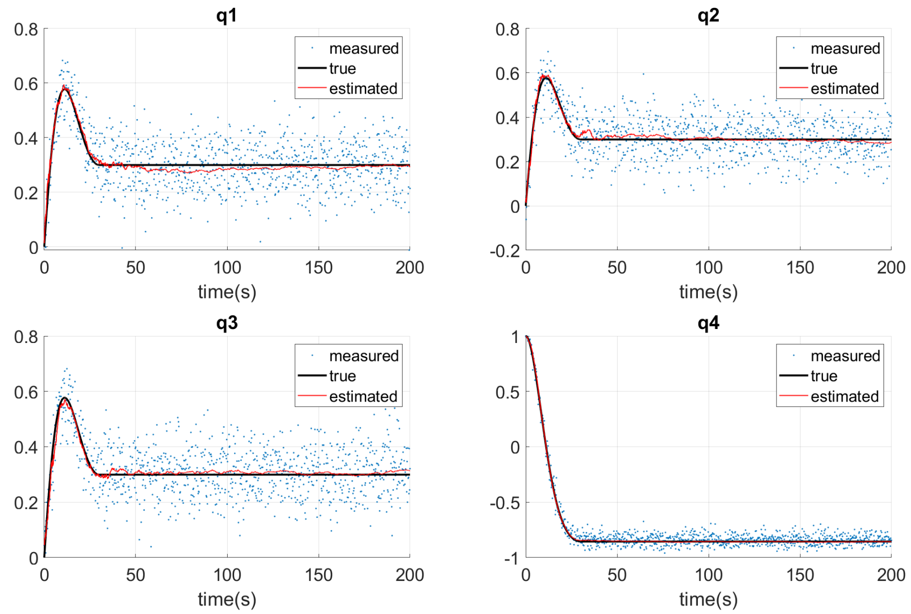


Figure 7.53: Quaternion Vector Elements

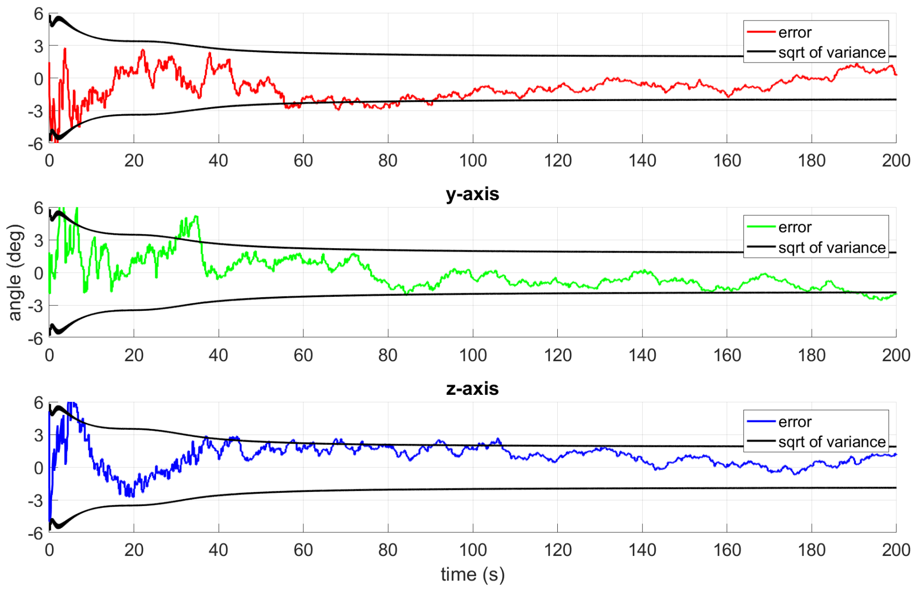


Figure 7.54: Attitude Error and Covariance

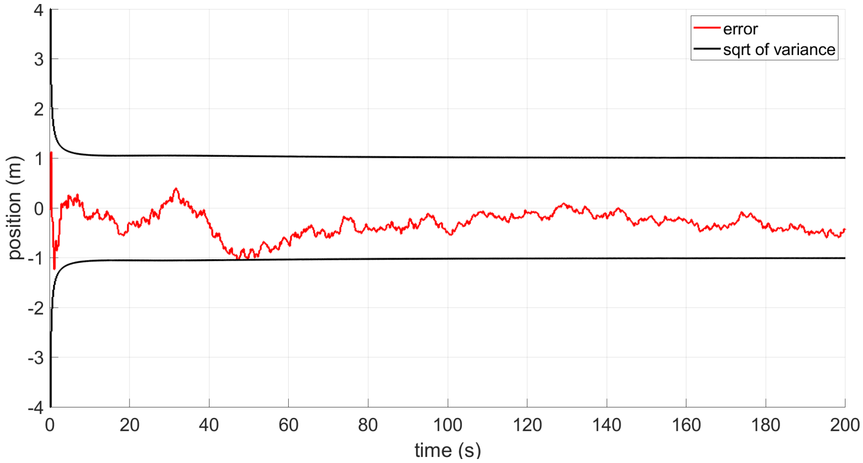


Figure 7.55: GPS Position Error and Error Covariance (x-axis)

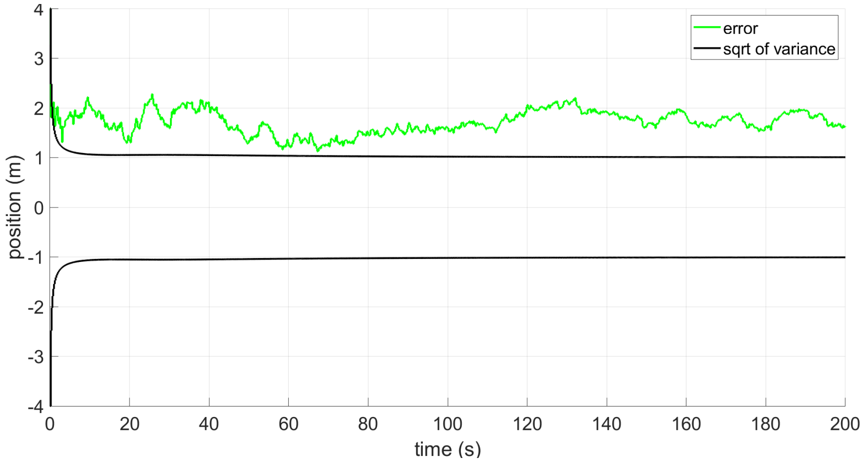


Figure 7.56: GPS Position Error and Error Covariance (y-axis)

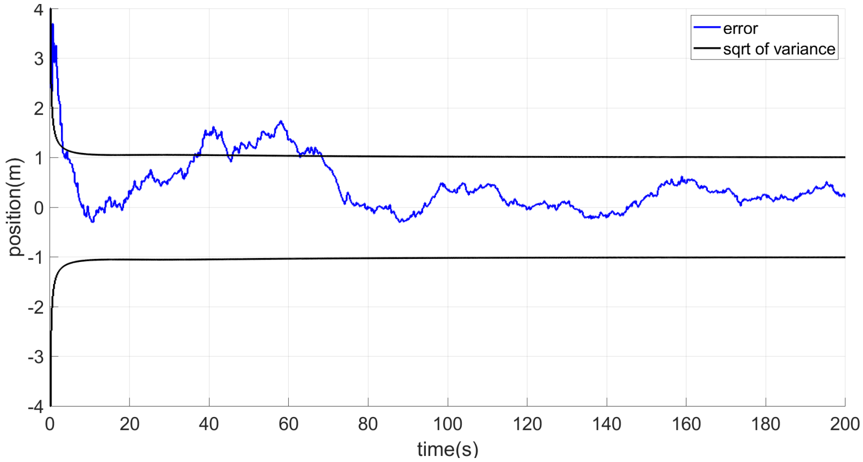


Figure 7.57: GPS Position Error and Error Covariance (z-axis)

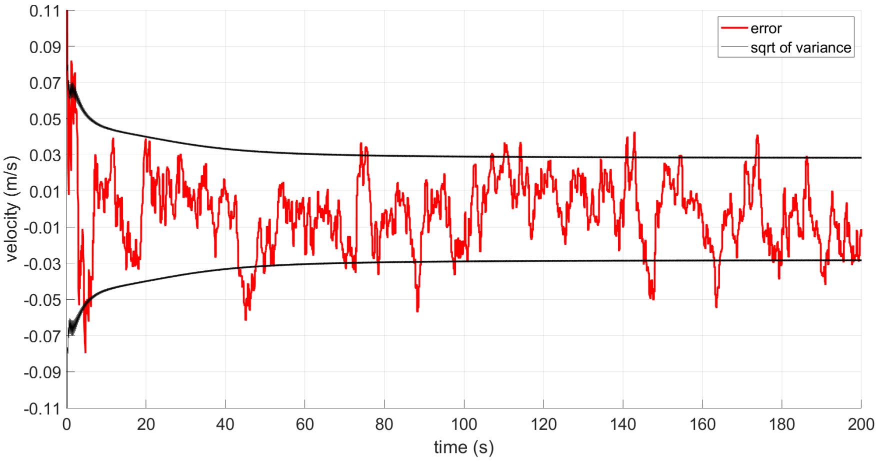


Figure 7.58: Velocimeter Error and Error Covariance (x-axis)

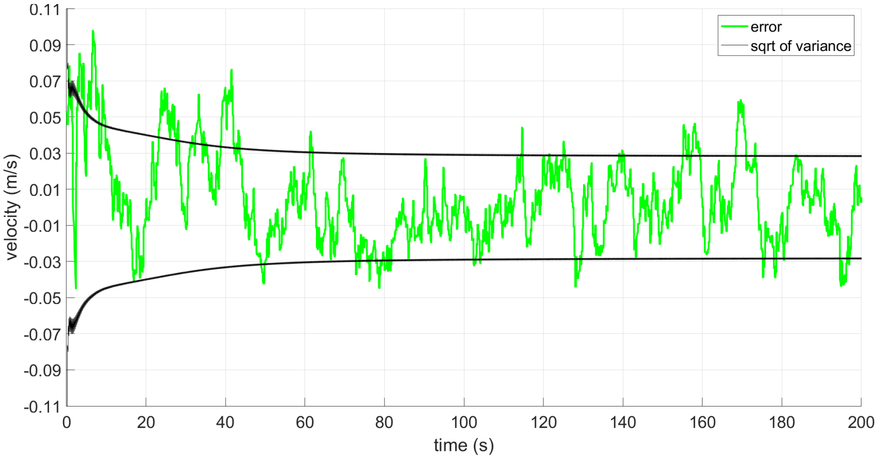


Figure 7.59: Velocimeter Error and Error Covariance (y-axis)

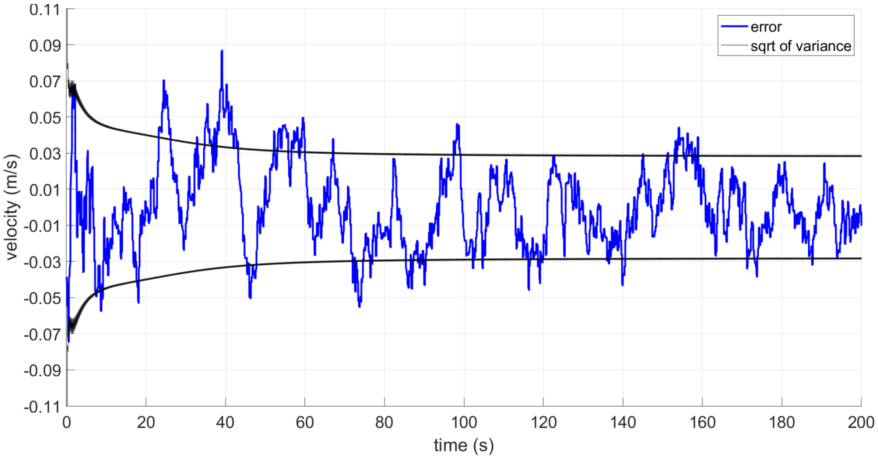


Figure 7.60: Velocimeter Error and Error Covariance (z-axis)

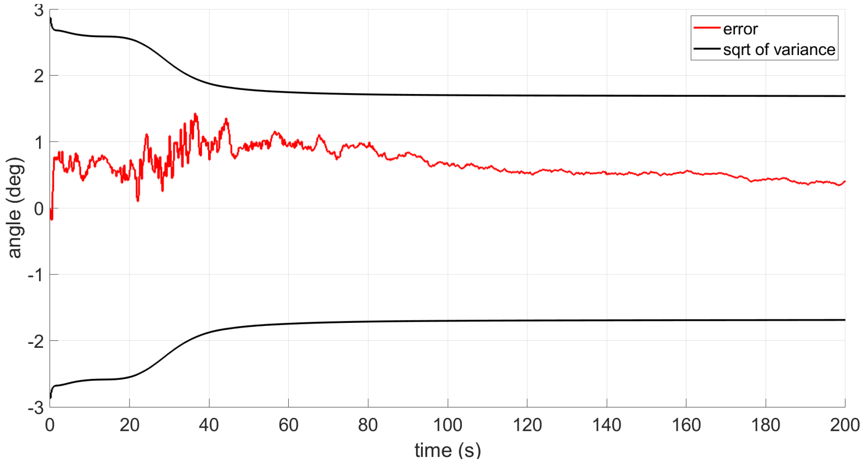


Figure 7.61: Quaternion Bias Error and Error Covariance (x-axis)

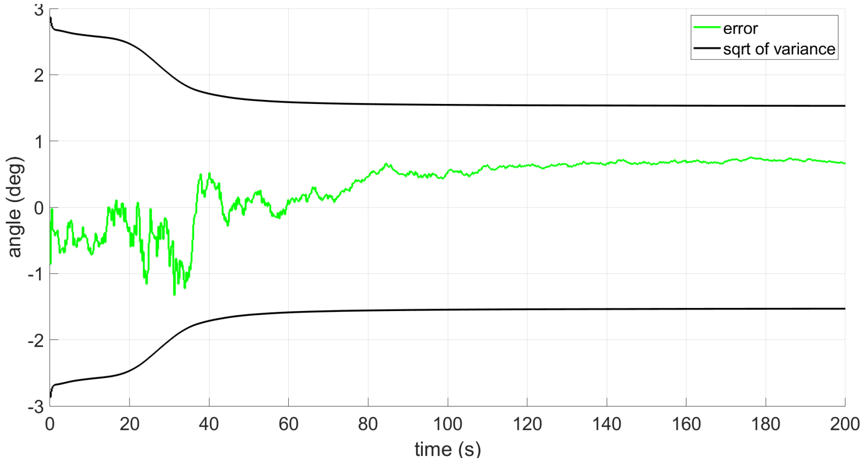


Figure 7.62: Quaternion Bias Error and Error Covariance (y-axis)

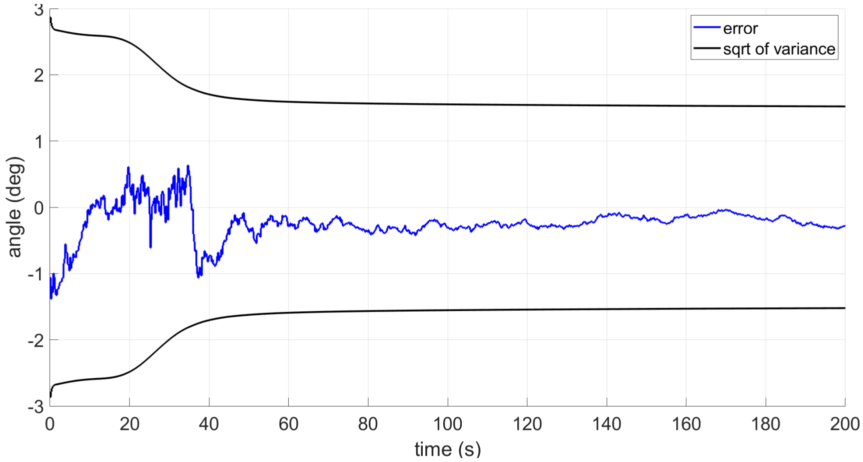


Figure 7.63: Quaternion Bias Error and Error Covariance (z-axis)

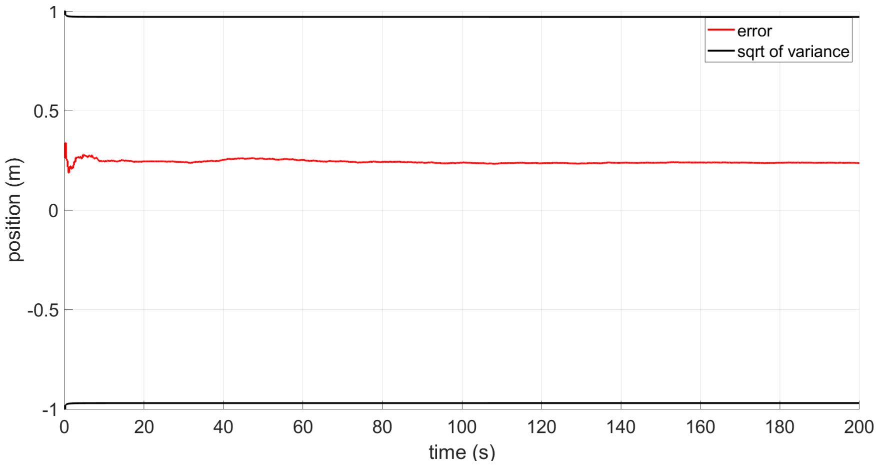


Figure 7.64: GPS Position Bias Error and Error Covariance (x-axis)

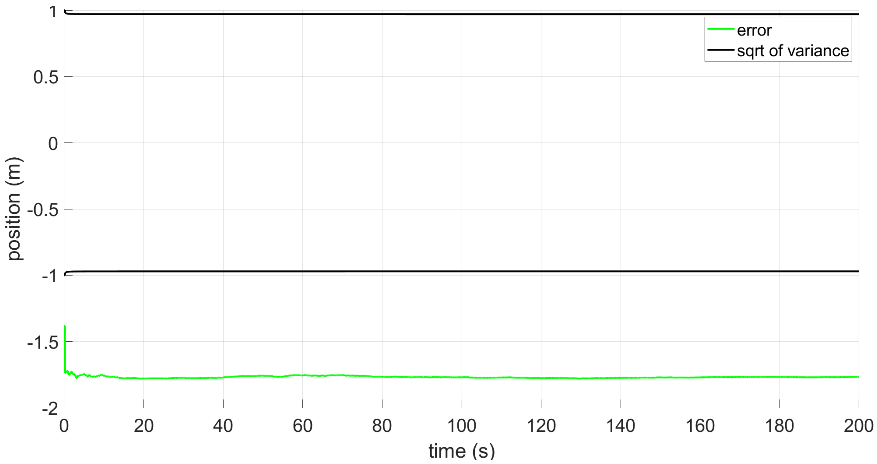


Figure 7.65: GPS Position Bias Error and Error Covariance (y-axis)

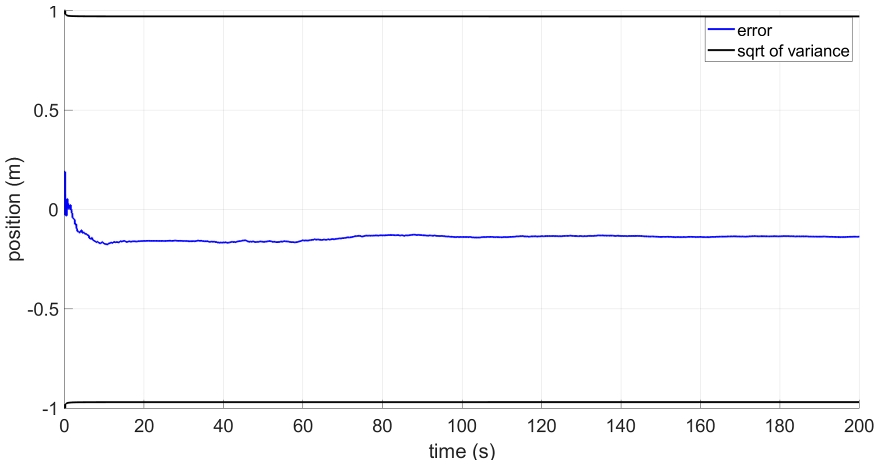


Figure 7.66: GPS Position Bias Error and Error Covariance (z-axis)

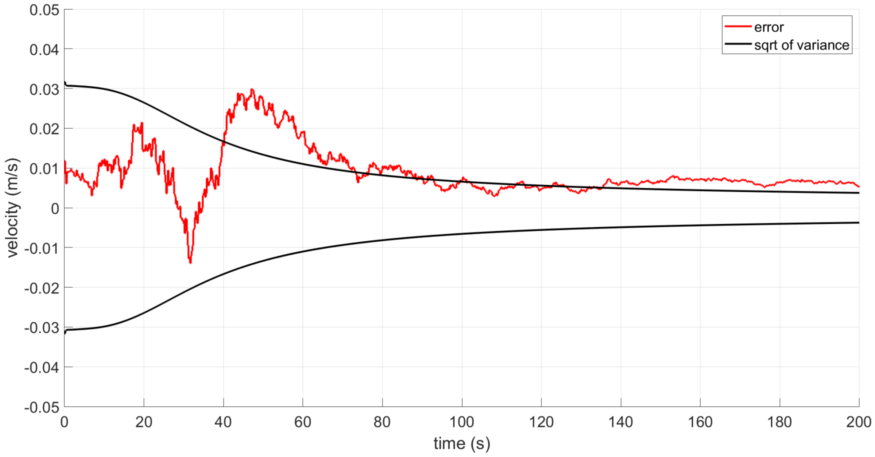


Figure 7.67: GPS Velocity Bias Error and Error Covariance (x-axis)

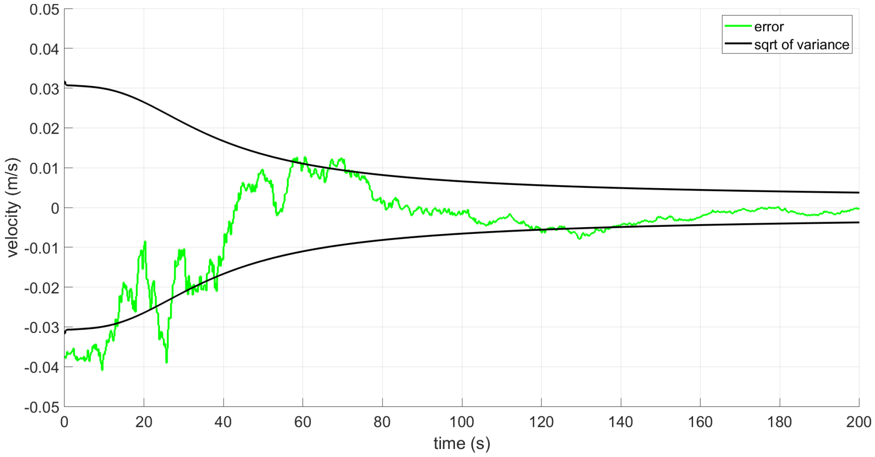


Figure 7.68: GPS Velocity Bias Error and Error Covariance (y-axis)

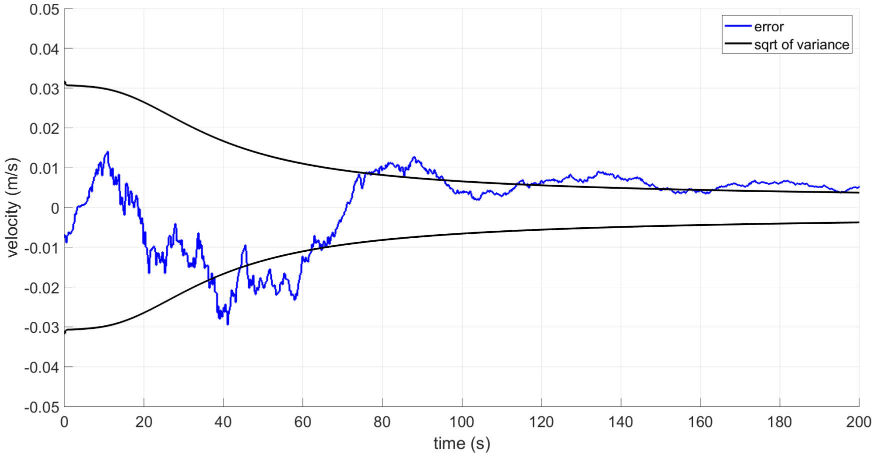


Figure 7.69: GPS Velocity Bias Error and Error Covariance (z-axis)

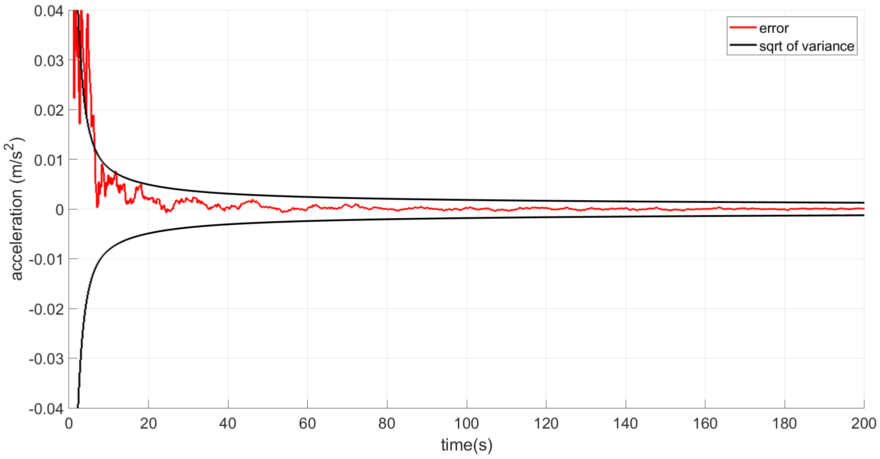


Figure 7.70: Accelerometer Bias Error and Error Covariance (x-axis)

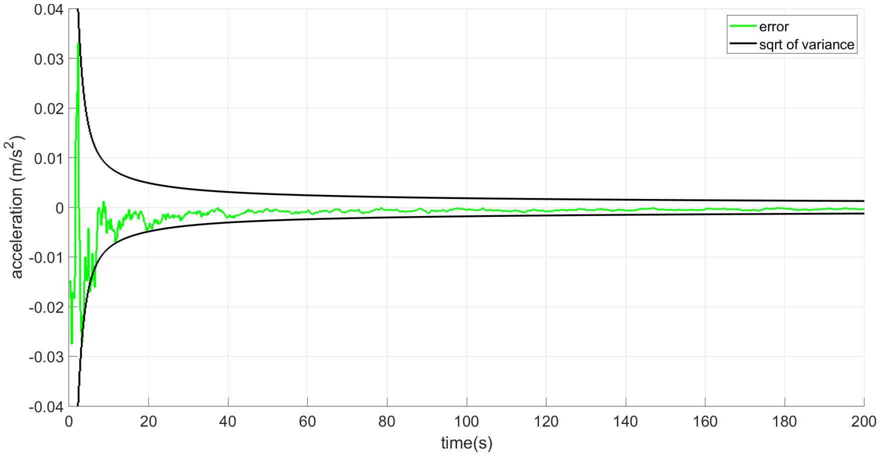


Figure 7.71: Accelerometer Bias Error and Error Covariance (y-axis)

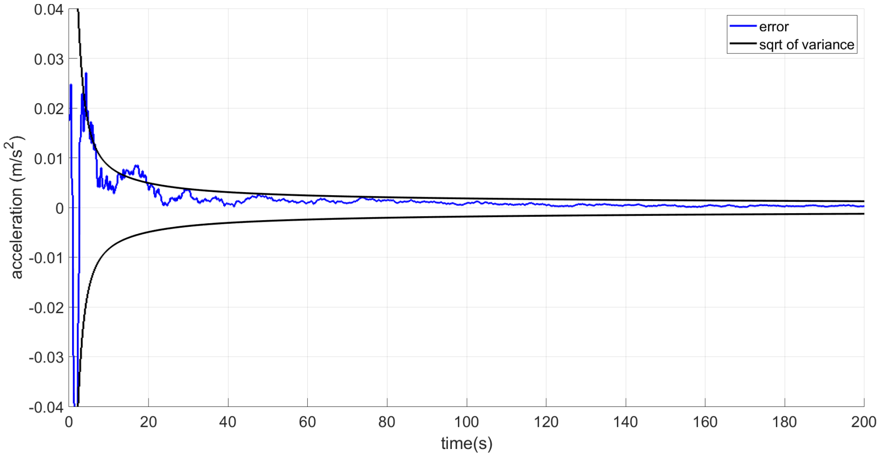


Figure 7.72: Accelerometer Bias Error and Error Covariance (z-axis)

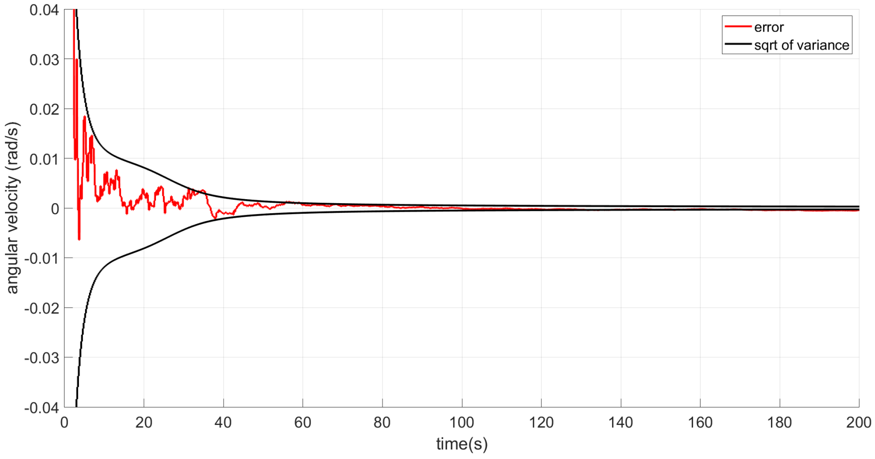


Figure 7.73: Gyroscope Bias Error and Error Covariance (x-axis)

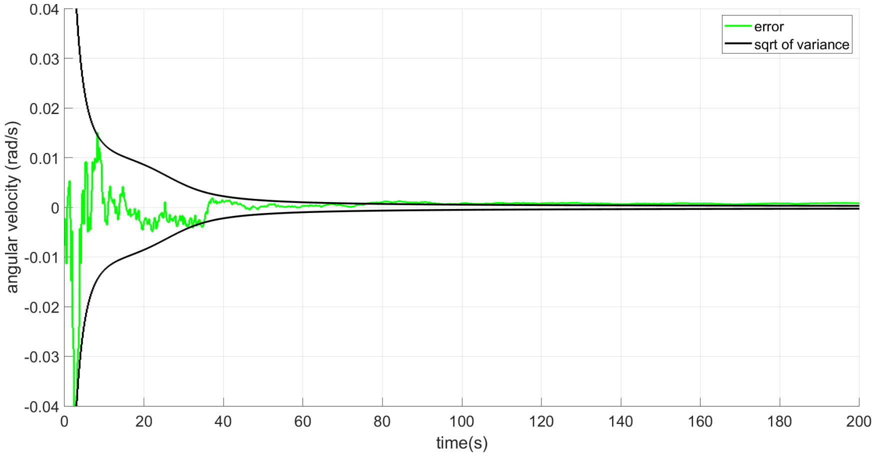


Figure 7.74: Gyroscope Bias Error and Error Covariance (y-axis)

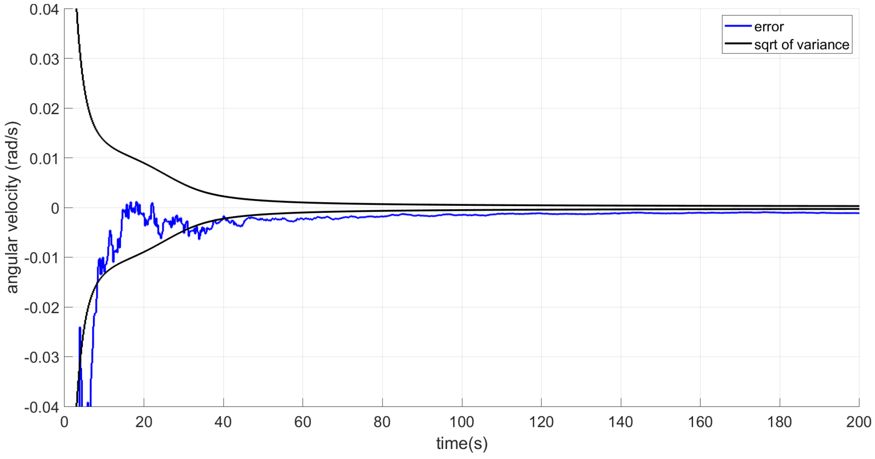


Figure 7.75: Gyroscope Bias Error and Error Covariance (z-axis)

7.6.2 Sub-Optimal Filter Monte Carlo Results

The Monte-Carlo results are depicted in Figures 7.76 - 7.83.

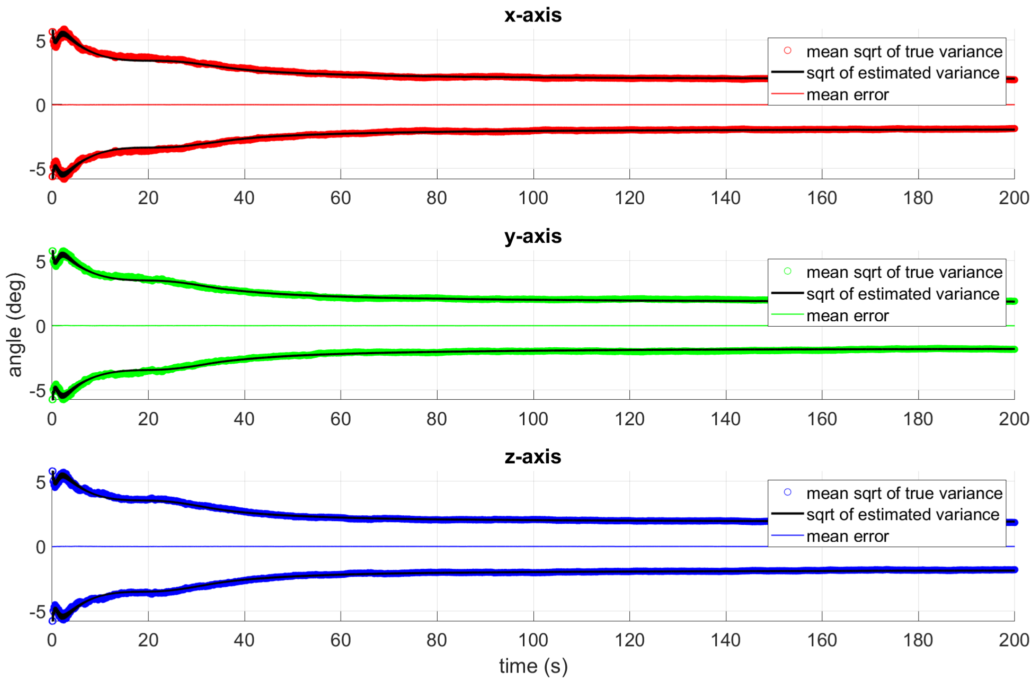


Figure 7.76: Monte-Carlo Attitude

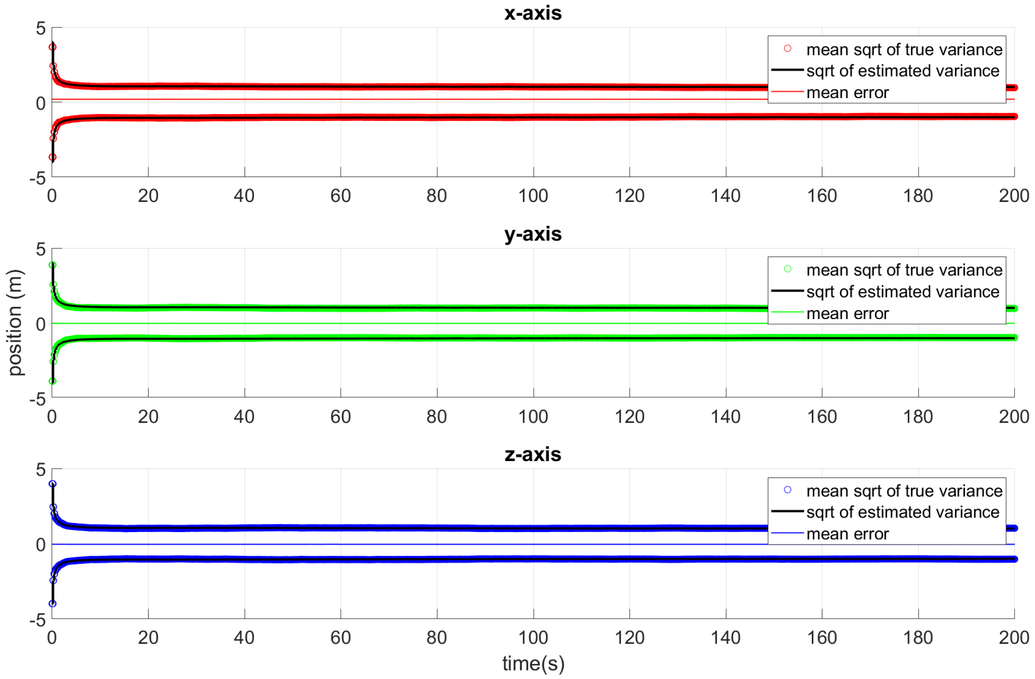


Figure 7.77: Monte-Carlo Position

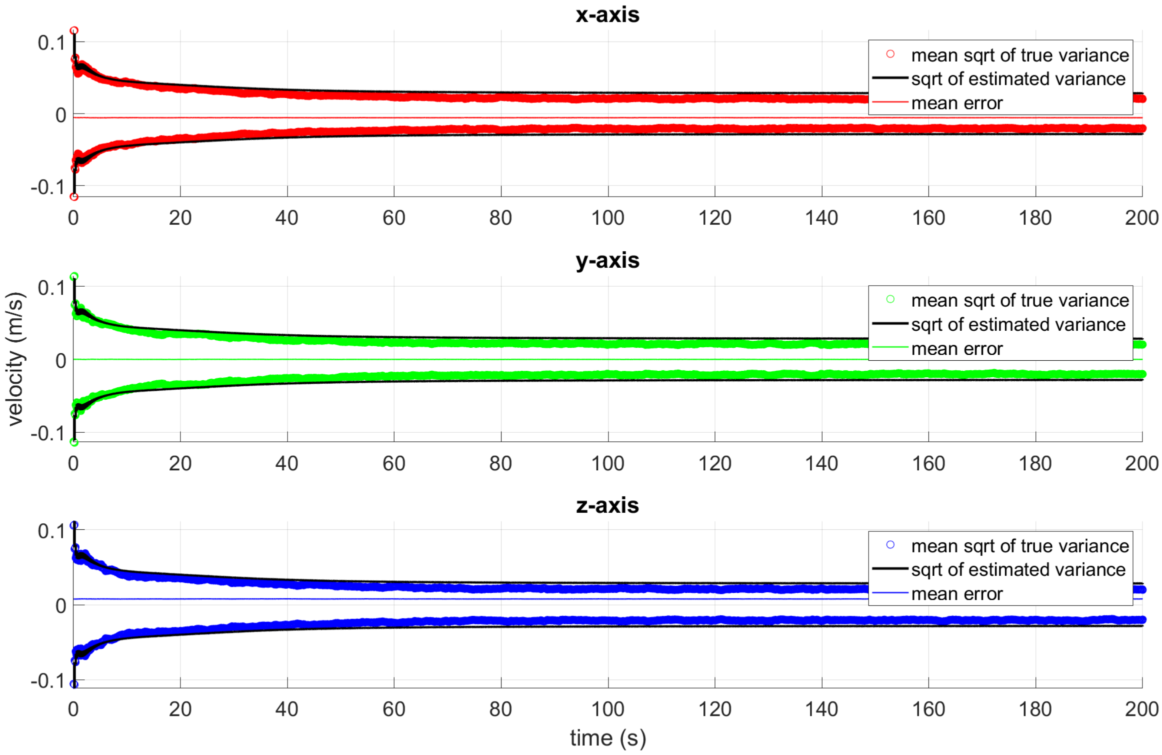


Figure 7.78: Monte-Carlo Velocity

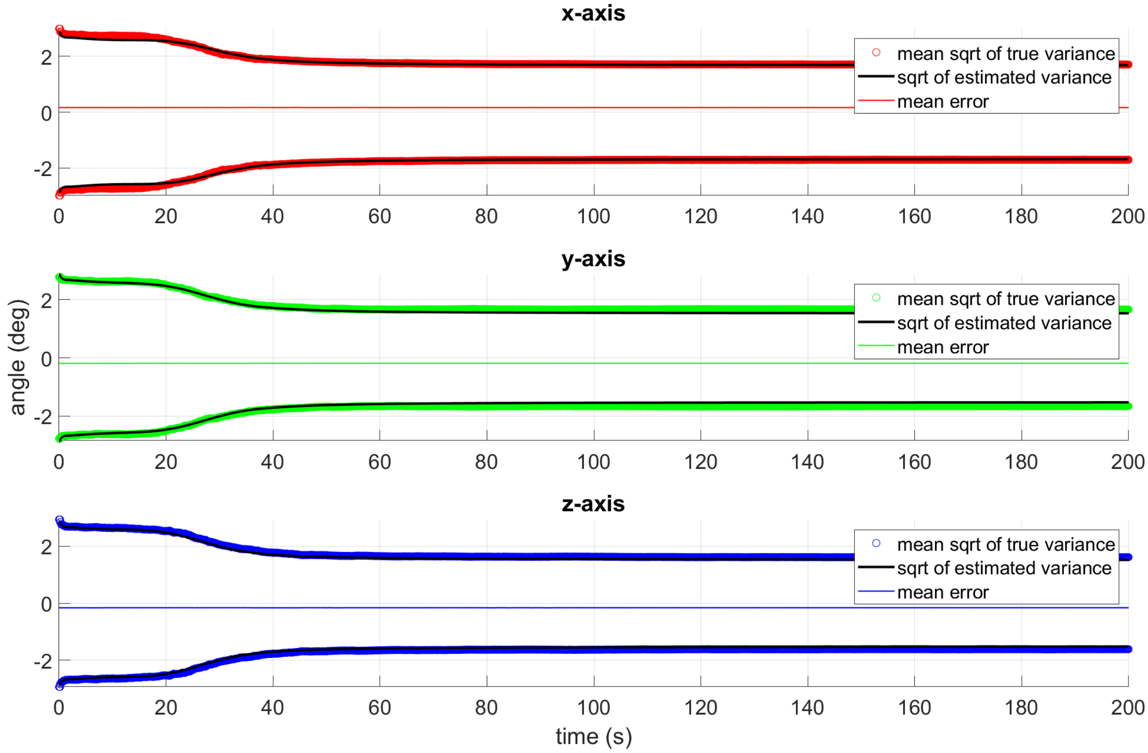


Figure 7.79: Monte-Carlo Quaternion Bias

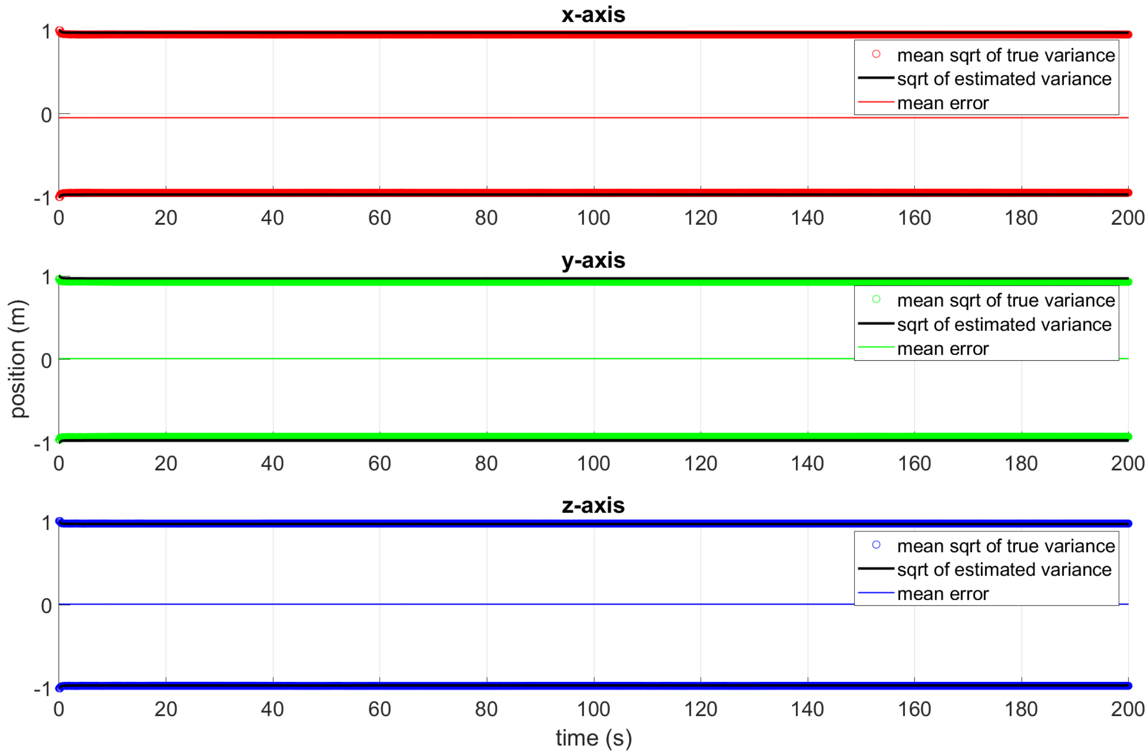


Figure 7.80: Monte-Carlo GPS Position Bias

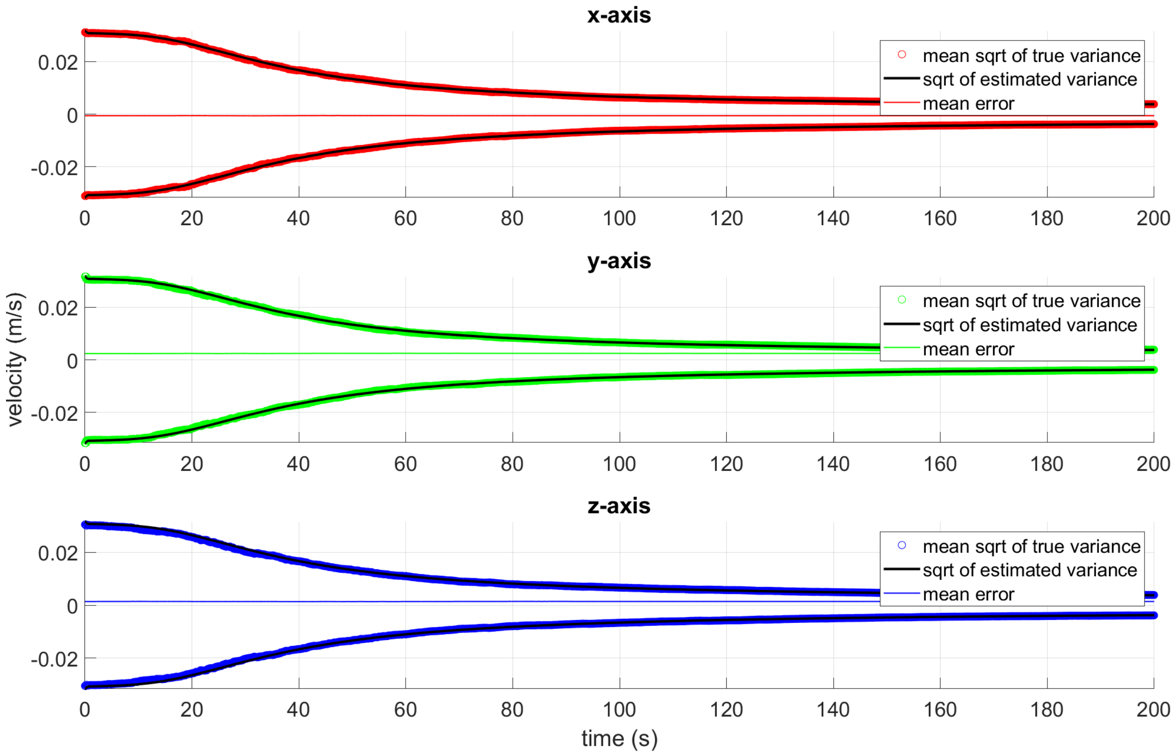


Figure 7.81: Monte-Carlo GPS Velocity Bias

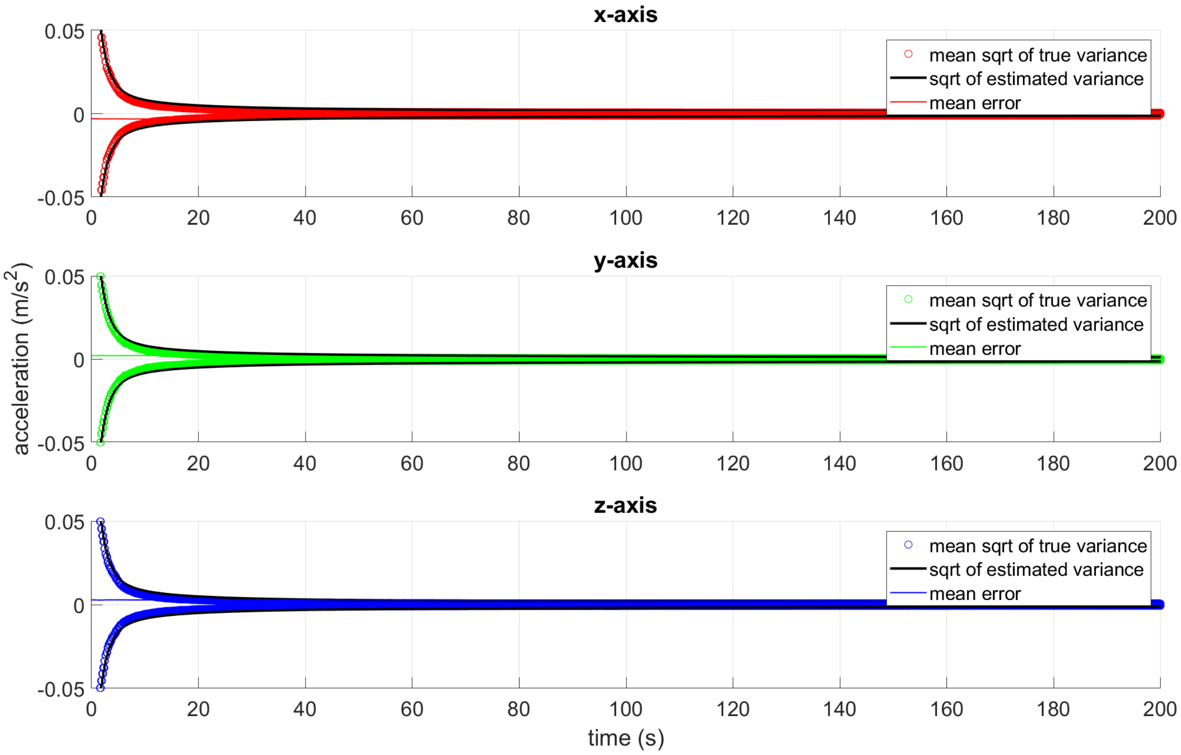


Figure 7.82: Monte-Carlo Accelerometer Bias

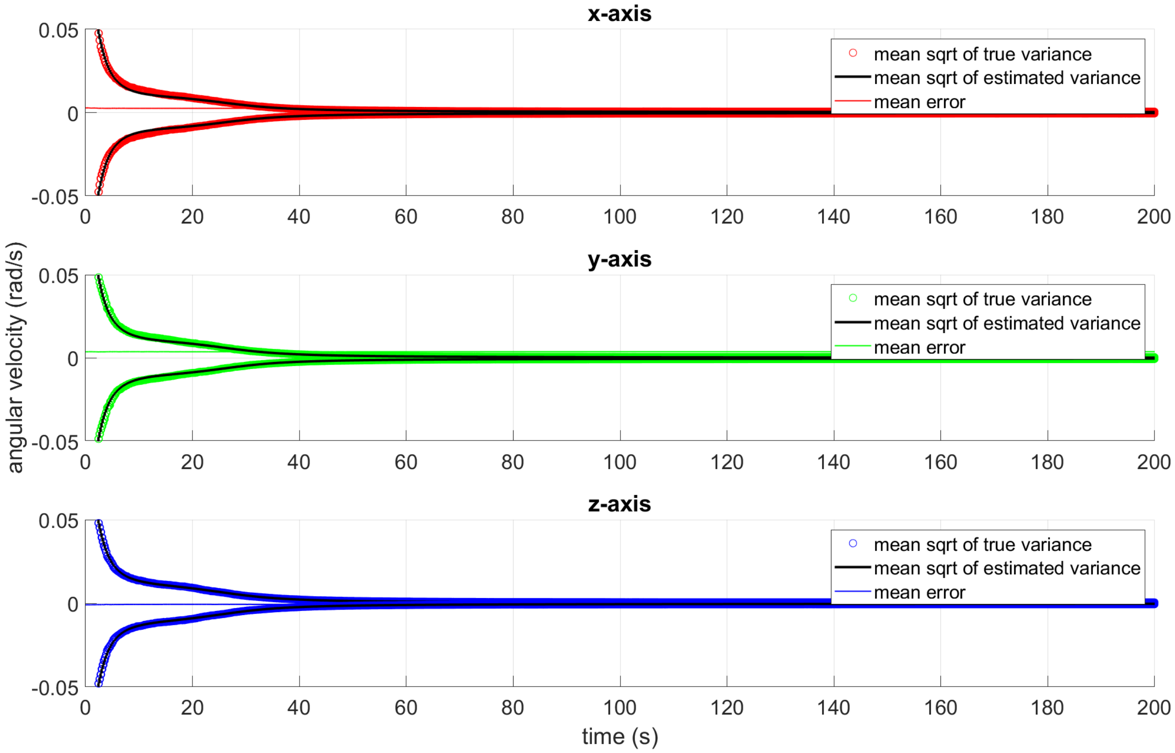


Figure 7.83: Monte-Carlo Gyroscope Bias

Chapter 8: Conclusion

8.1 Summary of Results

This investigation shows that utilizing radically inexpensive MEMS technology as a substitution for high-precision attitude sensors and mathematically complex models is viable for most smallsat missions. Given the satellite constraints, we were able to compute attitude measurements with an uncertainty of ±10°. Given the mission requirement to maintain a sun-pointing attitude within ±20°, a ±10° is certainly within that range, but may not provide good feedback to the attitude controller. To minimize this uncertainty we derived our MEKF and reduced the overall uncertainty to below ±5° while confirming that the results seen in our simulations represent reality through a high fidelity performance analysis of the MEKF. The Monte Carlo simulation shows that the Kalman filter estimated uncertainty matches the true estimation error when using different initial conditions. The error budget analysis revealed that the accelerometer and gyroscope systematic errors excluding bias showed very little contribution to the overall uncertainty of position, velocity, and attitude. Following the error budget with a sensitivity analysis it is shown that the position, velocity, and attitude were not sensitive to the scaling of these error groups. Therefore, we justifiably removed these terms from the state model and significantly simplified the complexity of our system. Initially, the optimal system had 42 elements. By removing the accelerometer and gyroscope non-orthogonality, misalignment, and scaling errors, we reduce the system to a sub-optimal 24 state filter. This allows much faster computation time and better real-time performance for a system that is already computationally constrained.

Through our single run results it is shown that there are unobservable states. The most significant of the unobservable states are the GPS and quaternion biases. While they do exhibit convergence in their estimated uncertainties, their uncertainties remain constant and relatively large; they also have large contributions to the overall estimates of position and attitude. The remaining unobservable states are the accelerometer and gyroscope systematic errors, excluding bias. However, since those terms were removed from the system due to their negligible contributions, the Kalman filter's inability to observe those states was not an issue. These error terms get "soaked up" by the bias term and since the bias for the accelerometer and gyroscope is observable, we can confidently remove these systematic error terms and, if necessary, adjust our tuning through the process noise to accommodate for the added uncertainty in our model.

The presence of unobservable errors such as the quaternion and GPS biases significantly downgrades the estimates of this particular Kalman filter, however, much can be done to minimize their effects. Since precise position estimation for our mission is not required, the GPS bias can be neglected for the purpose of our mission. The attitude bias significantly affects the attitude estimate if it is large enough and cannot be neglected. Since our derived attitude sensor measurement relies on magnetic field measurements, bias can come from soft/hard iron effects and alternating-current (AC) effects from the satellite power system. In this simulation, the bias uncertainty is small due to our strategic placement of the IMU which isolates the magnetometer from these sources of error. If our derived attitude measurements exhibit no measurement bias, then the attitude estimation accuracy can be further improved.

One way to improve accuracy is to minimize the magnetic interference or change our method of measuring attitude. Camera-based attitude sensors typically do not experience this significant bias and there is research currently being conducted on utilizing off the shelf cameras as star trackers [18]; of these cameras the Raspberry Pi camera is the one of most interest due to its size and compatibility with our system. The magnetometer may also be shielded against the AC effects if strategic placement is constrained, as well as the soft/hard iron effects calibrated to eliminate as much interference as possible, but a bias will still always be present. These calibration techniques may only decrease our uncertainty of the bias.

8.2 Future Work

Given the unobservability of the GPS and quaternion biases, future research may focus on including other sensors in the system. A relatively new field of research being pursued for navigation and communications missions is cubesat formation flying. If body-relative position and attitude measurements can be provided from another cubesat in a swarm of cubesats, then the biases can potentially be observed. This is a topic worth investigating, since formation flying of cubesats may require precise position estimation to safely avoid collision and allow for precise pointing of RF antennas and solar arrays.

Another method that can reduce both bias and noise magnitude is use of dual magnetometers. This is feasible for our system given the size of the IMU. Since the TRIAD algorithm utilizes two vector pair observations, two magnetometer pairs would suffice as long as they are mounted in different orientations. This would eliminate the need for $(\dot{\mathbf{b}}^{tr}, \dot{\mathbf{b}}^{ned})$ as the second pair along with the pre-processing of $\dot{\mathbf{b}}^{tr}$. Since $\dot{\mathbf{b}}^{tr}$ has a significantly higher error magnitude than $\mathbf{b}^{tr}$, the overall attitude uncertainty would become much smaller. Since the IMU uses i2c communication, an additional IMU would not be challenging to incorporate into the satellite system.

References

[1] cubesat.org, “cubesat.” https://www.cubesat.org/. Accessed: 2020-03-03.

[2] eoportal.org, “Prometheus cubesat missions.” https://directory.eoportal.org/web/eoportal/satellite-missions/p/prometheus#references. Accessed: 2020-03-02.

[3] eoportal.org, “Cubesat concept.” https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/cubesat-concept. Accessed: 2020-03-02.

[4] eoportal.org, “Satellite missions database.” https://directory.eoportal.org/web/eoportal/satellite-missions. Accessed: 2020-03-02.

[5] cubesat.org, “Developer resources.” https://www.cubesat.org/cubesatinfo. Accessed: 2020-03-02.

[6] cubesatshop.com, “Ku leuvun star tracker.” https://www.cubesatshop.com/product/kul-star-tracker/. Accessed: 2020-03-02.

[7] cubesatshop.com, “Mai-ss space sextant.” https://www.cubesatshop.com/product/mai-ss-space-sextant/. Accessed: 2020-03-02.

[8] cubesatshop.com, “Mai-ses ir earth sensor.” https://www.cubesatshop.com/product/mai-ses-ir-earth-sensor/. Accessed: 2020-03-02.

[9] cubesatshop.com, “Nss cubesat sun sensor.” https://www.cubesatshop.com/product/nss-cubesat-sun-sensor/. Accessed: 2020-03-02.

[10] cubesatshop.com, “Nano-ssoc-a60 analog sun sensor.” https://www.cubesatshop.com/product/nano-ssoc-a60-analog-sun-sensor/. Accessed: 2020-03-02.

[11] NOAA, "World magnetic model." https://www.ngdc.noaa.gov/geomag/WMM/, December 2019. Accessed: 2020-03-02.

[12] I. Y. Bar-Itzhack and R. R. Harman, "Optimized triad algorithm for attitude determination," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 208–211, 1997.

[13] G. Natanson, M. Challa, J. Deutschmann, and D. Baker, "Magnetometer-only attitude and rate determination for a gyro-less spacecraft," vol. -1, pp. 791–798, 10 1994.

[14] J. Craig, *Introduction to Robotics: Mechanics and Control*. Addison-Wesley series in electrical and computer engineering: control engineering, Pearson, 2005.

[15] D. J. Tylavsky and G. R. L. Sohie, "Generalization of the matrix inversion lemma," *Proceedings of the IEEE*, vol. 74, pp. 1050–1052, July 1986.

[16] digikey.com, "Lsm9ds1." https://www.digikey.com/catalog/en/partgroup/lsm9ds1/50138. Accessed: 2020-03-14.

[17] adafruit.com, "Adafruit ultimate gps." https://www.adafruit.com/product/746. Accessed: 2020-04-07.

[18] S. Pedrotty, R. Lovelace, J. Christian, D. Renshaw, and G. Quintero, "Design and performance of an open-source star tracker algorithm on commercial off-the-shelf cameras and computers," 02 2020.

[19] M. E. Davies, V. K. Abalakin, M. Bursa, J. H. Lieske, B. Morando, D. Morrison, P. K. Seidelmann, A. T. Sinclair, B. Yallop, and Y. S. Tjuflin, "Report of the iau/iag/cospar working group on cartographic coordinates and rotational elements of the planets and satellites: 1994," *Celestial Mechanics and Dynamical Astronomy*, vol. 63, pp. 127–148, Jun 1995.

[20] J. P. Hespanha, *Linear Systems Theory: Second Edition*. Princeton University Press, ned - new edition, 2 ed., 2018.

[21] D. Greenwood, *Principles of Dynamics*. Prentice Hall international series dynamics, Prentice-Hall, 1988.

[22] P. Antsaklis and A. Michel, *A Linear Systems Primer*. Birkhäuser Boston, 2007.

[23] H. Curtis, *Orbital Mechanics for Engineering Students*. Aerospace Engineering, Elsevier Science, 2013.

[24] J. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton paperbacks, Princeton University Press, 1999.

[25] C. Meyer, S. for Industrial, and A. Mathematics, *Matrix Analysis and Applied Linear Algebra*. Matrix Analysis and Applied Linear Algebra, Society for Industrial and Applied Mathematics, 2000.

[26] A. S. C. T. Staff, A. Gelb, and A. S. Corporation, *Applied Optimal Estimation*. Mit Press, MIT Press, 1974.

[27] P. J. Jorgensen, "Verification of a dual-state extended kalman filter with lidar-enabled autonomous hazard-detection for planetary landers," 2015.

[28] E. LEFFERTS, L. Markley, and M. SHUSTER, "Kalman filtering for spacecraft attitude estimation," 01 1982.

[29] M. Grewal and A. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*. Wiley - IEEE, Wiley, 2014.

Appendix A: Nomenclature

AC	Alternating Current
ADCS	Attitude Determination and Control System
DC	Direct Current
ECEF	Earth-Centered Earth Fixed
ECI	Earth-Centered Inertial
EKF	Extended Kalman Filter
GPS	Global Positioning System
GNC	Guidance Navigation and Controls
IMU	Inertial Measurement Unit
MEKF	Multiplicative Extended Kalman Filter
MEMS	Micro Electro-Mechanical System
NED	North-East-Down
NOAA	National Oceanic and Atmospheric Administration
PSD	Power Spectral Density
RK4	Runge-Kutta 4′th Order
RSS	Root Sum Square
WMM	World Magnetic Model

Appendix B: Sensitivity Analysis of Individual Error Groups

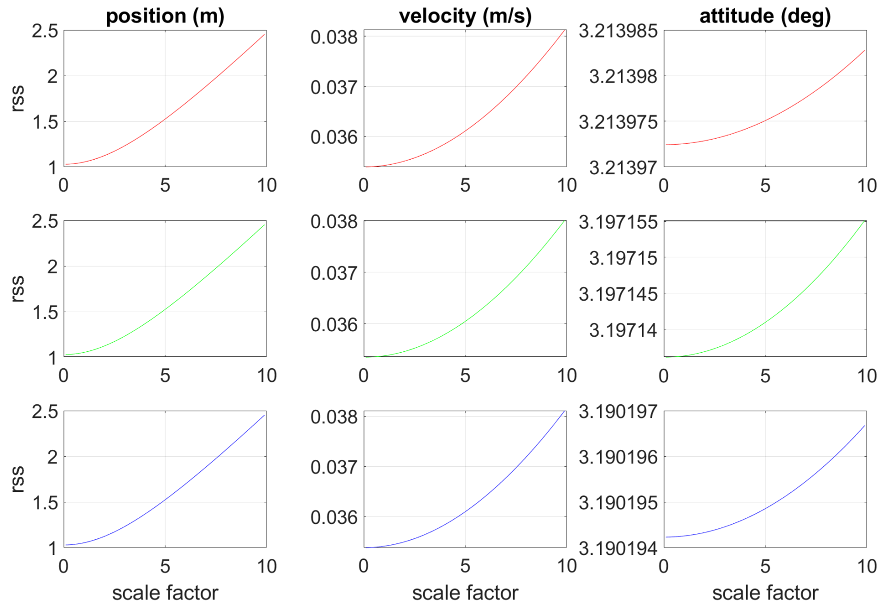


Figure B.1: Error Group 1 Sensitivity

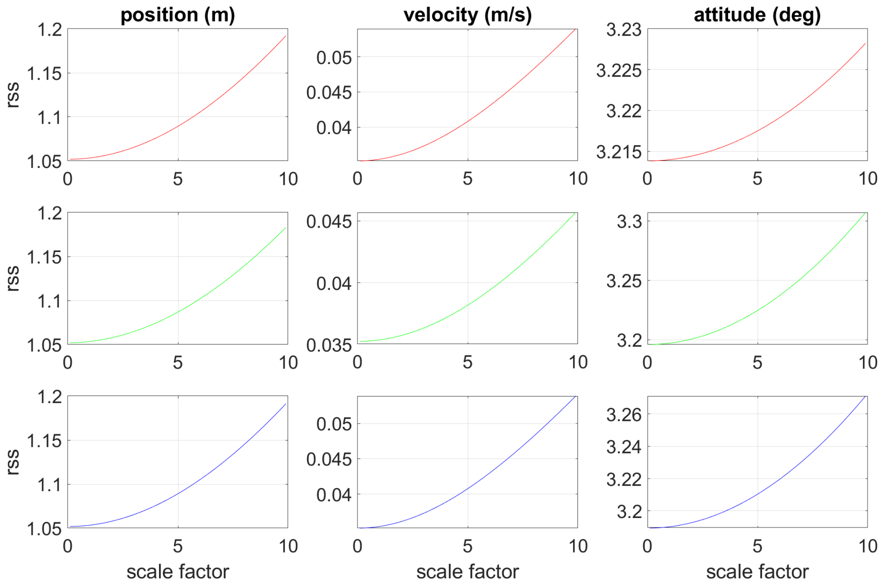


Figure B.2: Error Group 2 Sensitivity

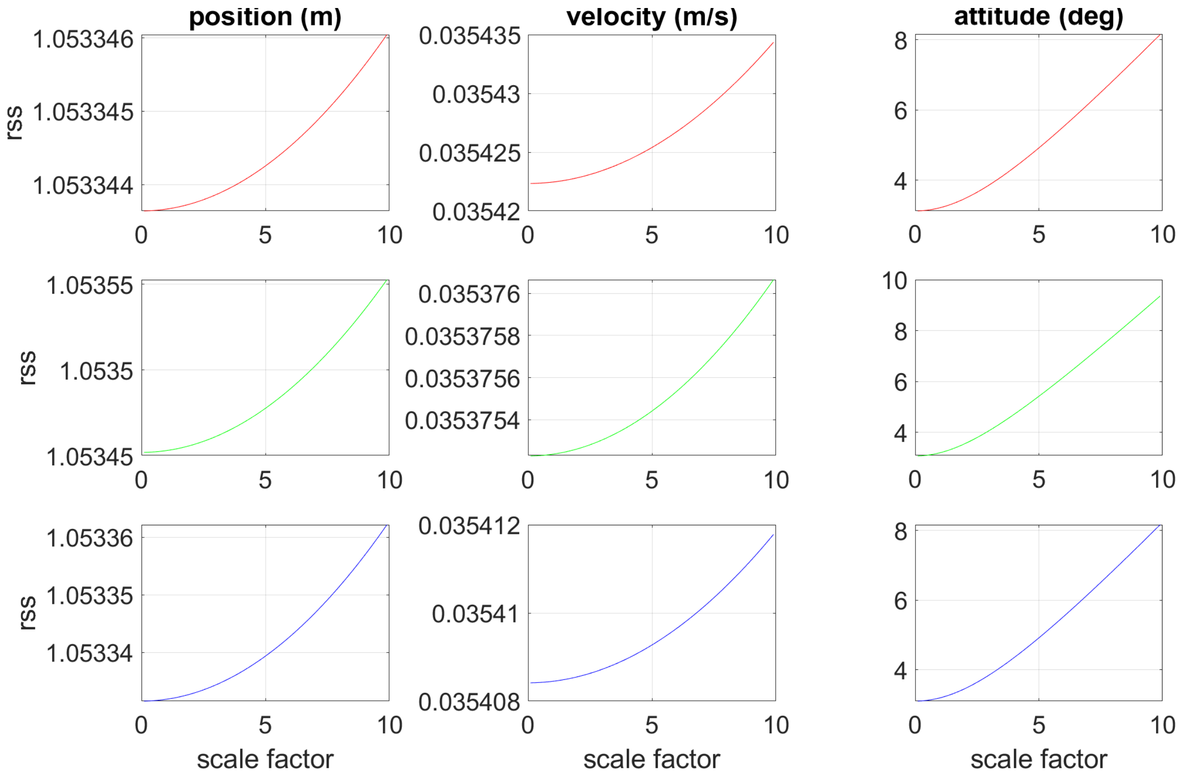


Figure B.3: Error Group 3 Sensitivity

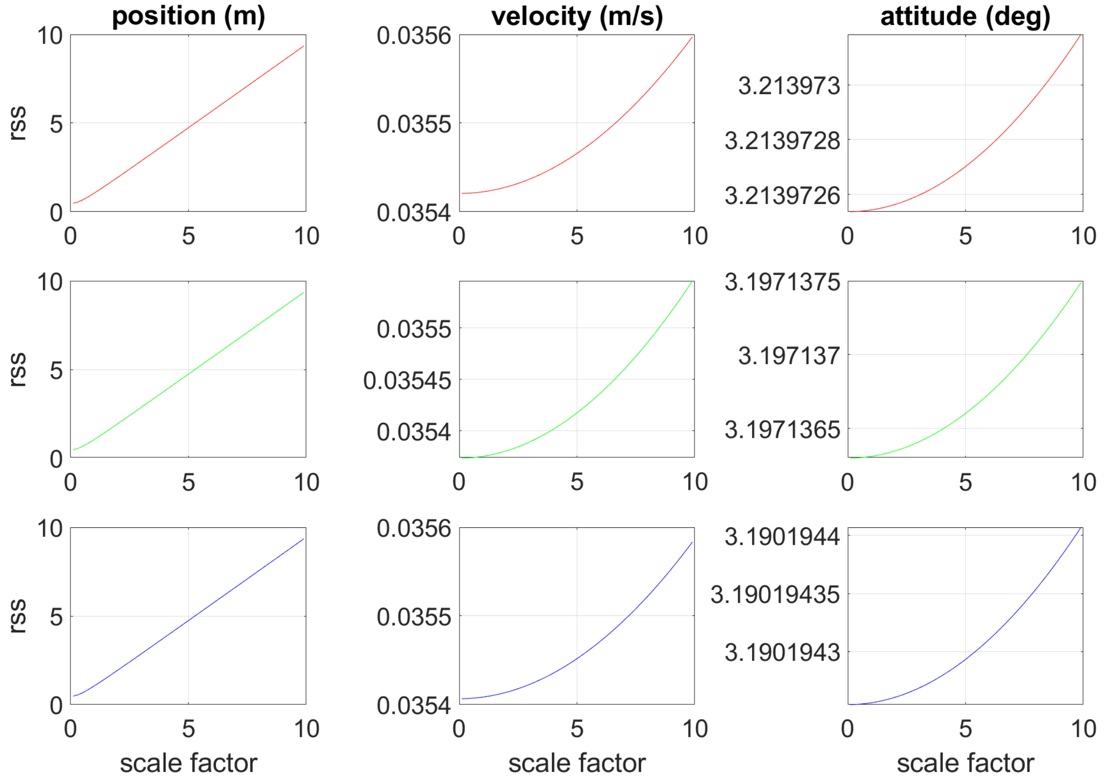


Figure B.4: Error Group 4 Sensitivity

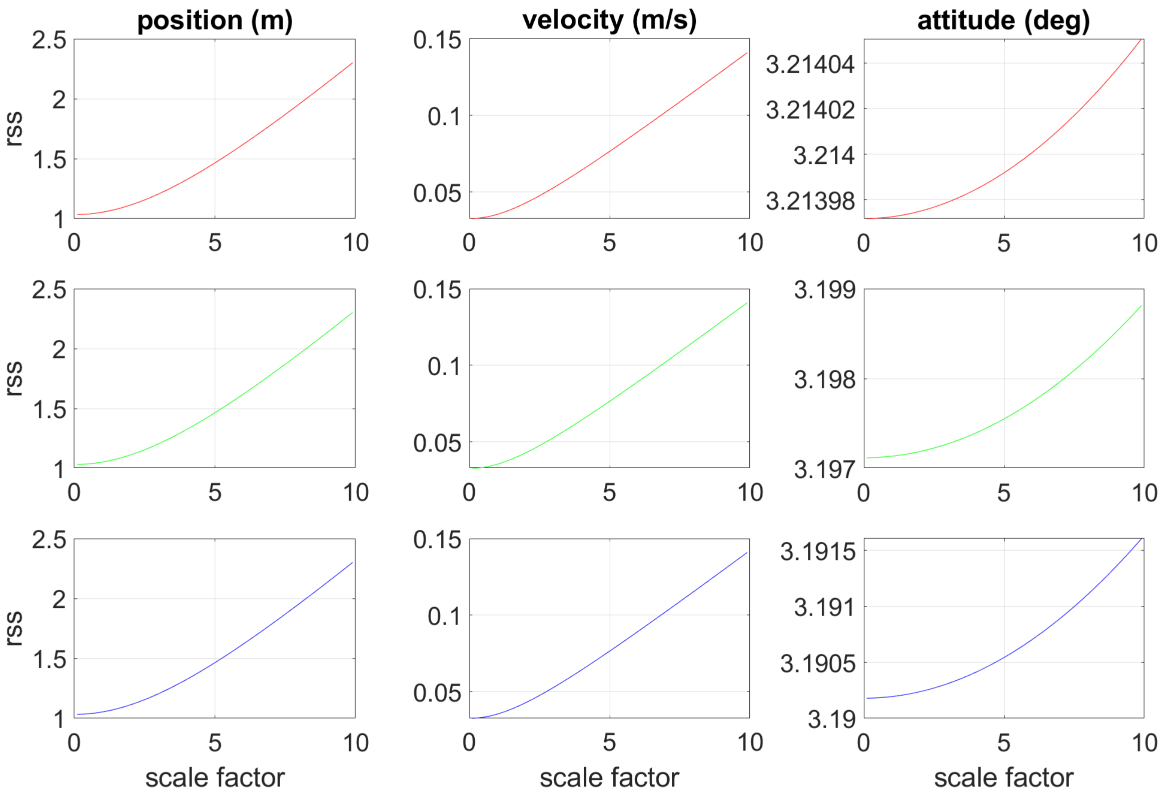


Figure B.5: Error Group 5 Sensitivity

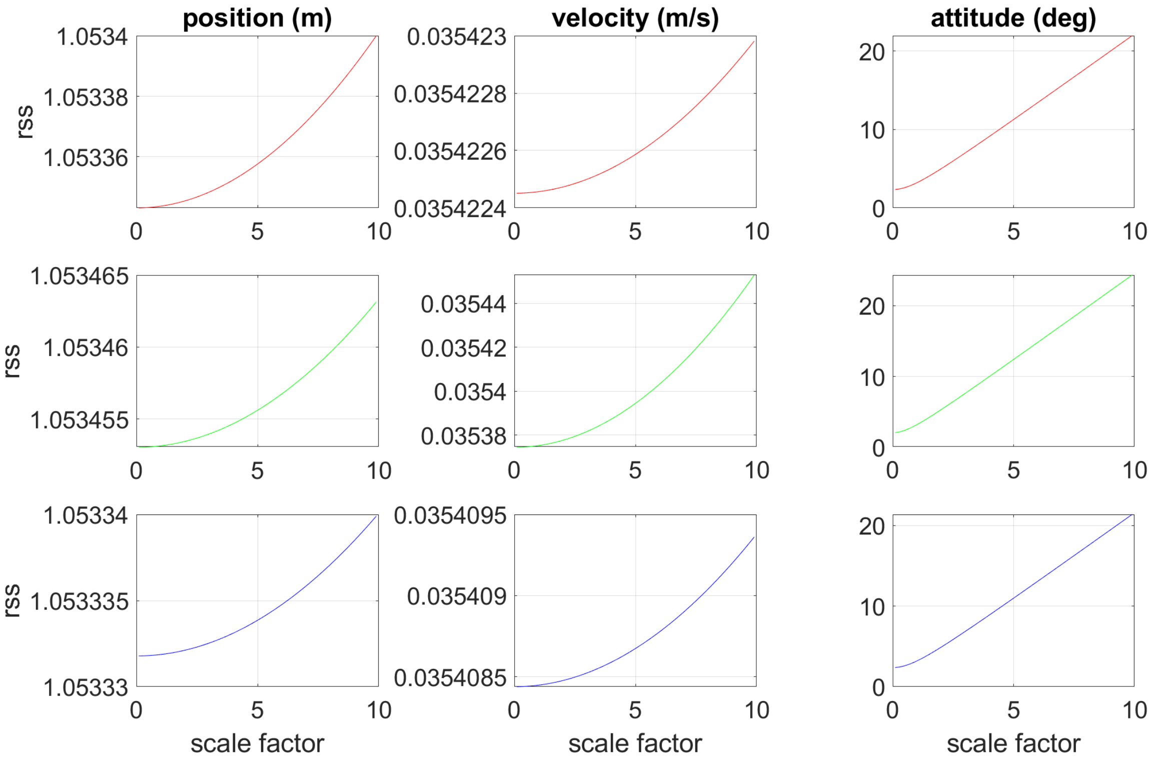


Figure B.6: Error Group 6 Sensitivity

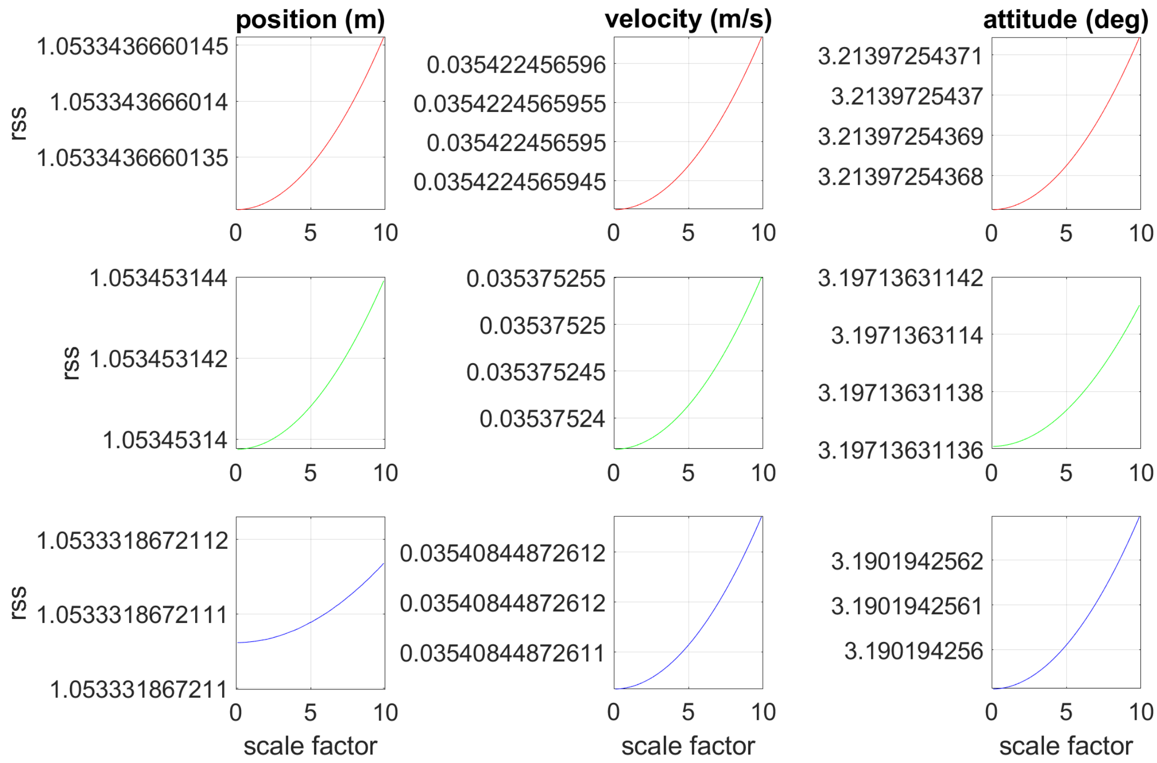


Figure B.7: Error Group 7 Sensitivity

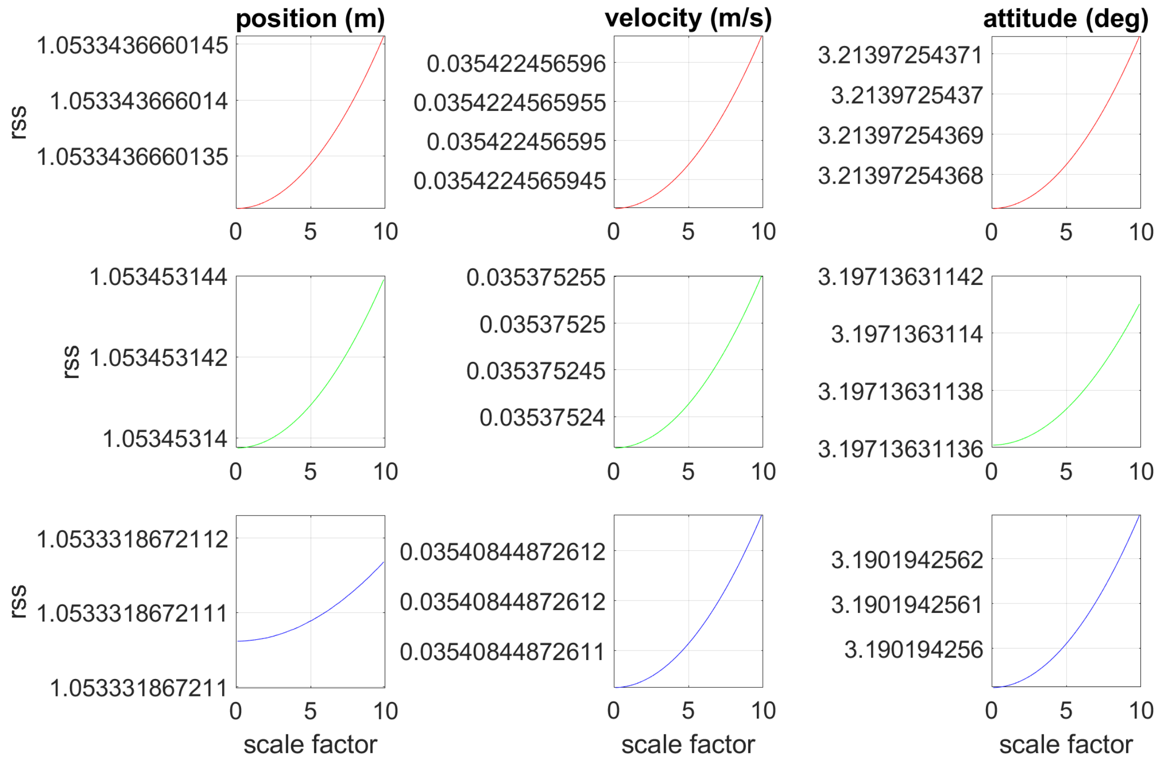


Figure B.8: Error Group 8 Sensitivity

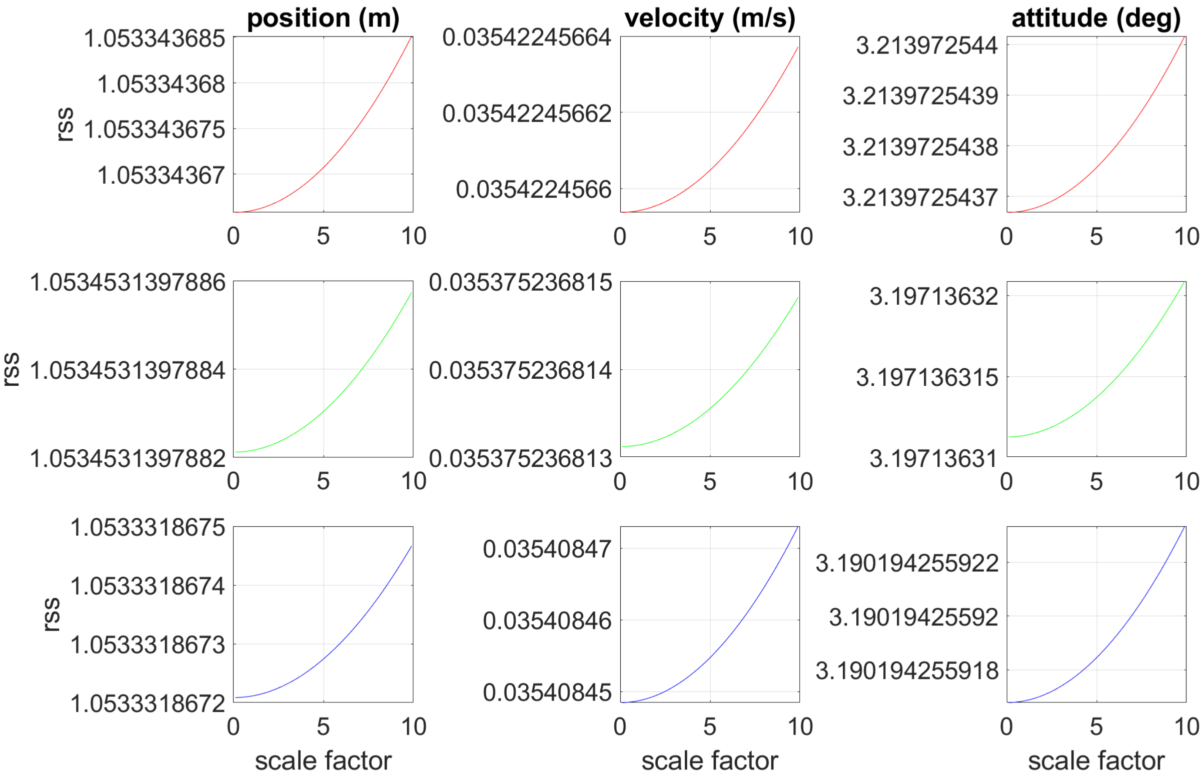


Figure B.9: Error Group 9 Sensitivity

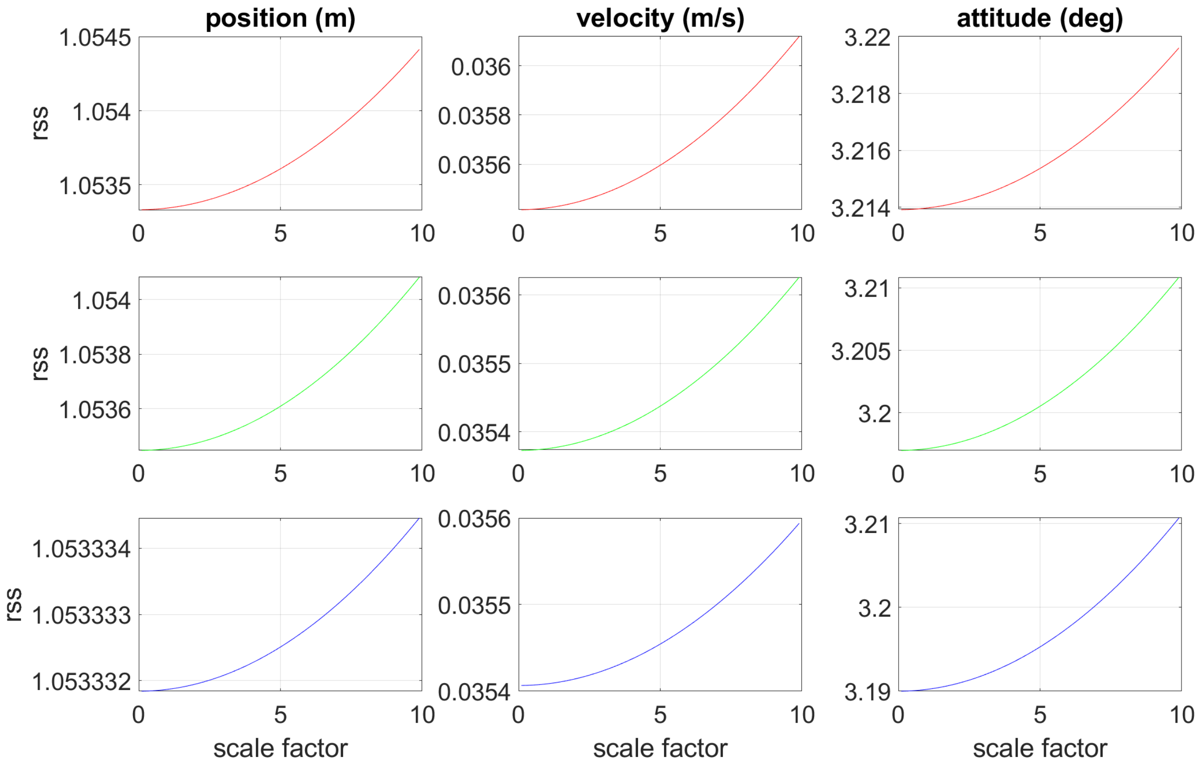


Figure B.10: Error Group 10 Sensitivity

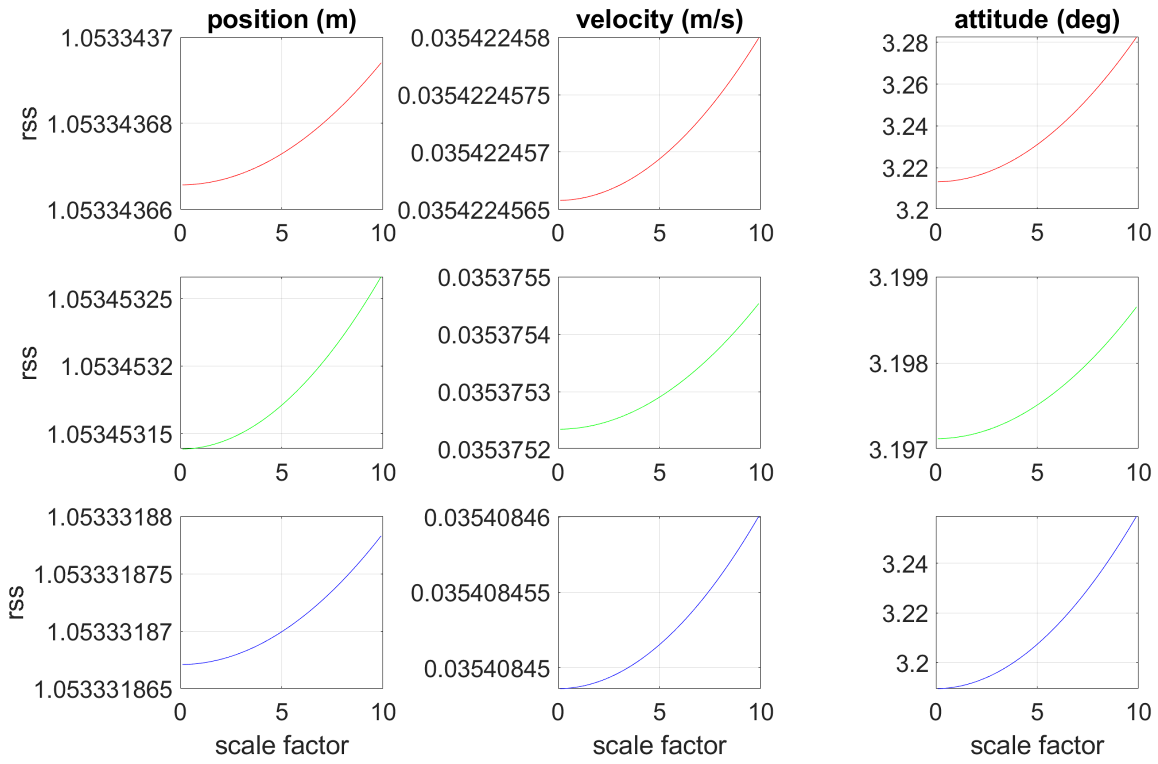


Figure B.11: Error Group 11 Sensitivity

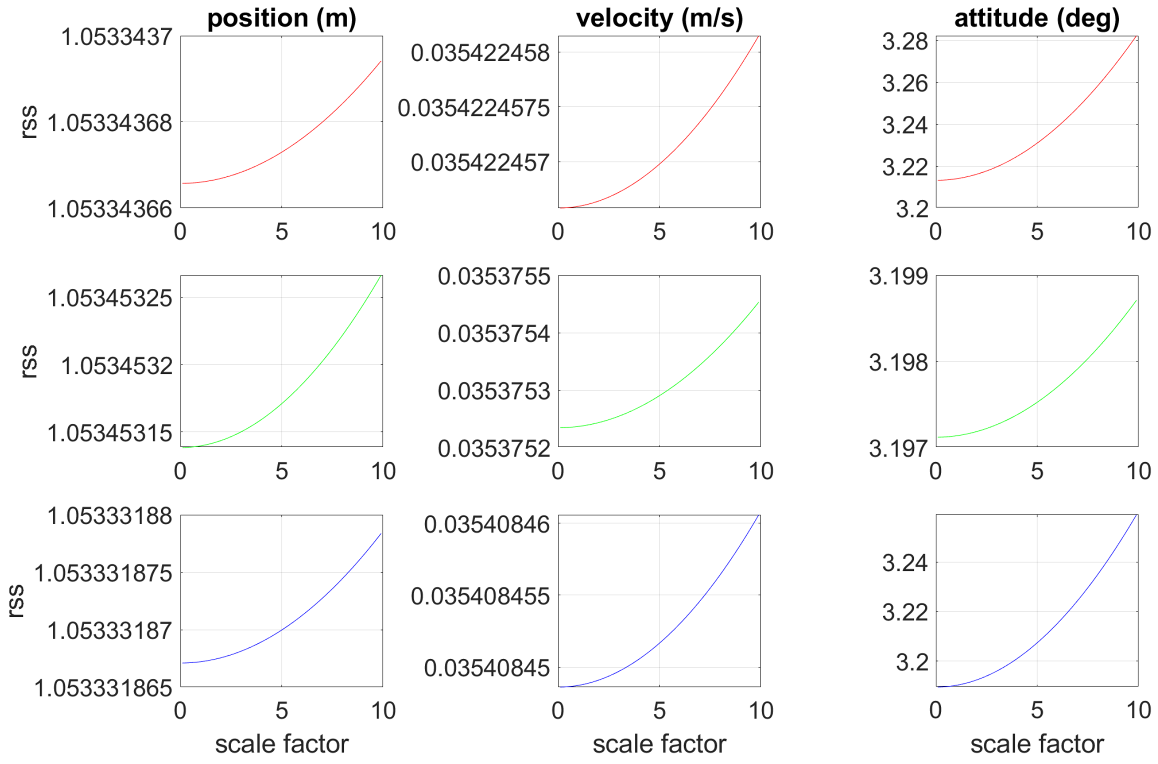


Figure B.12: Error Group 12 Sensitivity

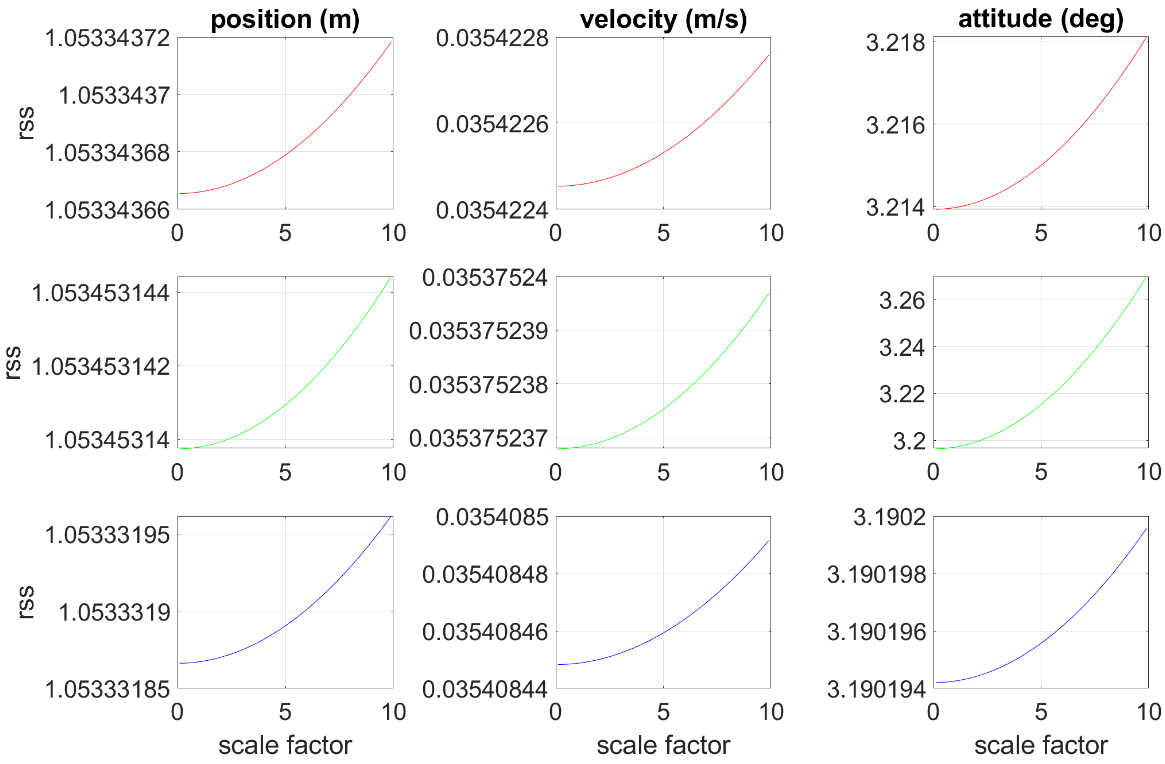


Figure B.13: Error Group 13 Sensitivity

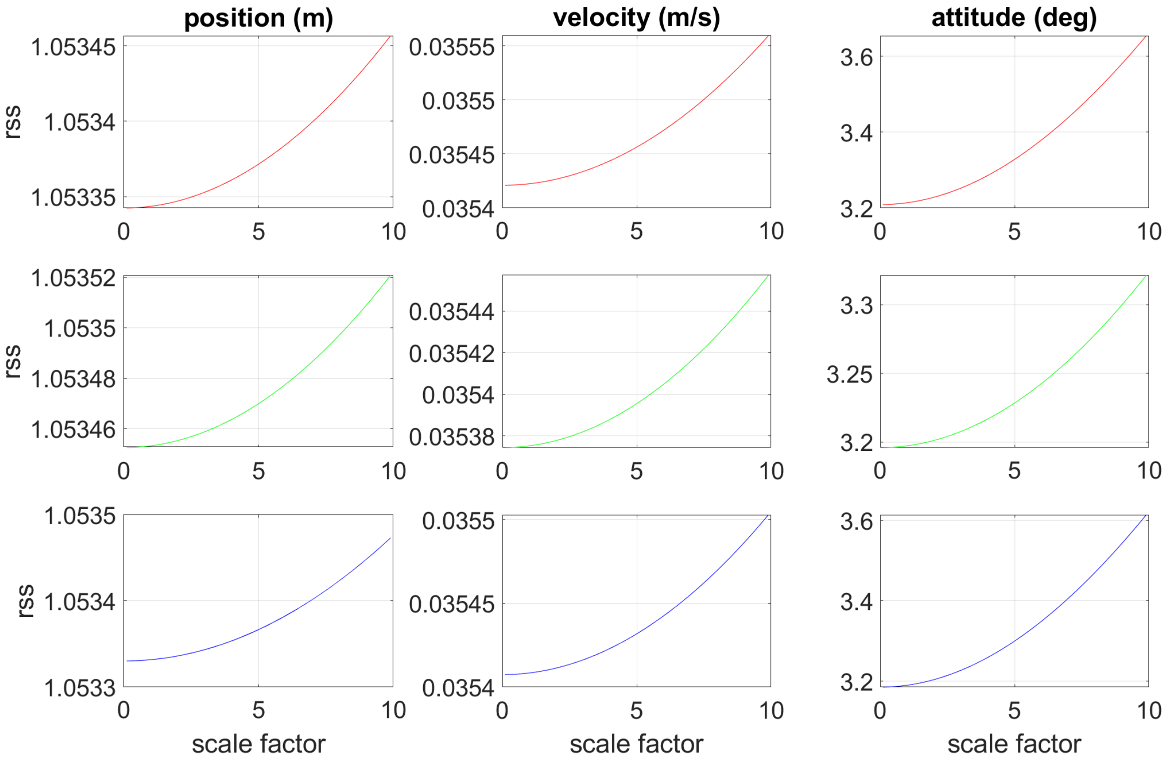


Figure B.14: Error Group 14 Sensitivity

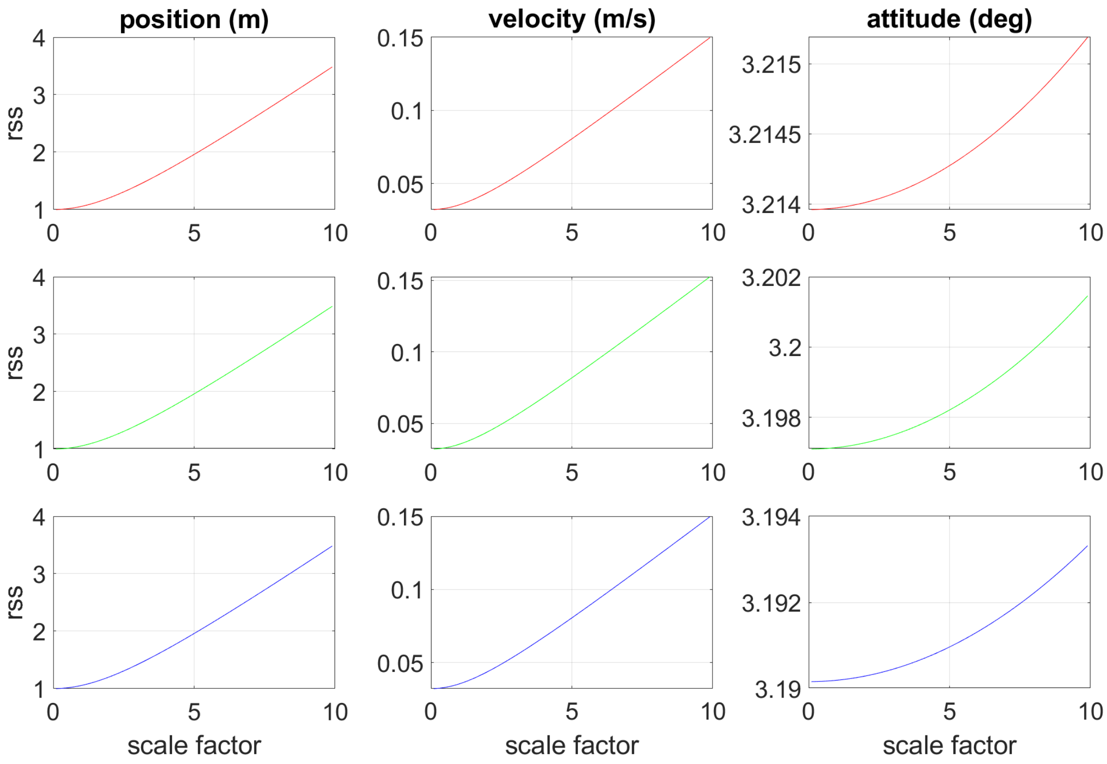


Figure B.15: Error Group 15 Sensitivity

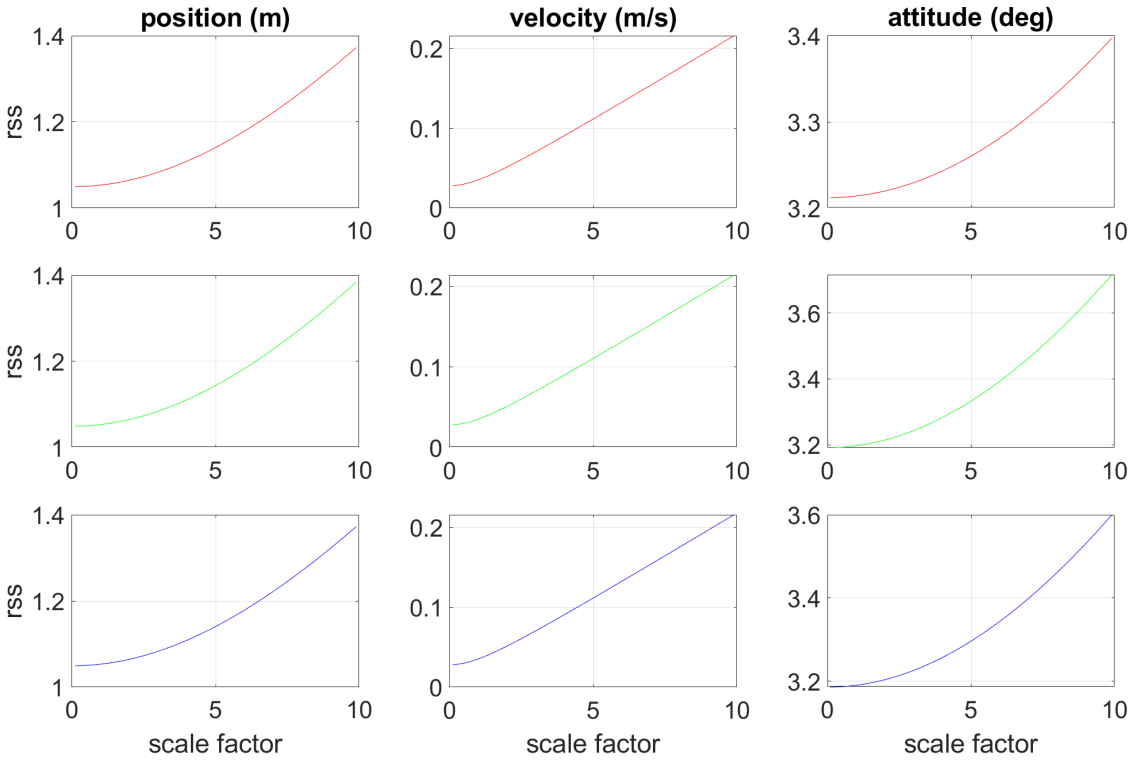


Figure B.16: Error Group 16 Sensitivity

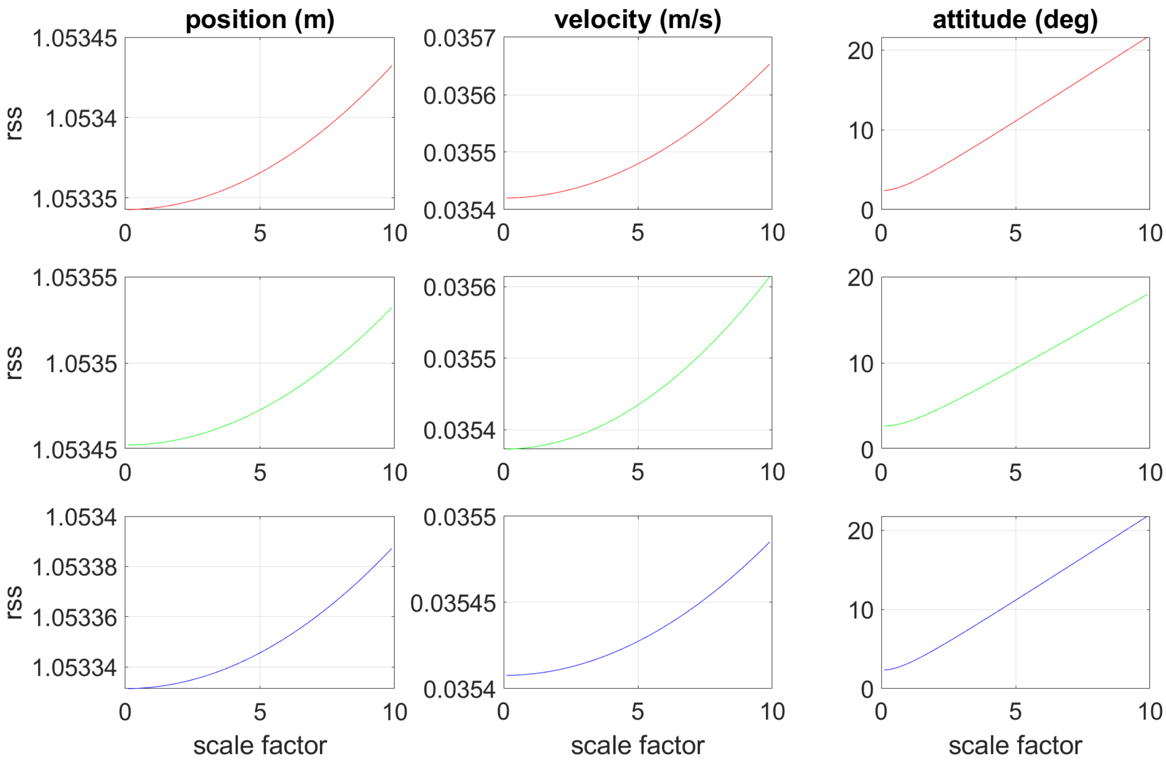


Figure B.17: Error Group 17 Sensitivity

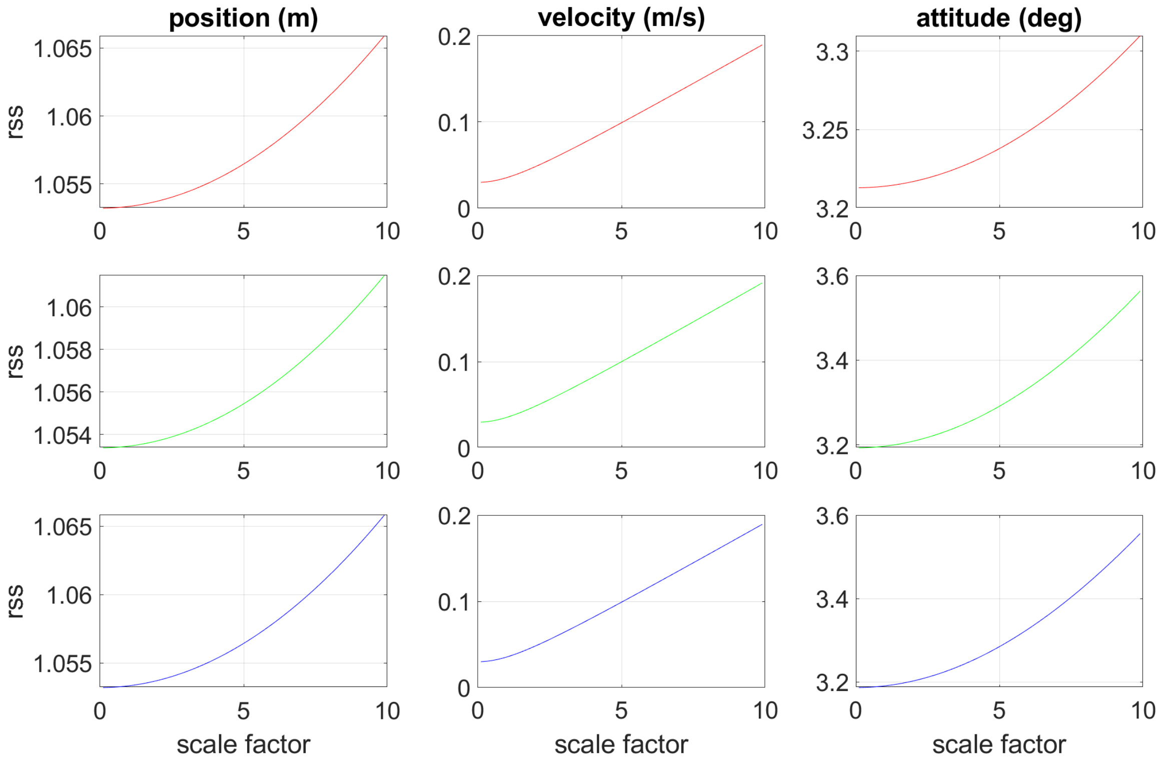


Figure B.18: Error Group 18 Sensitivity

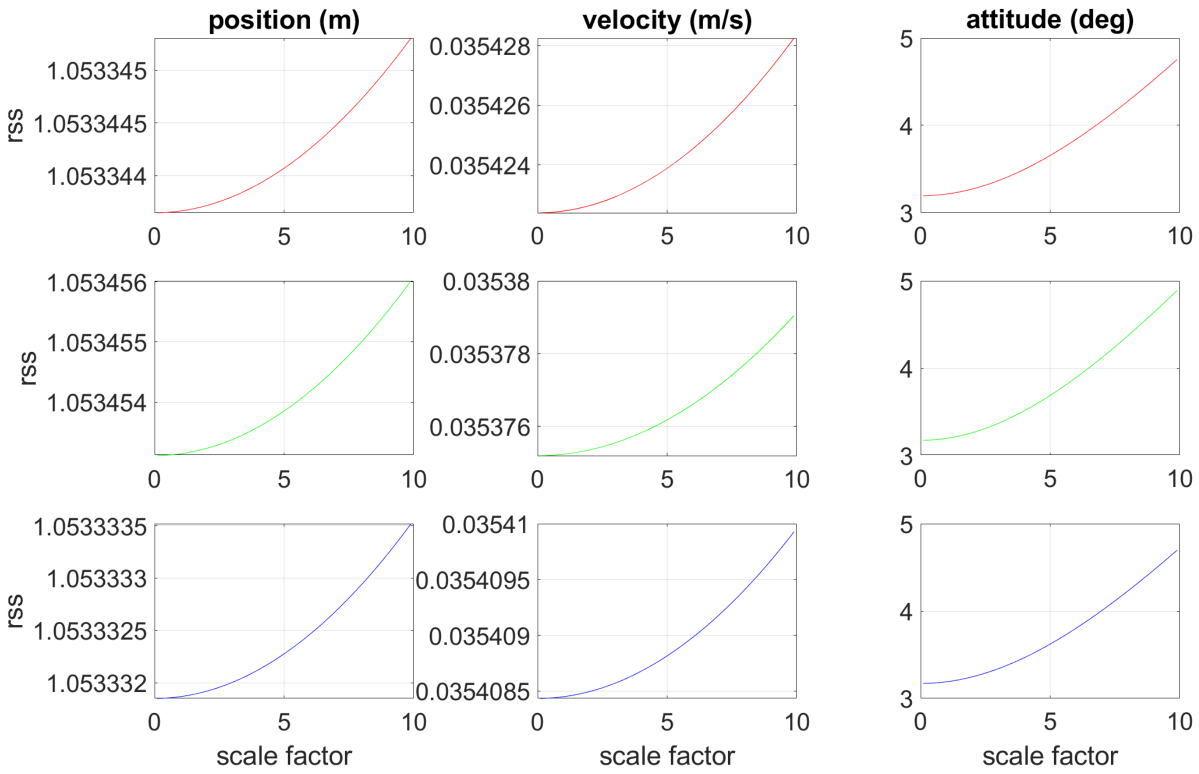


Figure B.19: Error Group 19 Sensitivity

Appendix C: Attitude Measurement Error Analysis

In Chapter 5 attitude measurements were computed using the TRIAD algorithm with the two vector pairs being the magnetic field $\mathbf{b}$ and the rate of change of the magnetic field with respect to time due to the velocity of the satellite $\dot{\mathbf{b}}$. Each vector pair represents the data in the NED frame and the IMU case frame. To obtain $\mathbf{b}^{ned}$, GPS measurements are used as an input to the WMM, along with the date and time. To compute $\dot{\mathbf{b}}^{ned}$, we make a finite difference approximation between two sequential measurements of $\mathbf{b}^{ned}$. The IMU can measure $\mathbf{b}^{tr}$ while $\dot{\mathbf{b}}^{tr}$ is also estimated through a finite difference approximation. Now we will discuss how this method of computing attitude affects the overall uncertainty of our derived attitude measurements and why pre-processing is required.

The uncertainty in $\mathbf{b}^{ned}$ comes from two sources of error: the WMM uncertainty and the GPS position uncertainty. Since the WMM uses GPS position as an input and returns the magnetic field at that point, any uncertainty in position will result in an uncertainty in the magnetic field (assuming the WMM is perfect). However, the WMM also comes with uncertainty that is well documented by the National Oceanic and Atmospheric Administration (NOAA) [11]. The uncertainty in $\mathbf{b}^{tr}$ comes from the measurement noise and is the largest contribution of error to our derived attitude measurements. Since $\dot{\mathbf{b}}^{tr}$ is computed using a finite difference approximation, any significant change due to noise will be seen as an actual change in the magnetic field. Due to this issue, we employ moving average filters for pre-processing $\mathbf{b}^{tr}$ and $\dot{\mathbf{b}}^{c}$. To determine an appropriate batch size n, we conduct an orbital survey of the magnetic field and rate of change of the magnetic field due to the orbital velocity.

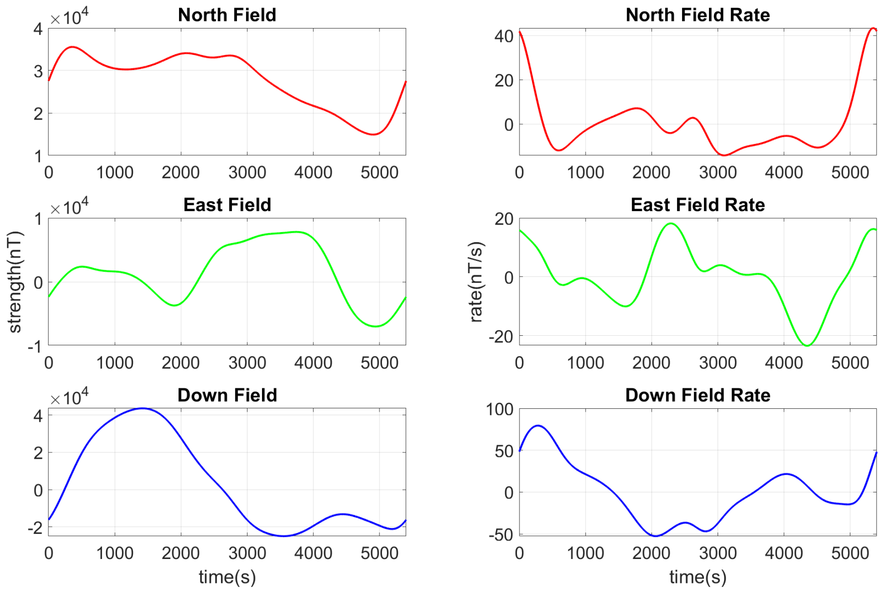


Figure C.1: Orbital Survey of Earth's Magnetic Field

From figure C.1 we can see that the magnetic field changes slowly over time with peaks near $75nT/s$. This allows us to use a large batch size for the moving average filter (assuming a non-rotating IMU). The moving average filter is given by

$$\bar{\mathbf{x}}_k = \bar{\mathbf{x}}_{k-1} + \frac{\mathbf{x}_k - \mathbf{x}_{k-n}}{n}\,, \tag{C.1}$$

and applying a batch size of $n = 100$, we obtain the following results shown in figure C.2

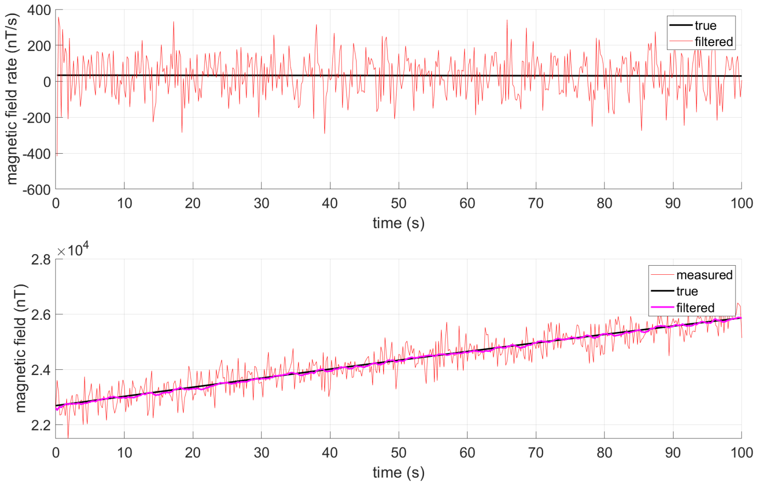


Figure C.2: Pre-Processed Data using Moving Average Filter with $n = 100$

While the error in $\mathbf{b}^{tr}$ has been significantly reduced, $\dot{\mathbf{b}}^{tr}$ still requires further error reduction. It is important to note that the small angle assumption plays a significant role in the tuning of our moving average filters since the magnitude of the attitude noise cannot be too large otherwise it will invalidate the small angle assumption made in earlier chapters. Therefore, we attempt to minimize the error as much as we can. Applying a much larger batch size of $n = 1000$ we obtain the following results shown in figure C.3.

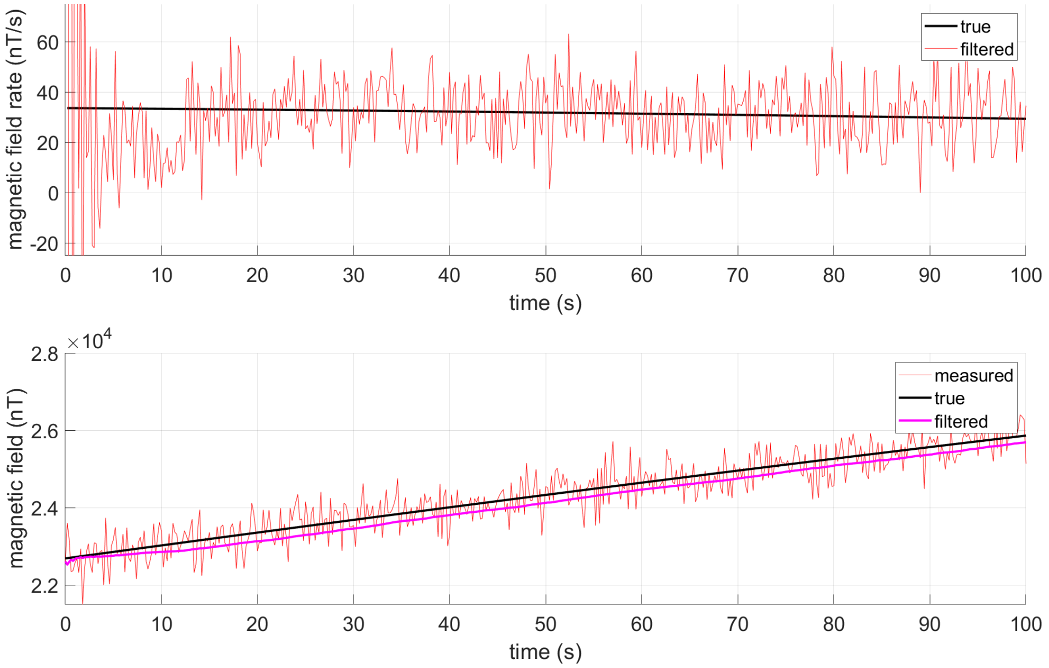


Figure C.3: Pre-Processed Data using Moving Average Filter with $n = 1000$

We see that $\dot{\mathbf{b}}^{tr}$ is significantly reduced, however, we have now introduced a bias in $\mathbf{b}^{tr}$. This bias is introduced because the data is changing too fast for the batch size and will directly influence the derived attitude measurements, therefore, we will use both batch sizes simultaneously. When $\mathbf{b}^{tr}$ is used for the first TRIAD pair it will be processed through the $n = 100$ filter and when $\dot{\mathbf{b}}^{tr}$ is used for the second TRIAD pair $\mathbf{b}^{tr}$ will be processed through the $n = 1000$ filter. This achieves the results seen in figure C.4. With the contribution to attitude error from the GPS, WMM, and IMU, we compute a maximum attitude uncertainty of $\pm 10^{\circ}$, shown in figure C.5.

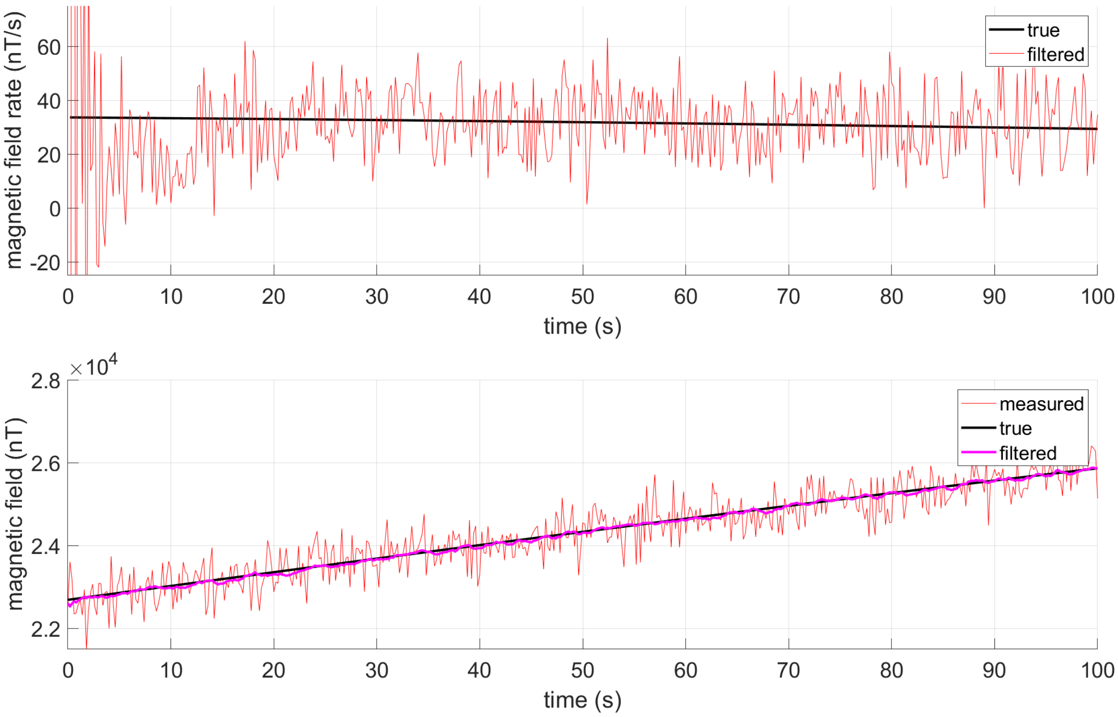


Figure C.4: Pre-Processed Data using Simultaneous Batch Sizes

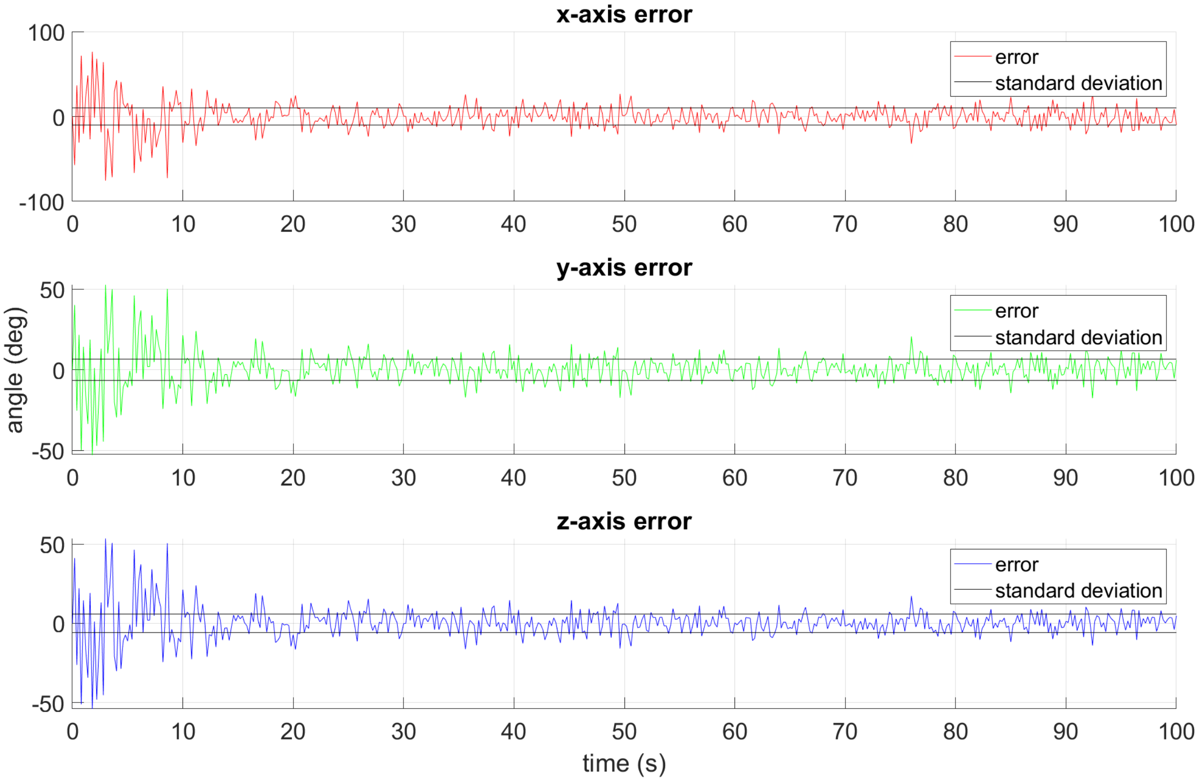


Figure C.5: Overall Attitude Measurement Uncertainty

It should be noted that due to the nature of the moving average filter the optimal results shown are achieved after the sample count approaches the largest batch size. The magnetometer is sampled at 80Hz and measurements are taken at 5Hz for attitude determination. Now we consider the scenario when the IMU is rotating due to the initial tumbling phase after ejection of the smallsat. Since the magnetic field is changing at a much more rapid rate, we must now modify our moving average filter to account for these rapid changes, otherwise the data will become biased or delayed as seen in figures C.6 and C.7

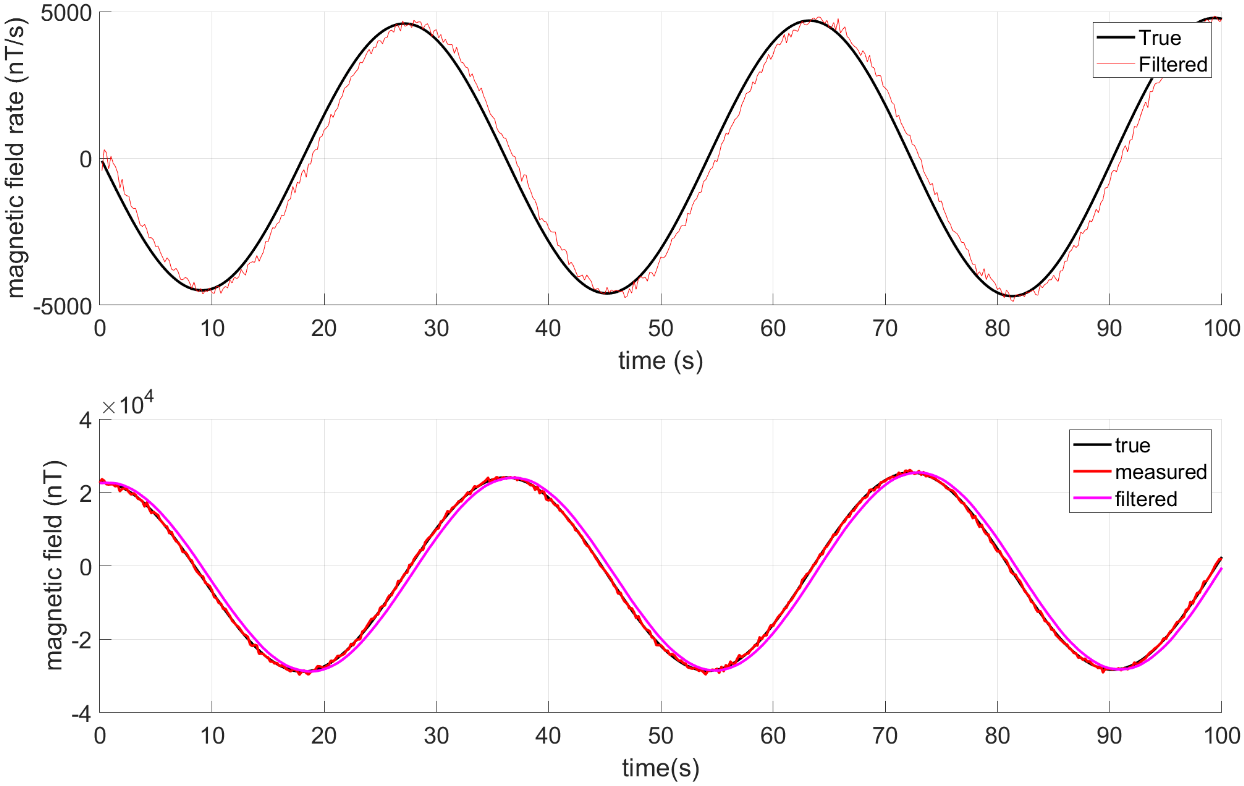


Figure C.6: Pre-Processed Rotating Data using Moving Average

To allow more accurate pre-processing we introduce a first order low-pass filter which takes the form

$$\bar{\mathbf{x}}_k = \alpha \mathbf{x}_k + (1 - \alpha)\bar{\mathbf{x}}_{k-1} \, . \tag{C.2}$$

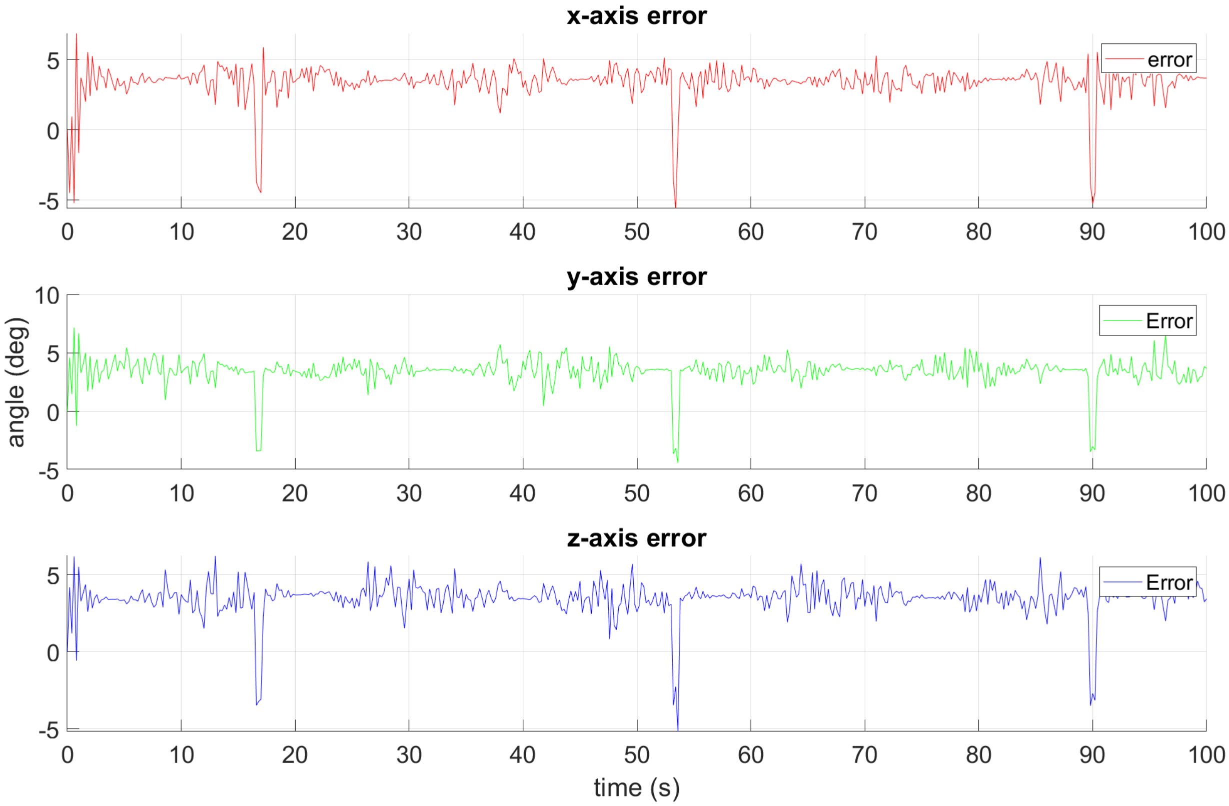


Figure C.7: Attitude Measurement Uncertainty from Moving Average Filter

Applying Eq. C.2 yields the results shown in figures C.8 and C.9 for a rotating body with rapidly changing data and $\alpha = 0.22$.

While both scenarios of spacecraft angular velocity are investigated, the realistic scenario does not involve an angular velocity of zero, therefore, the low-pass filter will be utilized at all time. It is also seen that spinning of the spacecraft yields a more accurate attitude measurement than the stabilized scenario. The scalar α is determined through tuning depending on the angular rate of the satellite. A value of $\alpha = 0.22$ was chosen based on our assumed initial angular velocity of the spacecraft. For the filter simulations the lower accuracy of $\pm 10°$ is used.

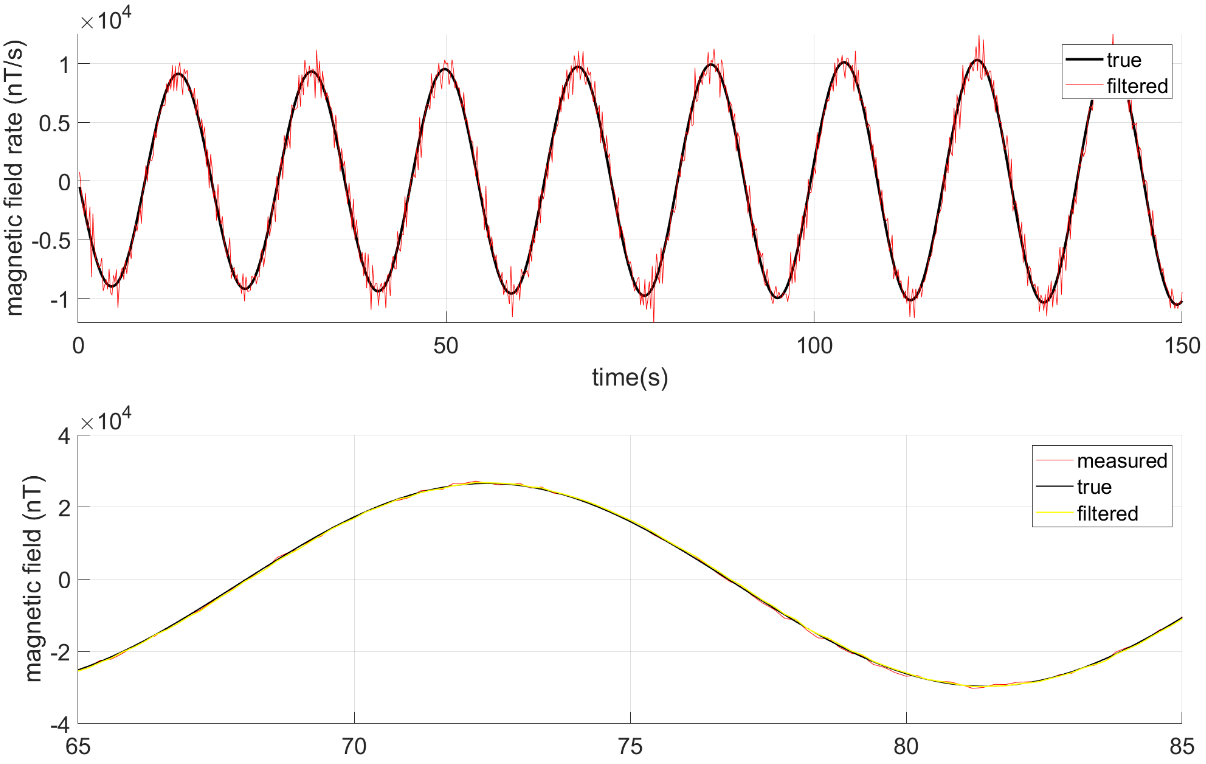


Figure C.8: Pre-Processed Rotating Data using Low-Pass Filter

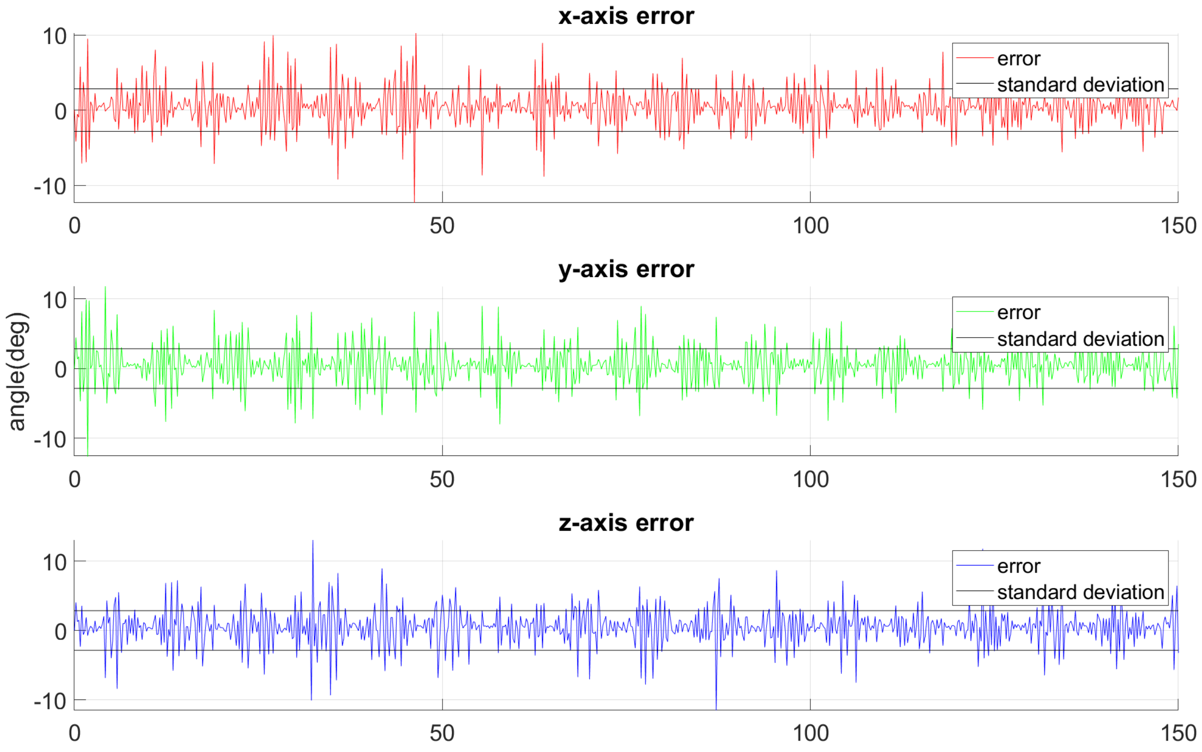


Figure C.9: Attitude Measurement Uncertainty from Low-Pass Filter