NTNU – Trondheim
Norwegian University of
Science and Technology

Model-free IMU-based DP State Estimator

Tor Eirik Østrem

Master of Science in Cybernetics and Robotics
Submission date: March 2014
Supervisor: Tor Arne Johansen, ITK
Co-supervisor: Thor Inge Fossen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology
Cybernetics

Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Engineering

MSC THESIS DESCRIPTION SHEET

Name:	Tor Eirik Østrem
Department:	Engineering Cybernetics
Thesis title:	Model-free IMU-based DP State Estimator

Thesis Description: Current model-based DP estimators are not optimal for handling large and unknown external forces, such as ice loads, cables, and similar. On the other hand, inertial measurement units (IMUs) are becoming more available at low cost and high quality, and can be used to complement model-based approaches. The overall objective is to develop a new model-free DP state estimator based on MEMS IMU technology, and evaluate its performance using a reference system.

Tasks:

1. Set up a logging system in collaboration with MT. Use ADIS IMU, and possibly others available, together with MT computer with data logging. Sensors, state estimator and reference system should be logged at suitable logging frequencies. The purpose of the data log is to validate the estimator.
2. Implement a model-free state estimator for position, velocity and attitude, using IMU, gyrocompass and position reference measurements. Use the method of Grip et al. or others. Evaluate the performance and compare to the reference system using logged data.
3. Implement the estimator in MATLAB/Simulink, and evaluate its performance in open loop. Compare with a model-based estimator in the presence of external forces.

Start date:	2013-09-16
Due date:	2014-03-02

Thesis performed at:	Department of Engineering Cybernetics, NTNU
Supervisor:	Professor Tor Arne Johansen, Dept. of Eng. Cybernetics, NTNU
Co-supervisor:	Professor Thor I. Fossen, Dept. of Eng. Cybernetics, NTNU

Abstract

The main goal of this thesis has been to develop a new model-free state estimator for dynamic positioning, based on an inertial measurement unit, a gyrocompass and a position reference system. The state estimator should provide estimates for position, velocity and attitude.

A literature study was given on different navigation systems and the associated sensors, where the nonlinear observer presented by Grip, Fossen, Johansen and Saberi has been described thoroughly. This model-free state estimator was modified to include a gyrocompass and to perform under varying position measurement rates. The modified nonlinear observer was tested both in simulations and experimental tests.

The simulations were realized with the use of MATLAB and Simulink, and was divided in different case studies. To begin with, two accelerometer bias estimation methods were compared to see which one was best suited. Secondly, the state estimator was tested during consistent and inconsistent position measurement rates. In the next case, the dead-reckoning capabilities of the nonlinear observer was tested under the presence of simulated noise and bias on the inertial measurement unit. Lastly, a comparison between the model-free state estimator and the model-based Kalman filter was carried out.

The experimental tests were realized in collaboration with Marine Technologies LLC in Egersund, Norway. The tests were conducted on a trolley using a MEMS based inertial measurement unit. The nonlinear observer was implemented on a computer, tested in different scenarioes, and compared with a reference system using a logging system. The tests were divided in three main case studies. From the first case, ideal sensor rates were considered. The second case tested the performance of the observer without gyrocompass measurements. Lastly, the performance of the observer without the position reference system was tested.

The modified nonlinear observer was found to provide satisfactory estimates, in both the simulations and the experimental tests. Due to a minor implementation problem in the experimental setup, the performance of the observer was at some degree degraded. Since the raw data from the sensors were logged, additional simulations were carried out with the use of MATLAB. These simulations gave good results. Because of sensor errors on the inertial measurement unit, the performance without aiding by the position reference system was degraded. On the other hand, the observer performed with great results in the absence of gyrocompass measurements.

Sammendrag

Hovedmålet med denne masteroppgaven har vært å utvikle en ny modellfri tilstandsestimator for dynamisk posisjonering, basert på en treghetsmåleenhet, et gyrokompass og et posisjonsreferansesystem. Tilstandsestimatoren skulle gi estimater for posisjon, hastighet og holdning.

Et litteraturstudie ble gitt for forskjellige navigasjonssystemer og tilhørende sensorer, hvor den ulineære observatøren presentert av Grip, Fossen, Johansen og Saberi har blitt grundig beskrevet. Denne modellfrie tilstandsestimatoren ble modifisert til å inkludere et gyrokompass og for å yte under varierende rater fra posisjonsreferansesystemet. Den modifiserte estimatoren ble testet både med simuleringer og eksperimentelle tester.

Simuleringene ble realisert med MATLAB og Simulink, og ble delt i forskjellig casestudier. To estimasjonsmetoder for akselerometerskjevhet ble testet til å begynne med. Dette ble gjort for å se hvilken metode som passet best. Tilstandsestimatoren ble så testet under konsistente og ukonsistente rater for posisjonsmålingene. Videre ble bestikknavigasjonsevnene til estimatoren testet med simulert støy og skjevhet på treghetsmåleenheten. Til slutt ble en sammenligning mellom den modellfrie tilstandsestimatoren og et modellbasert Kalman filter gjennomført.

De eksperimentelle testene ble realisert i samarbeid med Marine Technologies LLC i Egersund, Norge. Testene ble gjennomført på en tralle med bruk av en MEMS-basert treghetsmåleenhet. Tilstandsestimatoren ble implementert i en datamaskin, testet i ulike scenarier, og sammenlignet med et referansesystem ved bruk av et loggesystem. Testene ble delt i tre case-studier. I den første casen ble ideelle sensorrater ansett. Den andre casen testet ytelsen til estimatoren når gyrokompassmålinger var utilgjengelig. Til slutt ble ytelsen til tilstandsestimatoren testet under fravær av posisjonsreferansemålinger.

Den modifiserte tilstandsestimatoren ble funnet til å gi tilfredsstillende estimater, både under simuleringene og de eksperimentelle testene. På grunn av et lite implementasjonsproblem i de eksperimentelle testene ble ytelsen til estimatoren noe redusert. Logget rådata fra sensorene gjorde det mulig å utføre tilleggssimuleringer i MATLAB. Resultatene fra disse simuleringene ga gode resultater. Ytelsen til tilstandsestimatoren under bestikknavigasjon ble redusert på grunn av målefeil i treghetsmåleenheten. På den andre siden ga estimatoren gode resultater under fravær av gyrokompassmålinger.

Preface

This master's thesis concludes the Master of Science in Egineering Cybernetics program at the Norwegian University of Science and Technology, and was written during late fall and early spring of 2013 and 2014, respectively.

This report includes the necessary documentation for the results achieved in this thesis. In order to fully understand this report, general knowledge of mathematics, physics, signal processing and marine terminology will be useful.

I would like to thank my supervisor, professor Tor Arne Johansen and Thor Inge Fossen at the Department of Engineering Cybernetics for establishing this thesis in collaboration with Marine Technologies LLC. A special thanks goes to Tor Arne Johansen for his guidance whenever needed. Additional gratitude goes to Sveinung Tollefsen and Tore Flobakk for realizing the experimental tests in Egersund.

Furthermore, a big thanks goes to Håvard Hellvik, Thomas Kalve and Tore Havsø for the help during the experimental tests.

The biggest thanks goes to my mother, Tonny Pedersen, and my best friend, Christoffer Ofstad, for their support during this period. I would also like to thank my friends and family for additional support.

Through this thesis the following computer programs have been utilized:

- Mathworks MATLAB and Simulink R2013a
- Microsoft Visual Studio 2008

Tor Eirik Østrem

Trondheim, Norway

March 2, 2014

Contents

List of Tables

List of Figures

Nomenclature

Abbreviations

MT	Marine Technologies LLC
ECI	Earth-centered inertial reference frame
ECEF	Earth-centered Earth-fixed reference frame
LLA	Latitude, Longitude and Altitude coordinate system
NED	North-East-Down reference frame
BODY	Body-fixed reference frame
DOF	Degree of Freedom
DP	Dynamic Posistioning
RLG	Ring Laser Gyroscope
FOG	Fibre Optic Gyroscope
IMU	Inertial measurement unit
INS	Inertial Navigation System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GLONASS	Global'naya Navigatsionaya Sputnikovaya Sistema
DGNSS	Differential GNSS
HDOP	Horizontal Dillution of Precision
WGS84	World Geodetic System, 1984 (Earth's reference ellipsoid)
PVA	Position, Velocity and Attitude
CO	Center of origin
CG	Center of gravity
KF	Kalman filter
DR	Dead-Reckoning
JONSWAP	Joint North Sea Wave Project

SNR	Signal-to-Noise Ratio
MEMS	Micro-Electro-Mechanical Systems
VP	Vessel Parallel
ID	Identification
SOL	Speed of Light
GUI	Graphical User Interface
PoE	Power over Ethernet
FPGA	Field-Programmable Gate Array
CC	Control Computer

Lowercase

$\mathbf{a}$	Acceleration vector
$\mathbf{b}$	Bias vector
$\mathbf{c}$	Gyrocompass vector
$\mathbf{f}$	Specific force vector
$\mathbf{g}$	Gravity vector
$\mathbf{m}$	Magnetometer vector
$\mathbf{p}$	Position vector
$\mathbf{q}$	Unit quaternion
$\mathbf{r}$	Lever arm
$\bar{\mathbf{u}}$	Vector $\mathbf{u}$ given in quaternion notation
$\mathbf{v}$	Velocity vector
f_c	Cut-off frequency
h	Altitude
k_i	Observer gain
m	Lower norm
k	Discrete iteration step

Uppercase

$\mathbf{0}$	Matrix with zeros on all its entries
$\mathbf{C}$	Coriolis and centripetal matrix
$\mathbf{D}$	Damping matrix
$\mathbf{H}_K$	Error dynamic matrix
$\mathbf{I}$	Matrix with ones on the diagonal
$\mathbf{J}(\cdot)$	Matrix containing both the rotation and transformation matrices
$\mathbf{K}$	Observer gain matrix
$\mathbf{M}$	Mass matrix
$\mathbf{R}(\cdot)$	Rotation matrix
$\mathbf{S}(\cdot)$	Skew-symmetric matrix
$\mathbf{T}(\cdot)$	Transformation matrix
F	Discrete frequency
K	Discrete scale factor
M	Upper norm
T	Discrete time step

Greek

$\boldsymbol{\Theta}_{nb}$	Attitude given in Euler angles
$\boldsymbol{\xi}$	Main observer error
$\boldsymbol{\omega}$	Angular velocity vector
$\boldsymbol{\eta}$	Position and attitude vector
$\boldsymbol{\nu}$	Linear and angular velocity vector
$\boldsymbol{\tau}$	Force and moment vector
$\boldsymbol{\varepsilon}$	Real part vector in a unit quaternion
η	Imaginary part in a unit quaternion
ϕ	Roll angle

θ	Pitch angle
θ_{gnss}	Observer stability gain
ψ	Yaw angle
μ	Latitude
φ	Longitude
δ	Lower norm
ρ	Upper bound on the norm of the error dynamics

Chapter 1

Introduction

Model-based state estimators for dynamic positioning doesn't handle external forces optimally, i.e. ice loads and similar. On the contrary, inertial measurement units (IMUs) are becoming availible at a low cost with high accuracy. The integration of an IMU with a global navigation satellite system (GNSS) could complement the model based approach, resulting in a model-free state estimator.

1.1 Motivation

The most used navigation system for dynamic positioning systems today is the Kalman Filter (KF), [27] and [41]. The KF can provide position, velocity and attitude (PVA) estimates based on a stastical approach used to minimize the errors on the estimates. The nonlinear version of the KF is the Extended Kalman Filter (EKF), [16].

The KF is normally based on a model of the system it's going to navigate, which makes it dependent on the quality of the chosen model. Any deviations/noise in the model are filtered by the KF with appropriate tuning of the KF's design matrices, under the assumption that the process and measurement noise have zero mean and some variance. In addition, the KF rely on solving the computationally demanding Ricatti equation for the compution of KF gain, utilized on the system's injection term.

Since marine operations in the arctic is becoming an area with increasing interests, mostly due to petroleum extraction, the demand for precise navigation in higher latitudes are required, [2]. This is an area where ice is a problem for both control and navigation. Ice gathering around the hull of the ship would introduce an external force to the vessel, which would lead to errors in the KF approach, due to not being included in the model and incorrect design matrices for the new sea state.The KF also introduce a significant computational overhead, because of the Ricatti equation, and require time-consuming tuning of the design matrices.

The above reasons gives the motivation to develop a model-free state estimator for dynamic positioning, using an Inertial Measurement Unit (IMU) as a substitute for the model. This model-free approach can be realized with the nonlinear observer of Grip et al. [20], which uses nonlinear theory.

State estimators based on nonlinear theory give rise to several advantages over the KF estimation approach: The computational overhead is greatly reduced due to the removal of the Ricatti equation. Since the number of differential equations to solve is reduced, ta more maintainable code is achieved, less prone to faulty implementation. Using nonlinear techniques, a state estimator based on nonlinear theory can be proven to have global stabilty [29], unlike the EKF [13].

The nonlinear observer presented in [20] exploited measurements from an IMU to achieve rotational and translational motion on the system it was mounted on, i.e. the specific force and angular velocity measurements. These quantities could be integrated once and twice to acquire the PVA estimates. As the IMU directly measures the forces inflicted on the system, the nonlinear observer would be able to handle unknown forces, such as ice.

1.2 Theory Basis

The development of a model-free nonlinear observer for dynamic positioning requires insight in the following diciplines: Mathematics, physics, signal processing and marine terminology.

The kinematics and kinetics are needed to describe the motion of a marine vessel. The kinematics describe the motion of a vessel in different coordinate systems, whereas the kinetics describe the forces and moments causing the motion of the vessel. More information on these subjects are found in [15]. Further, the equations presented in the observer are based on the strapdown equations, containing the fundementals of position, velocity and attitude. See [39], [38] and [15] for references on this subject.

The nonlinear observer utilized in this thesis was originally introduced by Salcudean [35]. The work of Salcudean was further extended to include GNSS aiding and gyroscope bias estimation by Vik and Fossen [40], with the assumption of independent attitude measurements.

Since independent attitude measurements in most cases are unavailible, the use of vector measurements was pursued in the work of Hamel et al. [22] and Mahony et al. [30], assuming stationary reference vectors. Mahony et al. [34] and Hua [24] extended this research to include time-varying reference vectors, but without the capability of gyroscope bias estimation.

Grip, Fossen, Johansen and Saberi [17, 18] extended this work to include gyroscope bias estimation, in the case of time varying reference vectors. In addition, estimation of vector bias was included, i.e. accelerometer and magnetometer bias.

Grip et al. [20] augmenteded the results from [17, 18] to express estimates of position, velocity and attitude in the ECEF frame, using a unit quaternion to represent the attitude. This augmented observer will be the basis when deriving the modified nonlinear observer for dynamic positioning.

1.3 Contribution of the Thesis

This thesis should include these main contributions:

- Set up a logging system in collaboration with Marine Technologies LLC (MT) to collect data from sensors, state estimator and reference system.
- Develop a model-free state estimator for position, velocity and attitude, using IMU, gyrocompass and GNSS.
- Implement the estimator in MATLAB/Simulink and evaluate its performance in open loop. Compare with a model-based estimator in the presence of external forces.
- Test the state estimator using the ADIS16485 MEMS based IMU on a trolley in collaboration with MT.

The logging system was implemented on a control computer (CC) delivered by MT. The performance of the the model-free state estimator was compared with the TSS Orion INS, a navigation system with high-quality position and attitude estimates.

As earlier mentioned, the state estimator was based on the nonlinear observer from Grip et al. [20]. Since this observer used magnetometer measurements in the attitude observer's injection term, this had to be modified to utilize the gyrocompass heading measurements.

1.4 Outline of the Thesis

The thesis started with describing the kinematics and kinetics of the motion of an object in different coordinate systems. Secondly, an overview over different navigation systems and sensors are given, before presenting the modified observer used to achieve the results from the simulations and experimental tests. The results are then discussed before a conclusion is presented with a proposed list of future work.

The thesis is organized in the following way:

Chapter 1 Gives an introduction which includes the motivation, theory basis and contribution of the thesis.

Chapter 2 Studies the motion of an object in terms of kinematics and kinetics.

Chapter 3 Gives an overview of different navigation systems and sensors.

Chapter 4 Presents the nonlinear observer with necessary modifications to include a gyrocompass.

Chapter 5 Presents the setup utilized to test and vailidate the modified observer.

Chapter 6 Presents the results and discussion obtained from the simulations.

Chapter 7 Presents the results and discussion obtained from the experimental tests.

Chapter 8 Gives a conclusion and proposes future work based on the results of the thesis.

The following appendices are included:

Appendix A The simulator design.

Appendix B The Kalman filter and nonlinear observer comparison.

Appendix C The nonlinear observer code.

Appendix D Additional data.

Appendix E The digital appendix.

Chapter 2

Kinematics and Kinetics

This chapter describes the dynamics of an object using kinematics and kinetics. Section 2.1 studies the kinematics used to describe the motion of an object in different coordinate systems, using geometrical aspects. Section 2.2 studies the forces and moments causing the motion, more specifically denoted as the kinetics. The contents of the following sections are mainly based on [15] and [39].

2.1 Kinematics

When studying the motion of a marine vessel the knowledge of different reference frames are needed. Four main reference frames are going to be revised. The first two frames are the Earth-Centered Inertial (ECI) frame and the Earth-Centered Earth-Fixed (ECEF) frame, where both have their respective origins located in the center of the earth.

The next two reference frames are the North-East-Down (NED) frame and the body-fixed (BODY) frame, with defined origins relative to the Earth's reference ellipsoid (World Geodetic System, 1984 (WGS84)) and fixed to the vessel, respectively. See Figure 2.1 for an illustration of the different reference frames.

2.1.1 Reference Frames

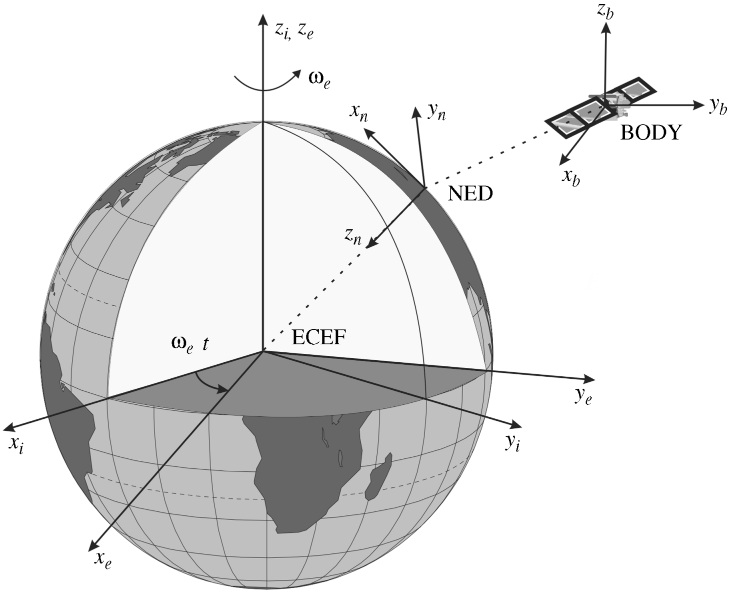


Figure 2.1: Illustration of different reference frames, Photo: [15]

Earth-Centered Reference Frames

ECI The Earth-Centered Inertial (ECI) frame $\{i\} = (x_i, y_i, z_i)$ is a global inertial reference frame with origin o_i in the center of the earth. The axes of the ECI frame are fixed in space, where the x-y plane coincides with the earth's equatorial plane and the z-axis extends through the earth's rotational axis.

ECEF The Earth-Centered Earth-Fixed (ECEF) frame $\{e\} = (x_e, y_e, z_e)$ has its origin o_e in the center of the earth. The x-y plane of the ECEF frame rotate relative to the ECI frame with an angular rate $\omega_{ie} = 7.2921 \times 10^{-5}$ $[rad/s]$, whereas the z-axis goes through the north pole. For a marine vessel moving at relatively low speed, the Earth's rotation rate can be neglected. The ECEF frame be considered as inertial. This frame is the most commonly used coordinate system for global navigation.

LLA Another way of representing position coordinates in $\{e\}$ can be done with the use of Latitude, Longitude and Altitude (LLA) coordinates $\{l\} = (\mu_l, \varphi_l, h_l)$, where μ_l and φ_l are represented in degrees and h_l in meters. The latitude $\mu_l = \pm 90$ $[deg]$ and $\varphi_l = \pm 180$ $[deg]$ cover the whole planet, where $\mu_l = 0$ $[deg]$ and $\varphi_l = 0$ $[deg]$ is the intersection between the equator and the prime meridian. The ECEF's x-axis goes through this point.

Geographic Reference Frames

NED A North-East-Down frame $\{n\} = (x_n, y_n, z_n)$ with origin o_n is a geographic coordinate system defined relative to the Earth's reference ellipsoid (WGS84). It's defined as the tangent plane on the surface of the Earth, with the x-axis pointing to the true North, y-axis pointing to the East, and the z-axis pointing downwards. For local navigation with nearly constant latitude, longitude and altitude, the NED frame can be considered to be inertial.

BODY A body-fixed frame $\{b\} = (x_b, y_b, z_b)$ with origin o_b is a moving coordinate system fixed to the vehicle. The axis x_b is directed forward (from aft to fore), the axis y_b is directed to the right (starboard), and the axis z_b is directed downwards (from top to bottom). The origin o_b will be referred as the center of origin (CO) of the vessel.

Notation

Scalar parameters and variables will be represented with lowercase notation throughout this thesis. Vectors and matrices are represented with bold lower and upper case typing, respectively. Superscripts will be used to denote which coordinate system the variable is expressed in. Subscripts will be used to denote which frames the variable is relative to.

Consider these vectors as examples:

$\mathbf{\Theta}_{nb} = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^\top \in \mathbb{S}^3$ Attitude between $\{b\}$ and $\{n\}$ in Euler angles.

$\mathbf{q}_{nb} = \begin{bmatrix}\eta & \boldsymbol{\varepsilon}^\top\end{bmatrix}^\top$, $\boldsymbol{\varepsilon} \in \mathbb{R}^3$, $\eta \in \mathbb{R}$ Attitude between $\{b\}$ and $\{n\}$ in quaternions.

$\mathbf{p}^n_{b/n} = \begin{bmatrix}x^n & y^n & z^n\end{bmatrix}^\top \in \mathbb{R}^3$ Position of $\{b\}$ with respect to $\{n\}$ expressed in $\{n\}$.

$\mathbf{v}^n_{b/e} = \begin{bmatrix}u^n & v^n & w^n\end{bmatrix}^\top \in \mathbb{R}^3$ Velocity of $\{b\}$ with respect to $\{e\}$ expressed in $\{n\}$.

$\mathbf{v}^b_{b/e} = \begin{bmatrix}u^b & v^b & w^b\end{bmatrix}^\top \in \mathbb{R}^3$ Velocity of $\{b\}$ with respect to $\{e\}$ expressed in $\{b\}$.

where $\mathbb{R}^3$ denotes the Euclidian space of three dimensions and $\mathbb{S}^3$ denotes a sphere of three dimensions. Most of the vectors in this thesis are described in $\mathbb{R}^3$, whereas matrices are described in $\mathbb{R}^{3\times3}$. Unless otherwise stated, this will be the case. Mathematical notations can also be found in [15].

2.1.2 Transformation between BODY and NED

There exists a rotation matrix $\mathbf{R}$ between two arbitrary reference frames $\{a\}$ and $\{b\}$ denoted as $\mathbf{R}_b^a$, which is a member of the special orthogonal group of order 3 ($SO(3)$), defined in [15] as:

$$SO(3) = \{\mathbf{R} | \mathbf{R} \in \mathbb{R}^{3\times3},\ \mathbf{R} \text{ is orthogonal and } det(\mathbf{R}) = 1\} \tag{2.1}$$

The group $SO(3)$ is a subset of all orthogonal matrices of order 3, which means that $SO(3) \subset O(3)$ where $O(3)$ is defined as:

$$O(3) := \{\mathbf{R} | \mathbf{R} \in \mathbb{R}^{3\times3},\ \mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I}_{3\times3}\} \tag{2.2}$$

where $\mathbf{I}_{3\times3}$ is defined as a matrix with ones on the diagonal, denoted as the identity matrix. The subscript of the identity matrix denotes its dimensions, in rows and columns, respectively.

A rotation matrix $\mathbf{R}_{\beta,\gamma}$, which corresponds to a rotation angle β about the $\boldsymbol{\gamma}$ axis, can according to [28] be defined as:

$$\mathbf{R}_{\beta,\gamma} = \mathbf{I}_{3\times3} + \sin(\beta)\mathbf{S}(\boldsymbol{\gamma}) + [1 - \cos(\beta)]\mathbf{S}^2(\boldsymbol{\gamma}) \tag{2.3}$$

where $\boldsymbol{\gamma} = \begin{bmatrix}\gamma_1 & \gamma_2 & \gamma_3\end{bmatrix}^\top$ is a unit vector parallel to the rotation axis, and $\mathbf{S}(\boldsymbol{\gamma})$ is a skew-symmetric matrix defined from [15] as:

$$\mathbf{S}(\boldsymbol{\gamma}) = -\mathbf{S}^\top(\boldsymbol{\gamma}) = \begin{bmatrix} 0 & -\gamma_3 & \gamma_2 \\ \gamma_3 & 0 & -\gamma_1 \\ -\gamma_2 & \gamma_1 & 0 \end{bmatrix} \tag{2.4}$$

Euler Angles

The most intuitive way to understand the attitude of an object is with the use of the Euler angle vector $\boldsymbol{\Theta}_{nb} = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^\top$ which describes the angles between BODY and NED, where ϕ is roll, θ is pitch and ψ is yaw.

Linear Velocity

A rotation matrix $\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})$ can be created from the Euler angles $\boldsymbol{\Theta}_{nb}$. This matrix can be used to transform vectors between the BODY and NED frame, such as the velocity and acceleration.

The following equation illustrates the transformation of the velocity vector between the two respective frames:

$$\dot{\mathbf{p}}_{b/n}^n = \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})\mathbf{v}_{b/n}^b \Leftrightarrow \mathbf{v}_{b/n}^b = \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})^\top \dot{\mathbf{p}}_{b/n}^n \Leftrightarrow \mathbf{v}_{b/n}^b = \mathbf{R}_n^b(\boldsymbol{\Theta}_{nb})\dot{\mathbf{p}}_{b/n}^n \tag{2.5}$$

The rotation matrix $\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})$ contains three principal rotations which corresponds to the x-, y- and z- axis:

$$\mathbf{R}_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}, \quad \mathbf{R}_{y,\theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}, \quad \mathbf{R}_{z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

where $c\cdot = cos(\cdot)$ and $s\cdot = sin(\cdot)$.

It is common to describe the rotation matrix $\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})$ with combinations of three principle rotations in the following order: First rotate around the z-axis, then about the y-axis, and last about the x-axis. This rotation order is called the zyx-convention and is mathematically equivalent to, [15]:

$$\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) := \mathbf{R}_{z,\psi}\mathbf{R}_{y,\theta}\mathbf{R}_{x,\phi} \tag{2.7}$$

Combining (2.6) and (2.7) yield the extended form:

$$\boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{2.8}$$

Angular Velocity

The relationship between the Euler rate vector $\dot{\boldsymbol{\Theta}}_{nb} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^\top$ and the body-fixed angular velocity vector $\boldsymbol{\omega}_{b/n}^b = \begin{bmatrix} p & q & r \end{bmatrix}^\top$ are related through a transformation matrix $\mathbf{T}_\Theta(\boldsymbol{\Theta}_{nb})$.

$$\dot{\mathbf{\Theta}}_{nb} = \mathbf{T}_{\Theta}(\mathbf{\Theta}_{nb})\boldsymbol{\omega}^{b}_{b/n} \tag{2.9}$$

This transformation matrix was derived in [15] in the following way:

$$\boldsymbol{\omega}^{b}_{b/n} = \begin{bmatrix}\dot{\phi}\\0\\0\end{bmatrix} + \mathbf{R}^{\top}_{x,\phi}\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix} + \mathbf{R}^{\top}_{x,\phi}\mathbf{R}^{\top}_{y,\theta}\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix} := \mathbf{T}^{-1}_{\Theta}(\mathbf{\Theta}_{nb})\dot{\mathbf{\Theta}}_{nb} \tag{2.10}$$

Expanding $\mathbf{T}_{\Theta}(\mathbf{\Theta}_{nb})$ gives:

$$\boldsymbol{T}_{\Theta}(\mathbf{\Theta}_{nb}) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \tag{2.11}$$

where $c\cdot = cos(\cdot)$, $s\cdot = sin(\cdot)$ and $t\cdot = tan(\cdot)$.

It's worth noticing that the transformation matrix has a singularity for $\theta = \pm 90°$. For marine surface vessels this will not be a problem since the pitch never would reach this boundary.

Linear and Angular Velocities

The complete equation set for the kinematics between BODY and NED, using Euler angles, can be represented as:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_{\Theta}(\mathbf{\Theta}_{nb})\boldsymbol{\nu} \tag{2.12}$$

where

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix}(\dot{\mathbf{p}}^{n}_{b/n})^{\top} & \mathbf{\Theta}^{\top}_{nb}\end{bmatrix}^{\top} \tag{2.13}$$

$$\boldsymbol{\nu} = \begin{bmatrix}(\mathbf{v}^{b}_{b/n})^{\top} & (\boldsymbol{\omega}^{b}_{b/n})^{\top}\end{bmatrix}^{\top} \tag{2.14}$$

$$\mathbf{J}_{\Theta}(\mathbf{\Theta}_{nb}) = \begin{bmatrix}\mathbf{R}^{n}_{b}(\mathbf{\Theta}_{nb}) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{T}_{\Theta}(\mathbf{\Theta}_{nb})\end{bmatrix} \tag{2.15}$$

$\mathbf{0}_{3\times3}$ denotes a matrix with zeroes on all of its entries.

Unit Quaternions

Representing the attitude as a unit quaternion, instead of Euler angles, would be a singularity free approach. Unit quaternions are less intuitive to understand, but gives a more elegant solution both mathematically and for implementations on a computer.

A unit quaternion $\mathbf{q}$ is according to [15] defined as:

$$\mathbf{q} = \begin{bmatrix} \eta & \varepsilon_1 & \varepsilon_2 & \varepsilon_3 \end{bmatrix}^\top = \begin{bmatrix} \eta & \boldsymbol{\varepsilon}^\top \end{bmatrix}^\top \tag{2.16}$$

where η is the real part and $\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \varepsilon_3 \end{bmatrix}^\top$ is the imaginary part.

A set of unit quaternions Υ possesses the following property, [15]:

$$\Upsilon := \{\mathbf{q}|\mathbf{q}^\top\mathbf{q} = 1,\ \mathbf{q} = \begin{bmatrix} \eta & \boldsymbol{\varepsilon}^\top \end{bmatrix}^\top,\ \boldsymbol{\varepsilon} \in \mathbb{R}^3 \text{ and } \eta \in \mathbb{R}\} \tag{2.17}$$

The conjugate of $\mathbf{q}$ is denoted $\mathbf{q}^*$ and is defined in [20] as:

$$\mathbf{q}^* = \begin{bmatrix} \eta & -\boldsymbol{\varepsilon}^\top \end{bmatrix}^\top \tag{2.18}$$

When deriving the strapdown equations and the nonlinear observer in Chapter 3 and Chapter 4, the following notation was found useful, [20]:

$$\bar{\mathbf{a}} = \begin{bmatrix} 0 & \mathbf{a}^\top \end{bmatrix}^\top \tag{2.19}$$

where $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^\top$ is a vector and can be defined in quaternion notation with real part $\eta = 0$ and imaginary part $\boldsymbol{\varepsilon} = \mathbf{a}$.

Linear Velocity

The real and imaginary parts η and $\boldsymbol{\varepsilon}$ in the unit quaternion $\mathbf{q}$ can be be defined as, [15]:

$$\eta := \cos(\frac{\beta}{2}) \tag{2.20}$$

$$\boldsymbol{\varepsilon} := \boldsymbol{\gamma}\sin(\frac{\beta}{2}) \tag{2.21}$$

where $\boldsymbol{\gamma}$ is a unit vector satisfying:

$$\boldsymbol{\gamma} = \pm \frac{\boldsymbol{\varepsilon}}{\sqrt{\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon}}}, \text{ if } \sqrt{\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon}} \neq 0 \tag{2.22}$$

Combining (2.3), (2.20) and (2.21) give the following rotation matrix $\boldsymbol{R}_b^n(\mathbf{q}_{nb})$ between BODY and NED:

$$\boldsymbol{R}_b^n(\mathbf{q}_{nb}) := \mathbf{R}_{\eta,\varepsilon} = \mathbf{I} + 2\eta\mathbf{S}(\boldsymbol{\varepsilon}) + 2\mathbf{S}^2(\boldsymbol{\varepsilon}) \tag{2.23}$$

where the expansion of (2.23) yields:

$$\boldsymbol{R}_b^n(\mathbf{q}) = \begin{bmatrix} 1-2(\varepsilon_2^2+\varepsilon_3^2) & 2(\varepsilon_1\varepsilon_2-\varepsilon_3\eta) & 2(\varepsilon_1\varepsilon_3+\varepsilon_2\eta) \\ 2(\varepsilon_1\varepsilon_2+\varepsilon_3\eta) & 1-2(\varepsilon_1^2+\varepsilon_3^2) & 2(\varepsilon_2\varepsilon_3-\varepsilon_1\eta) \\ 2(\varepsilon_1\varepsilon_3-\varepsilon_2\eta) & 2(\varepsilon_2\varepsilon_3+\varepsilon_1\eta) & 1-2(\varepsilon_1^2+\varepsilon_2^2) \end{bmatrix} \tag{2.24}$$

The rotation matrix $\boldsymbol{R}_b^n(\mathbf{q}_{nb})$ has the same properties as the rotation matrix $\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})$, obtained from the Euler angles $\boldsymbol{\Theta}_{nb}$. $\boldsymbol{R}_b^n(\mathbf{q}_{nb})$ can therefore be used in the same manner to rotate between BODY and NED:

$$\dot{\mathbf{p}}_{b/n}^n = \mathbf{R}_b^n(\mathbf{q}_{nb})\mathbf{v}_{b/n}^b \Leftrightarrow \mathbf{v}_{b/n}^b = \mathbf{R}_b^n(\mathbf{q}_{nb})^\top \dot{\mathbf{p}}_{b/n}^n \Leftrightarrow \mathbf{v}_{b/n}^b = \mathbf{R}_n^b(\mathbf{q}_{nb})\dot{\mathbf{p}}_{b/n}^n \tag{2.25}$$

Another way of representing rotations with quaternions can be given as, [20]:

$$\bar{\mathbf{v}}_{b/n}^b = \mathbf{q}_{nb} \otimes \bar{\dot{\mathbf{p}}}_{b/n}^n \otimes \mathbf{q}_{nb}^* \tag{2.26}$$

where ⊗ denotes the quaternion product, and the velocity vectors are defined on the notation given by (2.19).

Angular Velocity

The differential equation for a rotation matrix $\boldsymbol{R}_b^n$ can be defined as, [15]:

$$\dot{\boldsymbol{R}}_b^n = \boldsymbol{R}_b^n\mathbf{S}(\boldsymbol{\omega}_{b/n}^b) \tag{2.27}$$

Substituting (2.24) into (2.27) gives the relationship between the quaternion rate vector $\dot{\mathbf{q}}_{nb}$ and the body-fixed angular velocity vector $\boldsymbol{\omega}^{b}_{b/n}$:

$$\dot{\mathbf{q}}_{nb} = \mathbf{T}_q(\mathbf{q}_{nb})\boldsymbol{\omega}^{b}_{b/n} \tag{2.28}$$

where $\mathbf{T}_q(\mathbf{q}_{nb})$ is the transformation matrix, with the expanded form:

$$\mathbf{T}_q(\mathbf{q}_{nb}) = \frac{1}{2}\begin{bmatrix} -\varepsilon_1 & -\varepsilon_2 & -\varepsilon_3 \\ \eta & -\varepsilon_3 & \varepsilon_2 \\ \varepsilon_3 & \eta & -\varepsilon_1 \\ -\varepsilon_2 & \varepsilon_1 & \eta \end{bmatrix} \tag{2.29}$$

An alternative way of representing (2.28) is, [20]:

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

where $\bar{\boldsymbol{\omega}}^{b}_{b/n}$ is defined with the use of (2.19).

Linear and Angular Velocities

The complete equation set for the kinematics between BODY and NED, using quaternions, can be represented as

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_q(\mathbf{q}_{nb})\boldsymbol{\nu} \tag{2.31}$$

where

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \begin{bmatrix} (\dot{\mathbf{p}}^{n}_{b/n})^{\top} & \dot{\mathbf{q}}^{\top}_{nb} \end{bmatrix}^{\top} \\ \boldsymbol{\nu} &= \begin{bmatrix} (\mathbf{v}^{b}_{b/n})^{\top} & (\boldsymbol{\omega}^{b}_{b/n})^{\top} \end{bmatrix}^{\top} \\ \mathbf{J}_q(\mathbf{q}) &= \begin{bmatrix} \boldsymbol{R}^{n}_{b}(\mathbf{q}_{nb}) & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{4\times 3} & \mathbf{T}_q(\mathbf{q}_{nb}) \end{bmatrix} \end{aligned}$$

Transformation between Euler Angles and Unit Quaternions

The transformations between Euler angles and unit quaternions can be done using the functions `q2euler` and `euler2q` in MATLAB, which were found in the gnc library within the mss toolbox[1]. For more information on this topic the reader is referred to [15].

Transformation between Body-Fixed Coordinates

The body-fixed coordinate system $\{b\}$ has its center in CO. In addition to $\{b\}$, two other body-fixed coordinate systems will be defined as:

$\{m\}$ The origin of the IMU, where the lever arm $\mathbf{r}^b_m = \begin{bmatrix} x^b_m & y^b_m & z^b_m \end{bmatrix}^\top$ denotes the position of $\{m\}$ with respect to $\{b\}$.

$\{gnss\}$ The origin of the GNSS receiver, where the lever arm $\mathbf{r}^b_{gnss} = \begin{bmatrix} x^b_{gnss} & y^b_{gnss} & z^b_{gnss} \end{bmatrix}^\top$ denotes the position of $\{gnss\}$ with respect to $\{b\}$.

Position transformations between these body-fixed frames expressed in the NED or ECEF frame can be illustrated through an example, [15]:

$$\mathbf{p}^n_{b/n} = \mathbf{p}^n_{m/n} - \mathbf{R}^n_b(\boldsymbol{\Theta}_{nb})\mathbf{r}^b_m \Leftrightarrow \mathbf{p}^n_{m/n} = \mathbf{p}^n_{b/n} + \mathbf{R}^n_b(\boldsymbol{\Theta}_{nb})\mathbf{r}^b_m \tag{2.32}$$

where $\mathbf{R}^n_b(\boldsymbol{\Theta}_{nb})$ is the rotation matrix obtained from the Euler angle vector $\boldsymbol{\Theta}_{nb}$.

The velocity transformation can be realized using (2.27):

$$\mathbf{v}^n_{b/n} = \mathbf{v}^n_{m/n} - \mathbf{R}^n_b(\boldsymbol{\Theta}_{nb})\mathbf{S}(\boldsymbol{\omega}^b_{b/n})\mathbf{r}^b_m \Leftrightarrow \mathbf{v}^n_{m/n} = \mathbf{v}^n_{b/n} + \mathbf{R}^n_b(\boldsymbol{\Theta}_{nb})\mathbf{S}(\boldsymbol{\omega}^b_{b/n})\mathbf{r}^b_m$$

Since the angular acceleration $\dot{\boldsymbol{\omega}}^n_{b/n}$ is not utilized or availible in this thesis, the acceleration transformation is not considered.

[1] The toolbox can be found in http://www.marinecontrol.org/

2.1.3 Transformation between ECEF and LLA

Parameters	**Description**
$f = \frac{1}{298.257223563}$	Ellipsoid flattening
$a = 6378137\ [m]$	Semi-major axis
$b = a(1-f)\ [m]$	Semi-minor axis
$e = \sqrt{\frac{a^2-b^2}{a^2}}$	First eccentricity
$e' = \sqrt{\frac{a^2-b^2}{b^2}}$	Second eccentricity
$\omega_{ie} = 7.292115 \times 10^{-5}\ [rad/s]$	Turn rate of the Earth

Table 2.1: WGS-84 parameters

LLA to ECEF

The transformation from LLA coordinates $\mathbf{p}^l_{b/l} = \begin{bmatrix}\mu & \varphi & h\end{bmatrix}^\top$ to ECEF coordinates $\mathbf{p}^e_{b/e} = \begin{bmatrix}x^e & y^e & z^e\end{bmatrix}^\top$ can be done using the closed method defined in [31]:

$$\begin{aligned} x^e &= (R+h)\cos(\mu)\cos(\varphi) \\ y^e &= (R+h)\cos(\mu)\sin(\varphi) \\ z^e &= (\frac{b^2}{a^2}R+h)\sin(\mu) \end{aligned} \tag{2.33}$$

where

$$R = \frac{a}{\sqrt{1-e^2\sin^2(\mu)}} \tag{2.34}$$

The remaining parameters can be found in Table 2.1.

ECEF to LLA

The transformation from ECEF coordinates $\mathbf{p}^e_{b/e} = \begin{bmatrix}x^e & y^e & z^e\end{bmatrix}^\top$ to LLA coordinates $\mathbf{p}^l_{b/e} = \begin{bmatrix}\mu & \varphi & h\end{bmatrix}^\top$ can be done using the closed method defind in [31]:

$$\begin{aligned}
\mu &= \arctan(\frac{z^e + e'^2 b \sin^3(\beta)}{r - e^2 a \cos^3(\beta)}) \\
\varphi &= \arctan(\frac{y^e}{x^e}) \\
h &= \frac{p}{\cos(\mu)} - R
\end{aligned} \tag{2.35}$$

where R is defined in (2.34), and

$$r = \sqrt{(x^e)^2 + (y^e)^2} \tag{2.36}$$

$$\beta = \arctan(\frac{z^e a}{rb}) \tag{2.37}$$

The remaining parameters can be found in Table 2.1.

2.1.4 Transformation between ECEF and NED

From the LLA coordinates $\mathbf{p}^l_{b/e} = \begin{bmatrix}\mu & \varphi & h\end{bmatrix}^\top$ a new vector $\mathbf{\Theta}_{en} = \begin{bmatrix}\mu & \varphi\end{bmatrix}^\top \in S^2$ can be defined, containing the latitude μ and longitude φ. A principal rotation φ about the z-axis and a rotation $-\mu - \frac{\pi}{2}$ about the y-axis can be carried out to obtain the rotation matrix $\mathbf{R}^e_n(\mathbf{\Theta}_{en}) = \mathbf{R}_{z,\varphi}\mathbf{R}_{y,-\mu-\frac{\pi}{2}}$ between NED and ECEF, [15]:

$$\mathbf{R}^e_n(\mathbf{\Theta}_{en}) = \begin{bmatrix}\cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1\end{bmatrix} \begin{bmatrix}\cos(-\mu - \frac{\pi}{2}) & 0 & \sin(-\mu - \frac{\pi}{2}) \\ 0 & 1 & 0 \\ -\sin(-\mu - \frac{\pi}{2}) & 0 & \cos(-\mu - \frac{\pi}{2})\end{bmatrix} \tag{2.38}$$

Which after some calculations yields the expanded form:

$$\mathbf{R}^e_n(\mathbf{\Theta}_{en}) = \begin{bmatrix}-\cos(\varphi)\sin(\mu) & -\sin(\varphi) & -\cos(\varphi)\cos(\mu) \\ -\sin(\varphi)\sin(\mu) & \cos(\varphi) & -\sin(\varphi)\cos(\mu) \\ \cos(\mu) & 0 & -\sin(\mu)\end{bmatrix} \tag{2.39}$$

The relationship between $\dot{\mathbf{p}}^e_{b/e}$ and $\mathbf{v}^n_{b/e}$ can be expressed through $\mathbf{R}^e_n(\mathbf{\Theta}_{en})$ with:

$$\dot{\mathbf{p}}^e_{b/e} = \mathbf{R}^e_n(\mathbf{\Theta}_{en})\mathbf{v}^n_{b/e} \tag{2.40}$$

where the position $\mathbf{p}^e_{b/e}$ can be found by integrating (2.40).

When considering navigation in a small local area, $\mathbf{R}^e_n(\boldsymbol{\Theta}_{en})$ can be approximated using a constant latitude μ_0 and longitude φ_0 to obtain $\mathbf{R}^e_n(\boldsymbol{\Theta}_0)$.

With this approximation (2.40) gets simplified to:

$$\dot{\mathbf{p}}^e_{b/e} = \mathbf{R}^e_n(\boldsymbol{\Theta}_0)\mathbf{v}^n_{b/e} \tag{2.41}$$

which will be sufficient when deviations from μ_0 and φ_0 are small.

2.2 Kinetics

This section will shortly derive the equations of motion for a marine vessel with six degrees of freedom (6-DOF), both the generalized and the linearized version. The linearized version will be used to design a Kalman filter for comparison with the nonlinear observer in Chapter 6. The NED frame is utilized to describe the position, velocity and attitude.

2.2.1 Generalized

The generalized equations of motion for a 6-DOF marine vessel can according to [15] be given as:

$$\mathbf{M}_{RB}\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} \quad = \quad \boldsymbol{\tau}_{RB} \tag{2.42}$$

where

$$\begin{aligned} \boldsymbol{\nu} &= \begin{bmatrix} (\mathbf{v}^b_{b/n})^\top & (\boldsymbol{\omega}^b_{b/n})^\top \end{bmatrix}^\top && (2.43) \\ \boldsymbol{\tau}_{RB} &= \begin{bmatrix} (\mathbf{f}^b_b)^\top & (\boldsymbol{m}^b_b)^\top \end{bmatrix}^\top && (2.44) \end{aligned}$$

where $\boldsymbol{\nu}$ is the velocity vector and $\boldsymbol{\tau}_{RB}$ is the generalized force and moment vector. Further, $\mathbf{f}^b_b$ and $\boldsymbol{m}^b_b$ are defined as the force and the moment vectors, respectively. $\mathbf{M}_{RB}$ is the rigid-body mass matrix and $\mathbf{C}_{RB}$ is the rigid-body Coriolis and centripetal matrix.

2.2.2 Forces and Moments

The generalized vector $\boldsymbol{\tau}_{RB}$ from (2.42) is further defined as, [15]:

$$\boldsymbol{\tau}_{RB} = \boldsymbol{\tau}_{hd} + \boldsymbol{\tau}_{hs} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} + \boldsymbol{\tau}_{pid} \tag{2.45}$$

where $\boldsymbol{\tau}_{hd}$ is the hydrodynamic forces, $\boldsymbol{\tau}_{hs}$ is the hydrostatic forces, $\boldsymbol{\tau}_{wind}$ is the wind induced forces, $\boldsymbol{\tau}_{wave}$ is the wave induced forces, and $\boldsymbol{\tau}_{pid}$ is the forces made by the thrusters.

The hydrodynamic forces $\boldsymbol{\tau}_{hd}$ are generated by the liquids in motion and is given as, [15]:

$$\boldsymbol{\tau}_{hd} = -\mathbf{M}_A\dot{\boldsymbol{\nu}}_r - \mathbf{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r \tag{2.46}$$

where $\mathbf{M}_A$ is the added mass matrix, $\mathbf{C}_A(\boldsymbol{\nu}_r)$ is the added Coriolis and centripetal matrix, $\mathbf{D}(\boldsymbol{\nu}_r) = \mathbf{D} + \mathbf{D}_n(\boldsymbol{\nu}_r)$ is the damping matrix, containing a linear part $\mathbf{D}$ and a nonlinear quadratic part $\mathbf{D}_n(\boldsymbol{\nu}_r)$.

Further, the hydrostatic forces $\boldsymbol{\tau}_{hs}$ include buoyancy and gravitational forces and is defined as:

$$\boldsymbol{\tau}_{hs} = -\mathbf{g}(\boldsymbol{\eta}) \tag{2.47}$$

where $\mathbf{g}(\boldsymbol{\eta})$ is the restoring force and moment vector.

2.2.3 Maneuvering Model

The complete nonlinear 6-DOF maneuvering model is given by combining (2.12), (2.42), (2.45), (2.46) and (2.47):

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \mathbf{J}_\Theta(\boldsymbol{\Theta}_{nb})\boldsymbol{\nu}_r & (2.48)\\ \mathbf{M}\dot{\boldsymbol{\nu}}_r + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{g}(\boldsymbol{\eta}) &= \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} + \boldsymbol{\tau} & (2.49) \end{aligned}$$

where

$$\mathbf{M} = \mathbf{M}_A + \mathbf{M}_{RB} \tag{2.50}$$
$$\mathbf{C}(\boldsymbol{\nu}_r) = \mathbf{C}(\boldsymbol{\nu}_r) + \mathbf{C}(\boldsymbol{\nu}_r) \tag{2.51}$$
$$\mathbf{D}(\boldsymbol{\nu}_r) = \mathbf{D} + \mathbf{D}_n(\boldsymbol{\nu}_r) \tag{2.52}$$

The relative velocity vector $\boldsymbol{\nu}_r$ is defined as:

$$\boldsymbol{\nu}_r := \boldsymbol{\nu} - \boldsymbol{\nu}_c \tag{2.53}$$

where $\boldsymbol{\nu}_c$ is the current velocity.

Linearized Model for Low-Speed Applications

For low-speed applications, i.e. dynamic postioning, the current velocity can be treated as a slowly varying bias $\mathbf{b}$ and compensated by the use of integral action, [15]. The current velocity can therefore be neglected and $\boldsymbol{\nu}_r = \boldsymbol{\nu}$.

Another reasonable assumption when linearizing the nonlinear manevering model (2.49), is to assume small roll and pitch angles, $\phi = 0$ and $\theta = 0$. This assumption will simplify the rotation and transformation matrices such that $\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) = \mathbf{R}(\psi) = \mathbf{R}_{z,\psi}$ and $\boldsymbol{T}_\Theta(\boldsymbol{\Theta}_{nb}) = \mathbf{I}_{3\times3}$, [15].

(2.12) can then be reduced to:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_\Theta(\boldsymbol{\Theta}_{nb})\boldsymbol{\nu} \tag{2.54}$$
$$\Downarrow \quad \phi = \theta = 0$$
$$\dot{\boldsymbol{\eta}} = \mathbf{P}(\psi)\boldsymbol{\nu} \tag{2.55}$$

where

$$\mathbf{P}(\psi) = \begin{bmatrix} \mathbf{R}(\psi) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix} \tag{2.56}$$

When working with linear theory, it is useful to define a vessel parallel (VP) coordinate system, [15]:

$$\boldsymbol{\eta}_{vp} := \mathbf{P}(\psi)^\top \boldsymbol{\eta} \tag{2.57}$$

where $\boldsymbol{\eta}_{vp}$ is the position and attitude in $\{n\}$ expressed in $\{b\}$. Further, the differential equation $\dot{\boldsymbol{\eta}}_{vp}$ for low-speed applications can be approximated to, [15]:

$$\dot{\boldsymbol{\eta}}_{vp} \approx \boldsymbol{\nu} \tag{2.58}$$

for small attitude variations.

The hydrostatic forces $\mathbf{g}(\boldsymbol{\eta})$ can then be approximated with:

$$\mathbf{g}(\boldsymbol{\eta}) \approx \mathbf{P}(\psi)^\top \mathbf{G}\boldsymbol{\eta} = \mathbf{G}\boldsymbol{\eta}_{vp} \tag{2.59}$$

where $\mathbf{G}$ is the linear restoring force and moment matrix.

In the case of low speed $\boldsymbol{\nu} \approx 0$ the nonlinear Coriolis and centripetal matrix $\mathbf{C}(\boldsymbol{\nu})$, in addition to the nonlinear quadratic damping matrix $\mathbf{D}_n(\boldsymbol{\nu})$, can be neglected. This reduce (2.49) to the linearized system:

$$\begin{aligned}
\boldsymbol{\eta} &= \mathbf{P}(\psi)\boldsymbol{\eta}_{vp} && (2.60)\\
\dot{\boldsymbol{\eta}}_{vp} &= \boldsymbol{\nu} && (2.61)\\
\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{D}\boldsymbol{\nu} + \mathbf{G}\boldsymbol{\eta}_{vp} &= \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} + \boldsymbol{\tau} && (2.62)\\
\dot{\mathbf{b}} &= 0 && (2.63)
\end{aligned}$$

where $\mathbf{b}$ is the slowly varying bias vector added to compensate for ocean currents. Equations (2.60-2.63) are useful when implementing a Kalman Filter for dynamic positioning.

Chapter 3

Navigation Systems and Sensors

This chapter will review different navigation systems, such as the inertial navigation system (INS), the global navigation satellite system (GNSS), and the integration of both systems (INS/GNSS). Different integration architectures will be described, mentioning strengths and weaknesses. In addition, the fundementals about the inertial measurement unit (IMU) and the gyrocompass will be revised.

3.1 The Inertial Navigation System (INS)

3.1.1 The Fundementals

Inertial navigation utilize measurements of mainly two sensors, accelerometers and gyroscopes. The accelerometer measures the specific force, which can be integrated once to get the velocity and twice to get the position, after subtracting the gravitational components. The gyroscope measures the angular velocity, and can be integrated once to get the attitude. These sensors will in combination with proper electronics and software result in an inertial navigation system (INS), [39].

Three accelerometers and three gyroscopes are usually combined in one unit, measuring linear and angular motion about the main three axes in a body-fixed frame. This unit is referred to as the inertial measurement unit (IMU) with 6 degrees of freedom (6-DOF).

A navigation system usually estimate the position, velocity and attitude (PVA) expressed in a known Cartesian reference frame, i.e. global navigation coordinates in the ECEF frame or local navigaition in the NED frame. To acquire this, the inital position, velocity and attitude of the vessel in the respective navigation frame is supposed to be known.

The knowledge of the gravity in the navigation frame is needed to subtract static components from the accelerometers. This can be aquired by the use of an gravitational model, see [32] for more information. In combination with proper rotation and transformation matrices, estimates of PVA expressed in the navigation frame is achieved. This way of navigation is referred to as dead-teckoning (DR). More information about inertial navigation can be found in [39] and [15].

3.1.2 The Strapdown Equations

The navigational equation in the fixed inertial ECI frame is given as, [38, 39]:

$$\ddot{\mathbf{p}}^i = \mathbf{f}^i + \breve{\mathbf{g}}^i(\mathbf{p}^i) \tag{3.1}$$

where $\ddot{\mathbf{p}}^i$ is the total acceleration, $\breve{\mathbf{g}}^i(\mathbf{p}^i)$ is the gravitational acceleration given at a specific postition $\mathbf{p}^i$, and $\mathbf{f}^i$ is the specific force.

Position and Velocity

The relationship between $\mathbf{p}^i$ and $\mathbf{p}^e$ can be expressed through the transfomation matrix $\mathbf{R}_e^i$ as:

$$\mathbf{p}^i = \mathbf{R}_e^i \mathbf{p}^e \tag{3.2}$$

where the velocity can be achieved by taking the derivative of (3.2):

$$\dot{\mathbf{p}}^i = \dot{\mathbf{R}}_e^i \mathbf{p}^e + \mathbf{R}_e^i \dot{\mathbf{p}}^e \tag{3.3}$$

and the acceleration can be calculated by the derivative of (3.3):

$$\ddot{\mathbf{p}}^i = \ddot{\mathbf{R}}_e^i \mathbf{p}^e + 2\dot{\mathbf{R}}_e^i \dot{\mathbf{p}}^e + \mathbf{R}_e^i \ddot{\mathbf{p}}^e \tag{3.4}$$

The derivative of $\mathbf{R}_e^i$ is given by using (2.24):

$$\dot{\boldsymbol{R}}_e^i = \boldsymbol{R}_e^i \mathbf{S}(\boldsymbol{\omega}_{ie}^e) \tag{3.5}$$

with $\boldsymbol{\omega}_{ie}^e = \begin{bmatrix} 0 & 0 & \omega_{ie} \end{bmatrix}^\top$, where ω_{ie} is the Earth's rotational rate, see Table 2.1.

Taking the derivative of (3.5) yields:

$$\ddot{\mathbf{R}}_e^i = \dot{\boldsymbol{R}}_e^i \mathbf{S}(\boldsymbol{\omega}_{ie}^e) + \boldsymbol{R}_e^i \mathbf{S}(\dot{\boldsymbol{\omega}}_{ie}^e) \tag{3.6}$$

The Earth's rotational rate is assumed to be constant, hence $\dot{\boldsymbol{\omega}}_{ie}^e = \mathbf{0}_{3\times 1}$. With this assumption, substitution of (3.5) into (3.6) gives:

$$\ddot{\mathbf{R}}_e^i = \boldsymbol{R}_e^i \mathbf{S}(\boldsymbol{\omega}_{ie}^e)\mathbf{S}(\boldsymbol{\omega}_{ie}^e) \tag{3.7}$$

Then, by inserting (3.7) and (3.5) into (3.4):

$$\ddot{\mathbf{p}}^i = \boldsymbol{R}_e^i \mathbf{S}(\boldsymbol{\omega}_{ie}^e)\mathbf{S}(\boldsymbol{\omega}_{ie}^e)\mathbf{p}^e + 2\boldsymbol{R}_e^i \mathbf{S}(\boldsymbol{\omega}_{ie}^e)\dot{\mathbf{p}}^e + \mathbf{R}_e^i \ddot{\mathbf{p}}^e \tag{3.8}$$

Rearranging (3.8) with the substitution of (3.1) yields:

$$\begin{aligned} \ddot{\mathbf{p}}^e &= (\mathbf{R}_e^i)^\top(\mathbf{f}^i + \breve{\mathbf{g}}^i(\mathbf{p}^i)) - (\mathbf{R}_e^i)^\top \boldsymbol{R}_e^i \mathbf{S}(\boldsymbol{\omega}_{ie}^e)\mathbf{S}(\boldsymbol{\omega}_{ie}^e)\mathbf{p}^e && (3.9) \\ &\quad -2(\mathbf{R}_e^i)^\top \boldsymbol{R}_e^i \mathbf{S}(\boldsymbol{\omega}_{ie}^e)\dot{\mathbf{p}}^e && (3.10) \\ \ddot{\mathbf{p}}^e &= \mathbf{f}^e + \breve{\mathbf{g}}^e(\mathbf{p}^e) - \mathbf{S}(\boldsymbol{\omega}_{ie}^e)\mathbf{S}(\boldsymbol{\omega}_{ie}^e)\mathbf{p}^e - 2\mathbf{S}(\boldsymbol{\omega}_{ie}^e)\dot{\mathbf{p}}^e && (3.11) \end{aligned}$$

The differential equations for position and velocity can further be expressed as:

$$\begin{aligned} \dot{\mathbf{p}}^e &= \mathbf{v}^e && (3.12) \\ \dot{\mathbf{v}}^e &= -2\mathbf{S}(\boldsymbol{\omega}_{ie}^e)\dot{\mathbf{p}}^e + \mathbf{f}^e + \mathbf{g}^e(\mathbf{p}^e) && (3.13) \end{aligned}$$

where $\mathbf{g}^e(\mathbf{p}^e)$ is the plumb bob gravity vector defined as:

$$\begin{aligned}\mathbf{g}^e(\mathbf{p}^e) &= \breve{\mathbf{g}}^e(\mathbf{p}^e) - \mathbf{S}(\boldsymbol{\omega}^e_{ie})\mathbf{S}(\boldsymbol{\omega}^e_{ie})\mathbf{p}^e \\ &= \breve{\mathbf{g}}^e(\mathbf{p}^e) - \boldsymbol{\omega}^e_{ie} \times (\boldsymbol{\omega}^e_{ie} \times \mathbf{p}^e)\end{aligned}$$

using the cross product definition in [15]:

$$\mathbf{a} \times \mathbf{b} := \mathbf{S}(\mathbf{a})\mathbf{b} \tag{3.14}$$

where $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^\top$ and $\mathbf{b} = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}^\top$.

The specific force measured by the accelerometers $\mathbf{f}^b$ can be rotated to $\{e\}$ with:

$$\mathbf{f}^e = \mathbf{R}^e_b(\mathbf{q}_{eb})\mathbf{f}^b \tag{3.15}$$

where $\mathbf{q}_{eb}$ is the attitude quaternion between $\{e\}$ and $\{b\}$, revised in the next section.

Attitude

The differential equation for the unit quaternion $\mathbf{q}_{eb}$ between $\{e\}$ and $\{b\}$ is defined as, see (2.27) in Section 2.1.2:

$$\dot{\mathbf{q}}_{eb} = \mathbf{T}_q(\mathbf{q}_{eb})\boldsymbol{\omega}^b_{eb} \tag{3.16}$$

where $\boldsymbol{\omega}^b_{eb}$ is achieved by substracting the Earth's rotational vector $\boldsymbol{\omega}^e_{ie}$ from the angular velocity $\boldsymbol{\omega}^b_{ib}$, where the latter is measured by the gyroscopes. This gives the following expression for $\boldsymbol{\omega}^b_{eb}$:

$$\boldsymbol{\omega}^b_{eb} = \boldsymbol{\omega}^b_{ib} - \boldsymbol{R}^e_b(\mathbf{q}_{eb})^\top\boldsymbol{\omega}^e_{ie} \tag{3.17}$$

By substituting (3.17) in (3.16), an alternative representation of $\dot{\mathbf{q}}_{eb}$ can be achieved by using (2.19), (2.26) and (2.30), [20]:

$$\dot{\mathbf{q}}^e_b = \frac{1}{2}\mathbf{q}^e_b \otimes \bar{\boldsymbol{\omega}}^b_{ib} - \frac{1}{2}\bar{\boldsymbol{\omega}}^e_{ie} \otimes \mathbf{q}^e_b \tag{3.18}$$

Position, Velocity and Attitude

The complete strapdown equations in ECEF coordinates can be written as:

$$
\begin{aligned}
\dot{\mathbf{p}}^e &= \mathbf{v}^e \\
\dot{\mathbf{v}}^e &= -2\mathbf{S}(\boldsymbol{\omega}^e_{ie})\mathbf{v}^e + \mathbf{f}^e + \mathbf{g}^e(\mathbf{p}^e) \\
\dot{\mathbf{q}}^e_b &= \frac{1}{2}\mathbf{q}^e_b \otimes \bar{\boldsymbol{\omega}}^b_{ib} - \frac{1}{2}\bar{\boldsymbol{\omega}}^e_{ie} \otimes \mathbf{q}^e_b
\end{aligned}
\tag{3.19}
$$

which will be the fundament when deriving the nonlinear observer dynamics in Chapter 4.

3.1.3 Errors

With error free measurements and integration techniques, the INS would yield corresponding PVA estimates. Unfortunately, measurements from an IMU are never completely free of errors. Sensor noise, bias and calibration errors will at some degree always be present, [15]. These errors will propagate through the strapdown equations by integration and will cause the INS estimates to drift from the true values.

To prevent the INS from drifting, there is a need for an additional system such as the global navigation satellite system (GNSS). By integrating these two systems, the INS errors can by bounded by the GNSS to prevent any drift. This INS/GNSS integration is revised in Section 3.3, whereas the GNSS will be revised in the following section.

3.2 The Global Navigation Satellite System (GNSS)

A GNSS receiver computes the three dimensional position in the ECEF (or LLA) coordinate system with the use of a process called trilateration, [33]. The concept trilateration is based on knowing the distances from at least three known points to determine the position. A GNSS receiver needs to know the distances to at least four satellites to compute its location, for reasons explained later.

There exist different constellations of GNSS satellites, orbiting the earth at about 20000 [km] above the Earth's surface. These constellations are groups of $20 - 30$ satellites working together to provide regional or global coverage, with examples such as GPS, GLONASS and Galileo. Each of these systems will be explained in the following sections. For more information about GNSS, the reader is referred to [23] and [25].

3.2.1 The Fundementals

According to [25], each GNSS satellite broadcasts a signal containing its identification (ID), the date and time when the signal was broadcasted, its orbit and status. These signals are broadcasted with certain frequencies travelling at the speed of light (SOL). The SOL travels in vaccum at $3.0 \cdot 10^8$ $[m/s]$, covering about 300 $[m]$ in 1.0×10^{-6} $[s]$.

To compute the distance/range from the receiver to the satellite, the propagation time is multiplied with the SOL. This propagation time is the time elapsed from the satelletite sending the signal to the receiver recieves the signal. Since the light doesn't travel in a straight line in mediums other than vacuum, the signal is being curved and delayed traveling through the Earth's atmosphere. These delays make the range to the satellite appear longer than it really is, and will therfore be denoted as the pseudorange.

Since the GNSS receiver clock doesn't appear to be nearly as accurate as the GNSS satelitte clock, a receiver need the pseudoranges of four satellites to determine its location. With only having three satellites availible, the receiver will not have an intersecting postition with the use of the trilaterion method, caused by it's inaccurate clock. According to [25] a fourth satellite should be added to make receiver's internal clock to synchronize with the satellite clocks. This will result in an intersecting position.

The GNSS architecture consists of three main segments,[23]: The space segment, the control segment and the user segment. The space segment includes all of the satellites in the respective GNSS constellation. The control segment is a ground based segment including monitor stations, control stations and data uploading stations, where the satellites' orbit and clock are adjusted, when needed. The user segment is where the GNSS signals are being processed, i.e. GNSS receivers owned by end-users.

3.2.2 Different GNSS Systems

GPS

The US Department of Defense established the Global Positioning System (GPS) in the late 1970's, being the first GNSS system. It was originally named NAVSTAR and was intended for militrary use only, which was the case until 1983,[25].

In 1983, the GPS system was available for civilian use, where a GPS receiver will see minimum six GPS satellites at any time from any location on the Earth, given that the receiver is placed appropriate. The GPS constellation has 31 healthy satellites orbiting the Earth, with an orbit period of approximately 12 hours, [10].

GLONASS

The Global'naya Navigatsionaya Sputnikovaya Sistema (GLONASS) was, similar to the GPS system, developed with military intensions by the Soviet Union during the 1970's, [23]. The system was fully operational in 1993 and availible for civilian use in 1995. The GLONASS constellation has 28 satellites in orbit [9], whereas eight satellites are equally spaced in three orbital planes, giving 24 operational satellites for navigation. Since the GLONASS system orbits at a lower altitude than the GPS system, the orbit period for each satellite is about 11 hours and 15 minutes, [23].

Galileo

The Galileo constellation is Europe's answer to a GNSS system. Unlike GPS and GLONASS, this navigation system is intended for civilian use, when planned fully operational in 2019, [11]. The full constellation will consist of 27 satellites orbiting in three orbital planes, giving the user a position accuracy within one meter and with better focus on coverage in higher latitudes, [25].

3.2.3 Additional GNSS Abbreviations

DGNSS

Differential GNSS (DGNSS) utilize two or more receivers to improve the position accuracy of the remote GNSS receiver, [23]. The additional receivers are placed at fixed locations with known coordinates, i.e. ground base stations. These base stations transmits corrections of the pseudoranges to the remote GNSS receiver, such that any position miscalculation is removed.

HDOP

The horizontal dillution of precision (HDOP) can be obtained from the GNSS receiver. The HDOP represents a value dependent on the position accuracy in the horizontal plane, [23].

3.2.4 Errors

There are six main error sorces affecting the obtained position from a GNSS receiver [25], listed in Table 3.1.

Error source	**Error range**
Satellite clocks	±2 $[m]$
Orbit errors	±2.5 $[m]$
Ionospheric delays	±5 $[m]$
Trophosperic delays	±0.5 $[m]$
Receiver noise	±0.3 $[m]$
Multipath	±1 $[m]$

Table 3.1: GNSS Errors, [25]

Satelitte and Orbit Errors

Since the propagation time is being used to calculate the pseudorange to a satellite, errors in the clock of that specific satellite will degrade the receiver's calculated position. Slight shifts of the satellite's orbit affected by the Earth's gravitation will affect the position. Both clock and orbit errors will be adjusted and corrected by the control segment when necessary.

Atmospheric Errors

Ionospheric delays are influenced by ultraviolet rays from the sun. These rays ionize gas molecules resulting in a release of electrons. The electrons affect the GNSS satellite's broadcast signal, adding a delay to the propagation time. This delay is frequency dependent and can be eliminated by the receiver with the use of different broadcast frequencies.

Trophosperic delays are affected by temperature, humidity and pressure in the trophosphere. These delays are harder to eliminate since they vary with location, and can not be handled in the same manner as ionospheric delays. Although, according to [25], it's possible to model the trophosphere, which can be used to eliminate most of the delay.

Multipath and Receiver Errors

A multipath error is a result of the GNSS satellite signal taking multiple paths from the satelitte to the receiver, by reflection from buildings and other objects. The reflected signal will introduce a delay compared to the original signal, and may interfere with the orignal signal if the reflected signal is strong enough. These errors can be handled by making the receiver take action on the first received signal. In addition, receiver errors may be caused by an unsynchronized clock, antenna offset and additional white noise.

3.3 The INS/GNSS Integration

The INS and the GNSS are complementary in many ways. The INS provide a high data rate with a relatively low short-term error. Unfortunately, the error increases unbounded in the long-term. The GNSS provide bounded errors in the long-term, but suffer from a low data rate. The INS provide translational and rotational data, whereas the GNSS do not. Integrating these two systems would make the resulting system better.

There are different architectures on how this integration can be done, [39]. These are: Uncoupled systems, loosely coupled systems, tightly coupled systems, and deep or ultra-tightly coupled systems. A short review of each architecture will be carried out in the following sections.

3.3.1 Uncoupled

According to [39], the uncoupled system is the simplest INS/GNSS architecture, where the two systems work independently of each other. The GNSS is used to reset the INS, giving bounds to the INS errors. In the case of an GNSS outage, the INS estimates would drift unbounded. For this reason, the uncoupled architeture is not often used.

3.3.2 Loosely Coupled

In the loosely coupled scheme [39], the INS and the GNSS are put together in cascade. Estimates from both systems are provided to a INS/GNSS integrated Kalman filter (KF). This system provide simplicity and rendundancy by providing a stand-alone navigation solution from the GNSS, with the INS/GNSS integration filter acting as an alternative.

The integration filter provide estimates of the INS errors, which in turn can be used to correct the INS for better performance. In addition, the GNSS uses estimates from the INS to aid the satellite signal.

The main problem with this architecture is a result by having a two KF in series - a GNSS receiver is normally implemented with a KF to provide position estimates.. The architecture may therefore violate the requirement of delivering uncorrelated inputs to the second KF, [39]. The loosely coupled architecture can be shown in Figure 3.1.

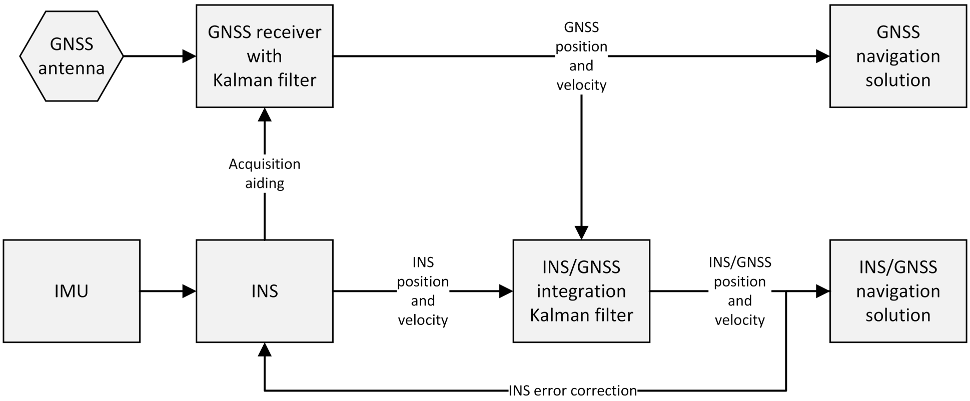

GNSS antenna
GNSS receiver with Kalman filter
GNSS position and velocity
GNSS navigation solution
Acquisition aiding
IMU
INS
INS position and velocity
INS/GNSS integration Kalman filter
INS/GNSS position and velocity
INS/GNSS navigation solution
INS error correction

Figure 3.1: INS/GNSS: Loosely coupled architecture, [39]

3.3.3 Tightly Coupled

In the tightly coupled scheme [39], the KF in the GNSS receiver is implemented within the INS/GNSS integration block as one KF. The INS/GNSS block will then use pseudo-range and pseudo-range rate measurements, obtained from the tracking-loops within the GNSS receiver. As with the loosely coupled scheme, the INS is being corrected by the INS/GNSS estimates while the INS provides additional aiding to the GNSS receiver.

The main benefit lies in the combination of the two KFs from the loosely coupled scheme, removing the previous problem in the Kalman filter design. The use of pseudo-range data from the GNSS will also make this architecture functionable in the case of having only one available satellite, at the expense of degraded accuracy.

Compared to the loosely coupled scheme the tightly coupled scheme will provide better performance, both in terms of robustness and accuracy. The tightly coupled architecture is illustrated in Figure 3.2.

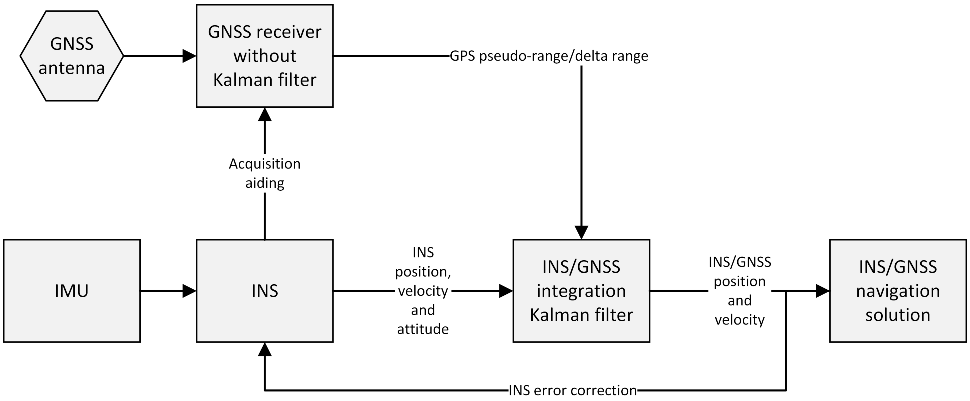

GNSS antenna
GNSS receiver without Kalman filter
GPS pseudo-range/delta range
Acquisition aiding
IMU
INS
INS position, velocity and attitude
INS/GNSS integration Kalman filter
INS/GNSS position and velocity
INS/GNSS navigation solution
INS error correction

Figure 3.2: INS/GNSS: Tightly coupled architecture, [39]

3.3.4 Deep or Ultra-Tightly Coupled

In the ultra-tightly coupled scheme the signal tracking-loops within the GNSS receiver are integrated with the INS into a single complex Kalman filter. The benefits from this solution lies in a better signal-to-noise ratio (SNR), an improvement in multi-path errors, and better handling when GNSS signals are being jammed. This archtecture may be hard to achieve as the source-code within the GNSS receiver may be difficult, or even impossible, to access. The deeply integrated INS/GNSS scheme is illustrated in Figure 3.3.

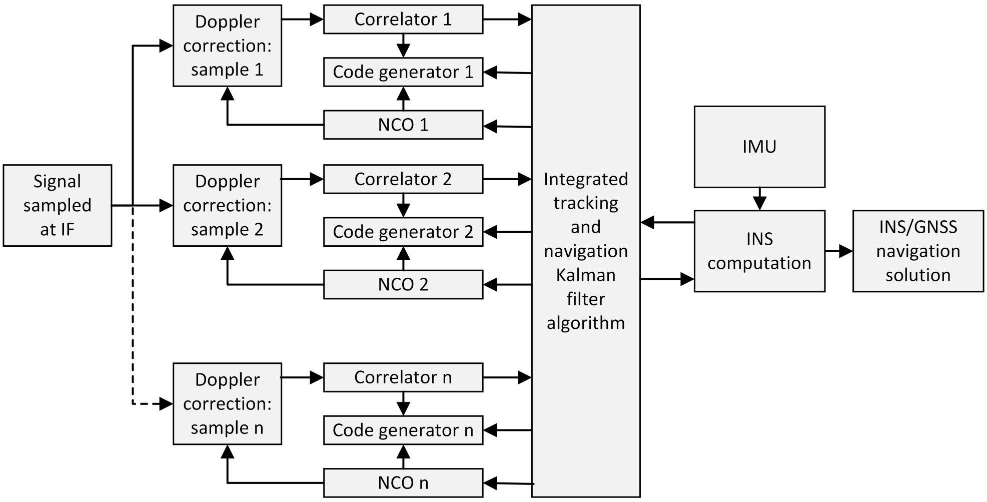

Doppler correction: sample 1
Correlator 1
Code generator 1
NCO 1
Signal sampled at IF
Doppler correction: sample 2
Correlator 2
Code generator 2
NCO 2
Doppler correction: sample n
Correlator n
Code generator n
NCO n
Integrated tracking and navigation Kalman filter algorithm
IMU
INS computation
INS/GNSS navigation solution

Figure 3.3: INS/GNSS: Deeply coupled architecture, [39]

3.4 The Inertial Measurement Unit (IMU)

The Inertial Measurement Unit (IMU) normally consist of three accelerometers and three gyroscopes providing information in six degrees of freedom (6-DOF), see [15] and [39]. The accelerometers and gyroscopes are presented in three pairs, where each accelerometer and gyrocscope pair describe translational and rotational information about one axis in the body-fixed frame. Together, the three pairs give the complete knowledge about the inertia in three dimensions.

Today, the inertial measurement technology can be found everywhere: In game controllers, mobile cellphones and other electronical gadgets. The result of this availabilty has made the cost of this technology to rapidly fall, making it possible to integrate an IMU with a GNSS receiver with low costs. The components and technology within and behind an IMU will be explained in the following sections.

3.4.1 Accelerometers

The Fundementals

An accelerometer detects translational motion and is based on Newton's second law on a rigid body, [39].

Newton's second law is defined as:

$$F = ma \tag{3.20}$$

where F is the sum of forces, m is the mass and a is the total acceleration.

Equation (3.20) can further be split into:

$$F = ma = mf + mg \tag{3.21}$$

where g is the acceleration caused by gravity and f is the acceleration produced by other forces than gravity, also denoted as the specific force.

The total forces acting upon the rigid body can be expressed in terms of forces per unit mass, which redefines (3.21) to:

$$F = a = f + g \tag{3.22}$$

Rearranging (3.22) yields the output from an ideal accelerometer:

$$f = a - g \tag{3.23}$$

An single accelerometer experiencing free fall in vacuum, normal to the ground, will yield $f = 0$ as $a = g$. In the case of being stationary ($a = 0$), the accelerometer output will yield $f = -g$. This will be the case since gravity keeps it stationary by acting normal upwards from the medium it rests on.

Different Technologies

According to [15], there are different types of accelerometers such as mechanical and solid-state accelerometers. The mechanical accelerometer can be defined as a mass-spring-damping type (MSD). The MSD relates to a proof mass being defined at an initial position, hold by a spring and a damper. Deflection by the mass with respect to its initial position will result in a spring force proportional to the acceleration being measured.

Solid-state accelerometers are small and reliable sensors which have gained intensive research during the last years. One solid-state accelerometer is a vibratory device commonly based on quarts crystal technology. Two quartz chrystal beams are mounted symmetrically vibrating at different resonant frequencies. When an acceleration is inflicted to the device, one beam is strechted and the other is compressed, resulting in the beams having different frequencies. The frequency difference between the two beams is proportional to the acceleration experienced by the device. More on this and other technologies can be found in [39].

3.4.2 Gyroscopes

The Fundementals

A gyroscope can either detect angle displacement or angular rate about an axis. The latter type of gyroscopes are normally used in an IMU. The conventional gyroscope utilizes the inertia of a wheel spinning at a high speed, supported by a pair of gimbals, [39]. The inertia of the spinning wheel define a fixed direction in space.

The equation below describe the behaviour of a gyroscope under the influence of a torque, [39]:

$$\mathbf{T} = \frac{d\mathbf{H}}{dt} = \frac{d(\mathbf{I}\boldsymbol{\omega})}{dt} = \mathbf{I}\boldsymbol{\alpha} \tag{3.24}$$

where $\mathbf{T}$ describe the tourque, $\mathbf{H}$ the angular momentum, $\mathbf{I}$ the moment of inertia, $\boldsymbol{\omega}$ the angular velocity and $\boldsymbol{\alpha}$ the angular acceleration.

When a torque $\mathbf{T}$ is applied perpendicular to the axis of rotation, a motion called precession occur. The precession, denoted $\boldsymbol{\omega}_p$, is normal to the spin axis and the torque $\mathbf{T}$, which can be defined in the following equation using Newton's second law, [39]:

$$\mathbf{T} = \boldsymbol{\omega}_p \times \mathbf{H} \tag{3.25}$$

From the precession principle accurate angular velocities can be obtained. For detailed information about how a gyroscope works, the reader is referred to [39].

Different Technologies

In addition to the conventional gyroscope, there exist other types. One of them is the ring laser gyroscope (RLG). The RLG is according to [39] based on the concept of having two independent laser beams being reflected by three mirrors in a trangular fashion back to the starting point.

On beam travels clockwise, while the other travels anti-clockwise. When the RLG is stationary, both laser beams have the same frequency. But in the case of being rotated perpendicular to the plane covered by the laser beams, a difference in the path length of the two lasers occur. This result in a frequency difference which can be used to determine the angular rate.

Another type is the fibre optic gyroscope (FOG), where the technology is based on a light source sending a light beam through a beam splitter, [39]. The splitted light travels in opposite directions around a fibre optic coil, before returning back to a second splitter. This second beam splitter sends the returning light to a photo-detector, where any phase changes between the two beams are detected. Rotating the FOG normal to the light plane result in one beam traveling longer than the other, thus a change in phase is detected and the angular rate can be calculated.

3.4.3 Error Sources

The gyroscope and accelerometer measurements are subject to different error sources, such as noises, biases, scale factors and axis misalignments, see [15, 39, 8]. The handling of these error sources can either be done using different algorithms online, i.e. noise filtering, bias estimation or coning/sculling algorithms. There also exists different calibration methods offline, but require expensive calibration equipment. Thus, choosing an appropriate IMU with low errors is essential.

Sensor Noise

Due to electrical interference in the sensors, the measurments will be affected by noise. The integration of noisy measurements will lead to values which walk randomly. In the case of acquiring a position estimate, a second integration will lead to a rapidly drifting estimate. To remove the effect of noise, one can either use analog or digital low-pass filters. The downside to this approach, is additional delay on the filtered signal. Therefore, choosing a sensor with a high signal-to-noise ratio (SNR) is important.

Sensor Bias

The sensor bias has a big impact when integrating the measurements in an INS. Both the gyroscope and accelerometer biases are temperature dependent, where the latter is most affected by temperature variations. For this reason, a look-up table for the accelerometer can be made to correct the sensor at different temperatures. Additionally, there exist a random walk component making the bias change randomly with time. This component can not be removed using offline calibrations, making online calibrations the best approach.

Scale Factors

The scale factor for each sensor axis is important to scale the sensors such that the output, with the selected unit, represent the real translational or rotational motion. For an accelerometer utilized to decide orientation, an incorrect chosen scale factor will affect the attitude dramatically. An IMU will be calibrated at the factory before being delivered to the customer, with the default scale factors equal to one. To ensure that this default scale factor is correct, appropriate calibration equipment is needed.

Axis Misalignments

The accelerometers and gyroscopes represent translational and rotational motion about three axes. Ideally, these axes should be perpendicular to each other. Unfortunately, this isn't always the case due to inaccurate sensor design and mounting. This means that for each axis, movement in the other to axes would cause a small impact on the first axis. Cross-coupling between the axes can degrade the system performance, especially in the case of high frequency vibrations. Vibrations with frequencies higher than half the IMU sampling rate will result in drifting estimates, [26]. To compensate for this error, a coning/sculling algorithm is needed, [39].

3.4.4 MEMS Technology

The micro-machined electromechanical system (MEMS) has changed the industry in terms of low-cost sensors for inertial navigation. MEMS devices have few components and is based on a solid-state archtecture, using silicon as the base material, [39]. The use of silicon result in robust and cost-effective components, with devices offering low weight, low power consumption and low maintenance.

Accelerometers

There are two classes of MEMS based accelerometers, [39]: The first is the displacement of a proof mass, which was explained in the introduction of Section 3.4, only being manufactured with silicon materials. The second is the vibrating element, similar to the quartz vibrating technology described earlier, where frequency changes caused by tension in the vibrating element is proportional to the acceleration.

Gyroscopes

A MEMS based gyroscope uses the Coriolis acceleration's effect on a vibrating proof mass to sense angular rotation, [39]. With this approach, no rotating parts are needed, resulting in very small sensor. There exist different types of MEMS gyroscopes, i.e. resonant ring and tuning fork gyroscopes. The tuning fork gyroscope utilizes a pair of masses which oscillate in opposite directions. When applying a rotational motion about the axis perpendicular to the velocities of the masses, the Coriolis force acts upon the masses, which in turn can be utilized to calculate the angular rate.

3.5 The Gyrocompass

A gyrocompass seeks the true north, [36]. It is not affected by magnetic disturbances created by ferromagnetic materials, as a magnetic compass would be. For this reason the gyrocompass is widely used to determine the heading in marine applications. The gyrocompass is based on the same principles as a gyroscope, but should not be mixed.

3.5.1 The Operation

The gyrocompass rely on the Earth's rotation, gravity, gyroscopic inertia and precession, see Section 3.4.2. To end up with a gyrocompass a gyroscope is mounted on a sphere, denoted gyrosphere, and supported on a vertical ring. The north seeking process is done in the following order, [14]:

The gyroscope is forced to align itself horizontally by seeking the meridian (line of a certain longitude), with the use of a weight at the bottom of the vertical ring. To prevent any oscillatory movements, a second weight is mounted on the side of the ring.

Secondly, the gyrocompass is made to seek the true north by applying a force against the spin axis of the rotating gyroscope. This force is made by reservoirs filled with mercury, denoted mercury ballistics. The ballistics is placed slightly to the east of the vertical plane such that the center of gravity (CG) of the gyroscope coincides with the ballistics' CG.

This results in precession, which makes the spin axis trace a spiral before settling down pointing towards the true north, [14].

Chapter 4

Nonlinear Observer Design

This chapter will present the nonlinear observer used in the simulations and experimental tests in Chapter 6 and 7. The observer was chosen to be expressed in the ECEF frame, due to the access of GNSS measurements in LLA coordinates. The GNSS measurements could, according to Marine Technologies LLC, be delayed before reaching nonlinear observer. These delays could affect the stability of the onlinear observer, due to a resulting inconsistent GNSS rate. Motivated by this reason, a dynamic GNSS gain will be proposed, such that the observer would handle inconsistent GNSS measurement rates. The observer dynamics and sensor configuration will be revised first. Secondly, the complete nonlinear observer will be presented, both in the continuous and discrete time domain. Lastly, the tuning of the observer will be given.

4.1 Observer Dynamics

The strapdown equations for position, velocity and attitude (PVA) expressed in ECEF coordinates are given in (3.19), from Section 3.1.2. By assuming that both the accelerometers and gyroscopes are subject to sensor biases, the complete dynamic equation set is given as, [20]:

$$\begin{aligned}
\dot{\mathbf{p}}^e &= \mathbf{v}^e && (4.1)\\
\dot{\mathbf{v}}^e &= -2\mathbf{S}(\boldsymbol{\omega}^e_{ie})\mathbf{v}^e + \mathbf{f}^e + \mathbf{g}^e(\mathbf{p}^e) && (4.2)\\
\dot{\mathbf{q}}^e_b &= \frac{1}{2}\mathbf{q}^e_b \otimes \bar{\boldsymbol{\omega}}^b_{ib} - \frac{1}{2}\bar{\boldsymbol{\omega}}^e_{ie} \otimes \mathbf{q}^e_b && (4.3)\\
\dot{\mathbf{b}}^b_g &= 0 && (4.4)\\
\dot{\mathbf{b}}^b_a &= 0 && (4.5)
\end{aligned}$$

where $\dot{\mathbf{b}}^b_g$ and $\dot{\mathbf{b}}^b_a$ represent the differential equations for gyroscope and accelerometer bias, respectively.

4.2 Sensor Configuration

Before the nonlinear PVA observer is derived, the following sensors are assumed availible:

- The posistion measurement (GNSS receiver): $\mathbf{p}^e_{GNSS} = \mathbf{p}^e$ or $\mathbf{p}^l_{GNSS} = \mathbf{p}^l$
- The heading measurement (Gyrocompass): ψ_{GYC}
- The acceleration measurement (IMU): $\mathbf{f}^b_{IMU} = \mathbf{f}^b + \mathbf{b}^b_a$
- The angular velocity measurement (IMU): $\boldsymbol{\omega}^b_{ib,IMU} = \boldsymbol{\omega}^b_{ib} + \mathbf{b}^b_g$

The velocity measurement $\mathbf{v}^e_{GNSS} = \mathbf{C}_v\mathbf{v}^e$ from the GNSS receiver is not utilized in this thesis, but its presence will be included in the nonlinear observer equations. When the velocity measurement is availible, $\mathbf{C}_v$ may be the identity matrix or another matrix with satisfying dimensions.

4.3 The Nonlinear Observer

The attitude observer presented in [20] is based on having inertial vector measurements $\mathbf{f}^b$ and $\mathbf{m}^b$ from accelerometers and magnetometers, respectively. Both measurement vectors have corresponding reference vectors $\mathbf{f}^e$ and $\mathbf{m}^e$. With theses quantities, a vector-based[1] attitude observer can be presented, see [30] and [24].

[1] The inertial vector measurement and its corresponding reference vector is denoted as a vector pair

Since magnetometer measurements aren't availible, the observer had to be complemented in another way. The proposed method in the next section utilize measurements from a gyrocompass to create substitutes for the vector pair $\mathbf{m}^b$ and $\mathbf{m}^e$. The block diagram of the modified nonlinear observer is given in Figure 4.1.

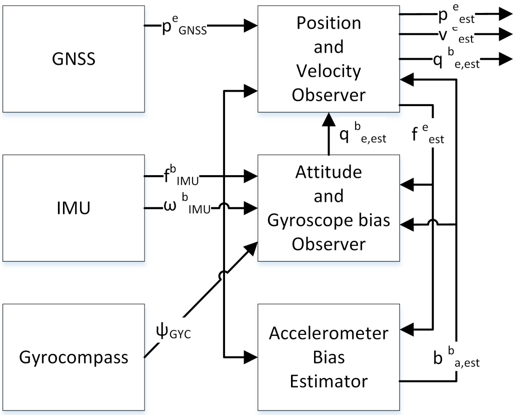

GNSS
p^e_GNSS
Position and Velocity Observer
p^e_est
v^e_est
q^b_e,est
IMU
f^b_IMU
ω^b_IMU
Attitude and Gyroscope bias Observer
q^b_e,est
f^e_est
Gyrocompass
ψGYC
Accelerometer Bias Estimator
b^b_a,est

Figure 4.1: The Modified Observer of Grip

4.3.1 The Gyrocompass Substitute

This method assumes partial or complete knowledge of the Euler angle vector $\mathbf{\Theta}_{nb} = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^\top$ and the latitude and longitude vector $\mathbf{\Theta}_{en} = \begin{bmatrix}\mu & \varphi\end{bmatrix}^\top$. Both $\mathbf{\Theta}_{nb}$ and $\mathbf{\Theta}_{en}$ are used to derive a vector pair in $\{b\}$ and $\{e\}$, with the use of the north-seeking unit vector $\mathbf{c}^n = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^\top$ in $\{n\}$.

The Inertial Measurement Vector

With the knowledge of $\mathbf{\Theta}_{nb}$, the rotation matrix $\boldsymbol{R}_b^n(\mathbf{\Theta}_{nb})$ can be computed. By rotating the true north vector $\mathbf{c}^n$ to $\{b\}$, the inertial measurement vector $\mathbf{c}^b = \begin{bmatrix}c_x^b & c_y^b & c_z^b\end{bmatrix}^\top$ can be given as:

$$\begin{aligned}
\mathbf{c}^b &= \boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb})^\top \mathbf{c}^n \\
&\Updownarrow \\
\mathbf{c}^b &= \begin{bmatrix} c\psi c\theta & s\psi c\theta & -s\theta \\ -s\psi c\phi + c\psi s\theta s\phi & c\psi c\phi + s\phi s\theta s\psi & c\theta s\phi \\ s\psi s\phi + c\psi c\phi s\theta & -c\psi s\phi + s\theta s\psi c\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
&\Downarrow \\
\mathbf{c}^b &= \begin{bmatrix} c\psi c\theta \\ -s\psi c\phi + c\psi s\theta s\phi \\ s\psi s\phi + c\psi c\phi s\theta \end{bmatrix}
\end{aligned} \tag{4.6}$$

The inertial measurement vector $\mathbf{c}^b$ can replace the inertial magnetometer measurement $\mathbf{m}^b$.

The Euler Angle Vector

The yaw angle $\psi \in [-\pi, \pi]$ can be transformed from the heading angle $\psi_{GYC} \in [0, \infty)$, which is provided by the gyrocompass. The following relationship between ψ and ψ_{GYC} can be given, assuming that both angles are given in radians:

$$\psi = \psi_{GYC} - 2\pi \cdot \mathrm{round}(\frac{\psi_{GYC}}{2\pi}) \tag{4.7}$$

where round$(\cdot)$ is a function rounding to the nearest integer.

The roll and pitch angles ϕ and θ can be assumed small, i.e. $\phi \approx 0$ and $\theta \approx 0$, which should be a valid asumption for marine surface vessels, [15].

Another way to determine ϕ and θ is based on the assumption of the vessel being at rest, i.e. in DP or at low speeds, where the non-gravitational accelerations are zero, [15] . This method assumes zero bias and noise on the accelerometer measurements:

$$\mathbf{f}_{imu}^b \approx -\mathbf{g}^b = \begin{bmatrix} g_x & g_y & g_z \end{bmatrix}^\top \tag{4.8}$$

where the roll and pitch angles can be calculated as:

$$
\begin{aligned}
\phi &\approx \arctan(\frac{g_y}{g_z}) && (4.9)\\
\theta &\approx -\arctan(\frac{g_x}{\sqrt{g_y^2 + g_z^2}}) && (4.10)
\end{aligned}
$$

Since the accelerometer is influenced by noise, the latter approach presented by (4.8-4.10) will not be utilized in the observer.

The Reference Vector

With the knowledge of $\mathbf{\Theta}_{en}$, and thus $\mathbf{R}_n^e(\mathbf{\Theta}_{en})$, the true north vector $\mathbf{c}^n$ can be rotated to $\{e\}$ to obtain $\mathbf{c}^e = \begin{bmatrix} c_x^e & c_y^e & c_z^e \end{bmatrix}^\top$:

$$
\begin{aligned}
\mathbf{c}^e &= \boldsymbol{R}_n^e(\mathbf{\Theta}_{en})\mathbf{c}^n\\
&\Updownarrow\\
\mathbf{c}^e &= \begin{bmatrix} -c\varphi s\mu & -s\varphi & -c\varphi c\mu \\ -s\varphi s\mu & c\varphi & -s\varphi c\mu \\ c\mu & 0 & -s\mu \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad (4.11)\\
&\Downarrow\\
\mathbf{c}^e &= \begin{bmatrix} -c\varphi s\mu \\ -s\varphi s\mu \\ c\mu \end{bmatrix}
\end{aligned}
$$

The reference vector $\mathbf{c}^e$ can replace the magnetic field reference vector $\mathbf{m}^e$. If the nonlinear observer is represented in the NED frame, the reference vector is given as $\mathbf{c}^n = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top$.

The Latitude and Longitude Vector

The GNSS position measurements can be used to set the latitude and longitude vector $\mathbf{\Theta}_{en}$. In cases of GNSS outage, $\mathbf{\Theta}_{en}$ should be set to the last known value until the GNSS return. This assumption is valid when navigating in small areas, since the reference vector $\mathbf{c}^e$ is nearly constant between each GNSS sample, even in the case of GNSS noise.

4.3.2 Attitude and Gyroscope Bias Estimation

When deriving the attitude and gyroscope bias observer, the accelerometer bias $\mathbf{b}_a^b$ is assumed to be zero, i.e. $\mathbf{f}_{IMU}^b = \mathbf{a}^b - \mathbf{g}^b$.

Known Specific Force

The attitude and gyroscope bias observer, based on (4.3) and (4.4), is presented in [20] as:

$$
\begin{aligned}
\dot{\hat{\mathbf{q}}}_b^e &= \frac{1}{2}\hat{\mathbf{q}}_b^e \otimes (\bar{\boldsymbol{\omega}}_{ib,IMU}^b - \bar{\hat{\mathbf{b}}}_b^e + \bar{\boldsymbol{\sigma}}) - \frac{1}{2}\bar{\boldsymbol{\omega}}_{ie}^e \otimes \hat{\mathbf{q}}_b^e && (4.12)\\
\dot{\hat{\mathbf{b}}}_g^b &= \mathbf{Proj}(\hat{\mathbf{b}}_g^b, -k_I\boldsymbol{\sigma}) && (4.13)
\end{aligned}
$$

where $\mathbf{Proj}(\hat{\mathbf{b}}_g^b, -k_I\boldsymbol{\sigma})$ is the projection of $\hat{\mathbf{b}}_g^b$ within the the compact set $\|\hat{\mathbf{b}}_g^b\| \leq M_{\hat{b}_g}$, where $k_I > 0$ is the bias tuning parameter and $M_{\hat{b}_g}$ is chosen to be slightly larger than the real upper bound M_{b_g}.

The attitude observer is based on the comparison between two body-fixed measurement vectors with respect to their corresponding vectors in the reference frame. Replacing the magnetometer vector pair, $\mathbf{m}^b$ and $\mathbf{m}^e$, with the gyrocompass substitute, $\mathbf{c}^b$ and $\mathbf{c}^e$, gives the following injection term:

$$
\boldsymbol{\sigma} := k_1\underline{\mathbf{f}^b} \times \mathbf{R}_b^e(\hat{\mathbf{q}}_b^e)^\top \underline{\mathbf{f}^e} + k_2\underline{\mathbf{c}^b} \times \mathbf{R}_b^e(\hat{\mathbf{q}}_b^e)^\top \underline{\mathbf{c}^e} \tag{4.14}
$$

where $\underline{\mathbf{c}^b}$, $\underline{\mathbf{c}^e}$, $\underline{\mathbf{f}^b}$ and $\underline{\mathbf{f}^e}$ are unit length vectors defined as:

$$
\underline{\mathbf{c}^b} = \frac{\mathbf{c}^b}{\|\mathbf{c}^b\|}, \ \underline{\mathbf{c}^e} = \frac{\mathbf{c}^e}{\|\mathbf{c}^e\|}, \ \underline{\mathbf{f}^b} = \frac{\mathbf{f}_{IMU}^b}{\|\mathbf{f}_{IMU}^b\|}, \ \underline{\mathbf{f}^e} = \frac{\mathbf{f}^e}{\|\mathbf{f}^e\|} \tag{4.15}
$$

where $k_1 > 0$ and $k_2 > 0$ are tuning parameters.

The attitude observer presented in this section is a stand-alone attitude observer, made possible with access to the specific force $\mathbf{f}^e$. This force is in practical cases not known, but can be estimated, which will be the case in the next section.

Unknown Specific Force

Since the specific force $\mathbf{f}^e$ is an unknown quantity, $\mathbf{f}^e$ is replaced by its estimate $\hat{\mathbf{f}}^e$. The specific force estimation is made possible by combining the proposed attitude and gyroscope bias observer (4.12-4.13) with the position and velocity observer, which will be revised in the next section.

In the case of the estimated specific force $\hat{\mathbf{f}}^e$, the attitude and gyroscope observer is defined as, [20]:

$$\begin{aligned} \dot{\hat{\mathbf{q}}}_b^e &= \frac{1}{2}\hat{\mathbf{q}}_b^e \otimes (\bar{\boldsymbol{\omega}}_{ib,IMU}^b - \bar{\hat{\mathbf{b}}}_b^e + \bar{\hat{\boldsymbol{\sigma}}}) - \frac{1}{2}\bar{\boldsymbol{\omega}}_{ie}^e \otimes \hat{\mathbf{q}}_b^e \quad (4.16) \\ \dot{\hat{\mathbf{b}}}_g^b &= \mathbf{Proj}(\hat{\mathbf{b}}_g^b, -k_I\hat{\boldsymbol{\sigma}}) \quad (4.17) \end{aligned}$$

where

$$\hat{\boldsymbol{\sigma}} := k_1\underline{\mathbf{f}^b} \times \mathbf{R}_b^e(\hat{\mathbf{q}}_b^e)^\top \underline{\hat{\mathbf{f}}^e} + k_2\underline{\mathbf{c}^b} \times \mathbf{R}_b^e(\hat{\mathbf{q}}_b^e)^\top \underline{\mathbf{c}^e} \quad (4.18)$$

with the following unit length vectors

$$\underline{\mathbf{c}^b} = \frac{\mathbf{c}^b}{\|\mathbf{c}^b\|},\ \underline{\mathbf{c}^e} = \frac{\mathbf{c}^e}{\|\mathbf{c}^e\|},\ \underline{\mathbf{f}^b} = \frac{\mathbf{f}_{IMU}^b}{\|\mathbf{f}_{IMU}^b\|},\ \underline{\hat{\mathbf{f}}^e} = \frac{\hat{\mathbf{f}}^e}{max\{\|\hat{\mathbf{f}}^e\|, \delta\}} \quad (4.19)$$

The delta parameter δ is defined such that $0 < \delta < m_f$, and makes $\hat{\mathbf{f}}^e$ well-defined in cases of being near zero. The parameter m_f is defined as $0 < m_f \leq \|\mathbf{f}_{IMU}^b\| \leq M_f$, where $\|\mathbf{f}_{IMU}^b\| = \|\mathbf{g}^e(\mathbf{p}^e)\|$ when the vessel experience zero non-gravitational accelerations. The constant M_f is the upper bound on the norm of the accelerometer measurement $\mathbf{f}_{IMU}^b$.

4.3.3 Position and Velocity Estimation

When deriving the position and velocity observer, the accelerometer bias $\mathbf{b}_a^b$ is not present, i.e. $\mathbf{f}_{IMU}^b = \mathbf{a}^b - \mathbf{g}^b$.

With GNSS Velocity

The position and velocity observer from [20] is presented as:

$$\begin{aligned}
\dot{\hat{\mathbf{p}}}^e &= \hat{\mathbf{v}}^e + \theta_{GNSS}\mathbf{K}_{pp}(\mathbf{p}^e_{GNSS} - \hat{\mathbf{p}}^e) + \mathbf{K}_{pv}(\mathbf{v}^e_{GNSS} - \mathbf{C}_v\hat{\mathbf{v}}^e) && (4.20)\\
\dot{\hat{\mathbf{v}}}^e &= -2\mathbf{S}(\boldsymbol{\omega}^e_{ie})\hat{\mathbf{v}}^e + \hat{\mathbf{f}}^e + \mathbf{g}^e(\hat{\mathbf{p}}^e)\\
&\quad +\theta^2_{GNSS}\mathbf{K}_{vp}(\mathbf{p}^e_{GNSS} - \hat{\mathbf{p}}^e) + \theta_{GNSS}\mathbf{K}_{vv}(\mathbf{v}^e_{GNSS} - \mathbf{C}_v\hat{\mathbf{v}}^e) && (4.21)\\
\dot{\boldsymbol{\xi}} &= -\mathbf{R}^e_b(\hat{\mathbf{q}}^e_b)\mathbf{S}(\hat{\boldsymbol{\sigma}})\mathbf{f}^b_{IMU}\\
&\quad +\theta^3_{GNSS}\mathbf{K}_{\xi p}(\mathbf{p}^e_{GNSS} - \hat{\mathbf{p}}^e) + \theta^2_{GNSS}\mathbf{K}_{\xi v}(\mathbf{v}^e_{GNSS} - \mathbf{C}_v\hat{\mathbf{v}}^e) && (4.22)\\
\hat{\mathbf{f}}^e &= \mathbf{R}^e_b(\hat{\mathbf{q}}^e_b)\mathbf{f}^b_{IMU} + \boldsymbol{\xi} && (4.23)
\end{aligned}$$

where the tuning parameter $\theta_{GNSS} \geq 1$ is chosen to guarantee stability.

The tuning matrices $\mathbf{K}_{pp}$, $\mathbf{K}_{pv}$, $\mathbf{K}_{vp}$, $\mathbf{K}_{vv}$, $\mathbf{K}_{\xi p}$ and $\mathbf{K}_{\xi v}$ are designed such that the matrix $\mathbf{H}_K = \mathbf{A} - \mathbf{KC}$ is Hurwitz and $\|\mathbf{H}_K(s)\|_\infty < \rho$, where ρ is chosen sufficiently low.

The matrices $\mathbf{A}$, $\mathbf{K}$ and $\mathbf{C}$ are defined as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}, \ \mathbf{K} = \begin{bmatrix} \mathbf{K}_{pp} & \mathbf{K}_{pv} \\ \mathbf{K}_{vp} & \mathbf{K}_{vv} \\ \mathbf{K}_{\xi p} & \mathbf{K}_{\xi v} \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{C}_v & \mathbf{0}_{3\times3} \end{bmatrix} \qquad (4.24)$$

Without GNSS Velocity

In the case of not having access to the velocity measurement $\mathbf{v}^e_{GNSS}$, the observer can be reduced to:

$$\begin{aligned}
\dot{\hat{\mathbf{p}}}^e &= \hat{\mathbf{v}}^e + \theta_{GNSS}\mathbf{K}_{pp}(\mathbf{p}^e_{GNSS} - \hat{\mathbf{p}}^e) && (4.25)\\
\dot{\hat{\mathbf{v}}}^e &= -2\mathbf{S}(\boldsymbol{\omega}^e_{ie})\hat{\mathbf{v}}^e + \hat{\mathbf{f}}^e + \mathbf{g}^e(\hat{\mathbf{p}}^e)\\
&\quad +\theta^2_{GNSS}\mathbf{K}_{vp}(\mathbf{p}^e_{GNSS} - \hat{\mathbf{p}}^e) && (4.26)\\
\dot{\boldsymbol{\xi}} &= -\mathbf{R}^e_b(\hat{\mathbf{q}}^e_b)\mathbf{S}(\hat{\boldsymbol{\sigma}})\mathbf{f}^b_{IMU}\\
&\quad +\theta^3_{GNSS}\mathbf{K}_{\xi p}(\mathbf{p}^e_{GNSS} - \hat{\mathbf{p}}^e) && (4.27)\\
\hat{\mathbf{f}}^e &= \mathbf{R}^e_b(\hat{\mathbf{q}}^e_b)\mathbf{f}^b_{IMU} + \boldsymbol{\xi} && (4.28)
\end{aligned}$$

where the terms dependent on $\mathbf{v}^e_{GNSS}$ are removed, i.e. $\mathbf{K}_{pv} = \mathbf{K}_{vv} = \mathbf{K}_{\xi v} = \mathbf{0}_{3\times 3}$ and $\mathbf{C}_v = \mathbf{0}$, using appropriate dimensions of $\mathbf{0}$.

4.3.4 Accelerometer Bias Estimation

The accelerometer bias $\mathbf{b}^b_a = \begin{bmatrix} b^b_{a,x} & b^b_{a,y} & b^b_{a,z} \end{bmatrix}^\top$ is always present in real applications, and will therefore be included from this point.

The accelerometer measurement is defined as:

$$\mathbf{f}^b_{IMU} = \mathbf{a}^b - \mathbf{g}^b + \mathbf{b}^b_a$$

Estimating the accelerometer bias is the most challenging part when designing the complete nonlinear observer. Two methods were proposed, both working ideally when the neccessary assumptions/conditions are met.

The Vector Bias Estimation Method

Grip et al. [20] presents a method based on an over-parmeterized observer, which can be used for all vector measurements with corresponding reference vectors, i.e. vector pairs for magnetometers and accelerometers.

The body-fixed biased vector measurement is defined as $\mathbf{v}^b_m = \mathbf{v}^b + \mathbf{b}^b$, where $\mathbf{v}^b$ is the unbiased vector and $\mathbf{b}^b$ is the bias vector. The corresponding bias-free reference vector is $\mathbf{v}^e_m = \mathbf{v}^e = \mathbf{R}^e_b \mathbf{v}^b$.

The observer can be derived using the squared norms $||\mathbf{v}^e_m||^2$, $||\mathbf{v}^b_m||^2$ and $p = ||\mathbf{b}^b||^2$, and the relationship between them:

$$\begin{aligned} y &= ||\mathbf{v}^e_m||^2 + ||\mathbf{v}^b_m||^2 \qquad (4.29) \\ &\Updownarrow \\ y &= p - 2(\mathbf{v}^b_m)^\top \mathbf{b}^b \ = \ \boldsymbol{\phi}^\top \mathbf{w} \qquad (4.30) \end{aligned}$$

where $\boldsymbol{\phi} = \begin{bmatrix} 1 & -2(\mathbf{v}^b_m)^\top \end{bmatrix}^\top$ and $\mathbf{w} = \begin{bmatrix} p & (\mathbf{b}^b)^\top \end{bmatrix}^\top$.

The over-parameterized observer is given as:

$$\dot{\hat{\mathbf{w}}} = \boldsymbol{\Upsilon}\boldsymbol{\phi}(\hat{y} - \boldsymbol{\phi}^\top\hat{\mathbf{w}}) \tag{4.31}$$

where $\boldsymbol{\Upsilon}$ is a constant positive-definite matrix. With a constant gain $\boldsymbol{\Upsilon}$, this particular observer is represented by linear regression. The recursive least-squares estimate of $\mathbf{w}$ can be obtained if the constant gain is replaced by a time-varying gain, which satisfies:

$$\dot{\boldsymbol{\Upsilon}} = \alpha\boldsymbol{\Upsilon} - \boldsymbol{\Upsilon}\boldsymbol{\phi}\boldsymbol{\phi}^\top\boldsymbol{\Upsilon} \tag{4.32}$$

where $\alpha \geq 0$ is the forgetting factor.

The vector based estimation method (4.31) can be defined such that the estimate $\hat{\mathbf{w}}$ stays within the compact set $\|\hat{\mathbf{w}}\| \leq M_{\hat{w}}$, where $M_{\hat{w}} \geq M_w = \|\mathbf{w}\|$:

$$\dot{\hat{\mathbf{w}}} = \mathbf{Proj}(\hat{\mathbf{w}}, \boldsymbol{\Upsilon}\boldsymbol{\phi}(\hat{y} - \boldsymbol{\phi}^\top\hat{\mathbf{w}})) \tag{4.33}$$

where $\mathbf{Proj}(\cdot,\cdot)$ is the same projection method used in the gyroscope bias observer, [20].

One requirement that is needed for the vector bias method to converge, is the satisfaction of the persistency-of-excitation (PE) condition. The PE condition says that the input vector $\boldsymbol{\phi} = \begin{bmatrix} 1 & -2(\mathbf{v}_m^b)^\top \end{bmatrix}^\top$ must vary independently in a sufficient way. This makes the vector bias estimation method dependent on the application.

The Mean Filter Estimation Method

Another way to estimate the accelerometer bias is under the assumption that the roll and pitch angles ϕ and θ have the mean values $\phi_{mean} = 0$ and $\theta_{mean} = 0$, over a sufficient long time period. This should be valid for an evenly loaded vessel when it's stationary, only exposed to waves.

The assumption is mathematically expressed as:

$$\phi_{mean} = \frac{1}{t_w}\int_{t=t_s}^{t_s+t_w} \phi(t)dt = 0 \tag{4.34}$$

$$\theta_{mean} = \frac{1}{t_w}\int_{t=t_s}^{t_s+t_w} \theta(t)dt = 0 \tag{4.35}$$

where t_s is the start of the time window, and $t_{win} = t_e - t_s$ is the properly chosen time window.

When (4.34) and (4.35) are true, the mean of the unbiased and noise-free accelerometer measurements yield:

$$f^b_{IMU,x,mean} = \frac{1}{t_w} \int\limits_{t=t_s}^{t_s+t_w} f^b_x(t)dt = 0 \tag{4.36}$$

$$f^b_{IMU,y,mean} = \frac{1}{t_w} \int\limits_{t=t_s}^{t_s+t_w} f^b_y(t)dt = 0 \tag{4.37}$$

$$f^b_{IMU,z,mean} = \frac{1}{t_w} \int\limits_{t=t_s}^{t_s+t_w} f^b_z(t)dt = -g_m \tag{4.38}$$

where $\mathbf{f}^b_{IMU} = \begin{bmatrix} f^b_{IMU,x} & f^b_{IMU,y} & f^b_{IMU,z} \end{bmatrix}^\top$, and $g_m \leq g = ||\mathbf{g}^b||$ is a value dependent on the characteristics of the waves.

In the biased case, $\mathbf{f}^b_{IMU} = \mathbf{a}^b - \mathbf{g}^b + \mathbf{b}^b$, where $\mathbf{b}^b = \begin{bmatrix} b^b_x & b^b_y & b^b_z \end{bmatrix}^\top$, the mean accelerometer output should be:

$$f^b_{IMU,x,mean} = \frac{1}{t_w} \int\limits_{t=t_s}^{t_s+t_w} f^b_x(t)dt = b^b_x \tag{4.39}$$

$$f^b_{IMU,y,mean} = \frac{1}{t_w} \int\limits_{t=t_s}^{t_s+t_w} f^b_y(t)dt = b^b_y \tag{4.40}$$

$$f^b_{IMU,z,mean} = \frac{1}{t_w} \int\limits_{t=t_s}^{t_s+t_w} f^b_z(t)dt = -g_m + b^b_z \tag{4.41}$$

The biases, b^b_x and b^b_y, can therefore be estimated by filtering $f^b_{IMU,x}$ and $f^b_{IMU,y}$ using a low-pass filter with a cut-off frequency f_c . f_c should be chosen low enough such that the filtered output is unaffected by single waves, but gives out the true mean values over the chosen time window .

The bias b^b_z can be estimated by filtering $f^b_{IMU,z} + \hat{g}^b_z$, where $\hat{g}^b_z$ is available from the estimated gravity vector $\hat{\mathbf{g}}^b = \mathbf{R}^e_b(\hat{\mathbf{q}}^e_b)^\top \mathbf{g}^e(\hat{\mathbf{p}}^e) = \begin{bmatrix} \hat{g}^b_x & \hat{g}^b_y & \hat{g}^b_z \end{bmatrix}^\top$ in $\{b\}$.

The complete differential equation for the mean filtering estimation method, using a first order low-pass filter, can be stated as:

$$\dot{\hat{\mathbf{b}}}^b_a = \omega_c(\mathbf{f}^b_{IMU} + \hat{\mathbf{g}}^b_z - \hat{\mathbf{b}}^b_a) \tag{4.42}$$

where $\hat{\mathbf{b}}^b_a = \begin{bmatrix}\hat{b}^b_{a,x} & \hat{b}^b_{a,y} & \hat{b}^b_{a,z}\end{bmatrix}^\top$, $\hat{\mathbf{g}}^b_z = \begin{bmatrix}0 & 0 & \hat{g}^b_z\end{bmatrix}^\top$ and $\omega_c = 2\pi f_c$ is the cut-off frequency in radians per second.

4.3.5 The Complete Observer

The complete nonlinear observer, including the mean filter accelerometer bias estimation method, is presented as:

$$\begin{aligned}
\dot{\hat{\mathbf{p}}}^e &= \hat{\mathbf{v}}^e + \theta_{GNSS}\mathbf{K}_{pp}(\mathbf{p}^e_{GNSS} - \hat{\mathbf{p}}^e) && (4.43)\\
\dot{\hat{\mathbf{v}}}^e &= -2\mathbf{S}(\boldsymbol{\omega}^e_{ie})\hat{\mathbf{v}}^e + \hat{\mathbf{f}}^e + \mathbf{g}^e(\hat{\mathbf{p}}^e) \\
&\quad +\theta^2_{GNSS}\mathbf{K}_{vp}(\mathbf{p}^e_{GNSS} - \hat{\mathbf{p}}^e) && (4.44)\\
\dot{\boldsymbol{\xi}} &= -\mathbf{R}^e_b(\hat{\mathbf{q}}^e_b)\mathbf{S}(\hat{\boldsymbol{\sigma}})(\mathbf{f}^b_{IMU} - \hat{\mathbf{b}}^b_a) \\
&\quad +\theta^3_{GNSS}\mathbf{K}_{\xi p}(\mathbf{p}^e_{GNSS} - \hat{\mathbf{p}}^e) + \mathbf{R}^e_b(\hat{\mathbf{q}}^e_b)\dot{\hat{\mathbf{b}}}^b_a && (4.45)\\
\hat{\mathbf{f}}^e &= \mathbf{R}^e_b(\hat{\mathbf{q}}^e_b)(\mathbf{f}^b_{IMU} - \hat{\mathbf{b}}^b_a) + \boldsymbol{\xi} && (4.46)\\
\dot{\hat{\mathbf{q}}}^e_b &= \frac{1}{2}\hat{\mathbf{q}}^e_b \otimes (\bar{\boldsymbol{\omega}}^b_{ib,IMU} - \bar{\hat{\mathbf{b}}}^b_g + \bar{\hat{\boldsymbol{\sigma}}}) - \frac{1}{2}\bar{\boldsymbol{\omega}}^e_{ie} \otimes \hat{\mathbf{q}}^e_b && (4.47)\\
\dot{\hat{\mathbf{b}}}^b_g &= \mathbf{Proj}(\hat{\mathbf{b}}^b_g, -k_I\hat{\boldsymbol{\sigma}}) && (4.48)\\
\dot{\hat{\mathbf{b}}}^b_a &= \omega_c(\mathbf{f}^b_{IMU} + \hat{\mathbf{g}}^b_z - \hat{\mathbf{b}}^b_a) && (4.49)
\end{aligned}$$

where

$$\hat{\boldsymbol{\sigma}} = k_1\underline{\mathbf{f}^b} \times \mathbf{R}^e_b(\hat{\mathbf{q}}^e_b)^\top\underline{\hat{\mathbf{f}}^e} + k_2\underline{\mathbf{c}^b} \times \mathbf{R}^e_b(\hat{\mathbf{q}}^e_b)^\top\underline{\mathbf{c}^e} \tag{4.50}$$

and

$$\underline{\mathbf{c}^b} = \frac{\mathbf{c}^b}{\|\mathbf{c}^b\|},\ \underline{\mathbf{c}^e} = \frac{\mathbf{c}^e}{\|\mathbf{c}^e\|},\ \underline{\mathbf{f}^b} = \frac{\mathbf{f}^b_{IMU} - \hat{\mathbf{b}}^b_a}{max\{\|\mathbf{f}^b_{IMU} - \hat{\mathbf{b}}^b_a\|, \delta\}},\ \underline{\hat{\mathbf{f}}^e} = \frac{\hat{\mathbf{f}}^e}{max\{\|\hat{\mathbf{f}}^e\|, \delta\}} \tag{4.51}$$

Since the accelerometer bias estimation is included, the bias is subtracted from the measured specific force $\mathbf{f}^b_{IMU}$. In the case of the estimated bias making $\|\mathbf{f}^b_{IMU} - \hat{\mathbf{b}}^b_a\|$ small or close to zero, the parameter δ is added to ensure well-definition of $\underline{\mathbf{f}}^b$. The derivative of the accelerometer bias is added to complete the dynamics, [20].

Alternative Representation of Velocity and Attitude

It is convenient to represent the nonlinear observer in the ECEF frame due to the availible GNSS measurements in ECEF/LLA. Nonetheless, this may not be the best representation for a control system or viewing the estimates to an operator. Grip et al. [20] presents a way to convert the velocity to NED, and to achieve the attitude quaternion $\hat{\mathbf{q}}^n_b$ between BODY and NED. This method utilizes the estimated position in LLA, which can be found with the use of (2.35) from Section 2.1.3.

The method from [20] yields:

$$\hat{\mathbf{q}}^n_b = \hat{\mathbf{q}}^n_e \otimes \hat{\mathbf{q}}^e_b \tag{4.52}$$

where $\hat{\mathbf{q}}^n_e$ could be calculated using

$$\hat{\mathbf{q}}^n_e = \hat{\mathbf{q}}_\mu \otimes \hat{\mathbf{q}}_\varphi \tag{4.53}$$

where

$$\hat{\mathbf{q}}_\mu = \begin{bmatrix} \cos(\frac{\hat{\mu}+\pi/2}{2}) & 0 & \sin(\frac{\hat{\mu}+\pi/2}{2}) & 0 \end{bmatrix}^\top \tag{4.54}$$

$$\hat{\mathbf{q}}_\varphi = \begin{bmatrix} \cos(\frac{\hat{\varphi}}{2}) & 0 & 0 & -\sin(\frac{\hat{\varphi}}{2}) \end{bmatrix}^\top \tag{4.55}$$

The Euler angle vector $\mathbf{\Theta}_{nb}$ can then be calculated from $\hat{\mathbf{q}}^n_b$ using the function `q2euler` in MATLAB.

4.4 Observer Implementation

The notation in Table 4.1 is needed when implementing the discrete nonlinear observer with multiple sensor rates, and in the case of having an observer loop integrating measurement buffers. The integration loop within the observer loop will therefore have a different discrete time step than the observer loop. In this thesis the observer integration time step will be equal to the IMU measurement time step, $\bar{T}_{obs} = T_{imu}$, with the corresponding frequency $\bar{F}_{obs} = F_{imu}$.

Parameter	**Description**
$\bar{T}_{obs}\ [s]$	Observer integration time step
$\bar{F}_{obs} = \frac{1}{\bar{T}_{obs}}\ [Hz]$	Observer integration frequency
$T_{obs}\ [s]$	Observer iteration time step
$F_{obs} = \frac{1}{T_{obs}}\ [Hz]$	Observer loop frequency
$T_{gnss}\ [s]$	GNSS measurement time step
$F_{gnss} = \frac{1}{T_{gnss}}\ [Hz]$	GNSS measurement rate
$K_{gnss} = \frac{\bar{F}_{obs}}{F_{gnss}}$	GNSS integration scale factor
$T_{gyc}\ [s]$	Gyrocompass measurement time step
$F_{gyc} = \frac{1}{T_{gyc}}\ [Hz]$	Gyrocompass measurement rate
$K_{gyc} = \frac{\bar{F}_{obs}}{F_{gyc}}$	Gyrocompass integration scale factor
$T_{imu}\ [s]$	IMU measurement time step
$F_{imu} = \frac{1}{T_{imu}}\ [Hz]$	MU measurement rate

Table 4.1: Discrete observer parameters

4.4.1 Corrector-Predictor Representation

The corrector-predictor representation in the discrete time domain is useful when handling sensors with multiple measurement rates, and when the observer is in the dead-reckoning state, i.e. unaided by GNSS position measurements. This representation is mainly used when implementing Kalman filters (KFs), but can also be utilized for non-linear observers, [15].

The corrector-predictor representation for (4.43) utilizing using forward Euler integration is given as:

$$\hat{\mathbf{p}}^e[k] \overset{\text{corr}}{=} \overline{\hat{\mathbf{p}}}^e[k] + \bar{T}_{obs} K_{gnss} \theta_{gnss} \mathbf{K}_{pp} (\mathbf{p}^e_{gnss}[k] - \overline{\hat{\mathbf{p}}}^e[k]) \tag{4.56}$$

$$\overline{\hat{\mathbf{p}}}^e[k+1] \overset{\text{pred}}{=} \hat{\mathbf{p}}^e[k] + \bar{T}_{obs} \hat{\mathbf{v}}^e[k] \tag{4.57}$$

The second term in the corrector step occurs every time a new GNSS position measurement is availible, whereas the predictor step occurs one or multiple times every observer loop iteration step. In the case of $\bar{F}_{obs} = F_{imu} = F_{obs}$, the predictor step happens every observer iteration. In dead-reckoning, the second term in the corrector step is absent until the GNSS measurements return.

Inconsistent GNSS Rate

The corrector-predictor representation method given in [15] is given for a constant K_{gnss}. To make the observer able to handle variable measurement rates, a corrector-predictor presentation for inconsistent measurement rates is realized. This method assumes that the observer loop frequency F_{obs} and the IMU measurement rate F_{imu} are known and constant. This method also matches the GNSS measurement with the corresponding IMU measurement.

Algorithm 4.1 Corrector-predictor representation for inconsistent GNSS rates

```
%% START OBSERVER THREAD
% F_OBS_ = F_IMU
k_GNSS = 0;
n_GNSS = 0
% Observer Iteration Loop
while (true),
    % Observer Integration Loop
    for (k=1:ImuBufferSize),
        K_GNSS = 0;
        % Increase Counter
        k_GNSS = k_GNSS + 1;
        % New GNSS Measurement
        if (n_GPS = 1) && (idIMU = idGNSS),
            F_GNSS = 1/(k_GNSS*T_OBS_);
            K_GNSS = F_OBS_/F_GNSS;
            % Reset Counter and Flag
            k_GNSS = 0;
            n_GNSS = 0;
        end
        % CORRECTOR
        p_e = p_e_ + T_OBS*K_GNSS*...
            THETA_GNSS*K_pp*...
            (p_e_GNSS-p_e_);
        % PREDICTOR
        p_e_ = p_e + T_OBS*v_e;
    end
end
%% END OBSERVER THREAD

%% START GNSS THREAD
while (true),
    if (NewGNSSMeasurement),
        idGNSS = idIMU;
        n_GNSS = 1;
        p_e_GNSS = NewMeasurement;
    end
end
%% END GNSS THREAD
```

4.4.2 The Dynamic GNSS Gain

The matrix $\mathbf{H}_K = \mathbf{A} - \mathbf{KC}$ must be Hurwitz to ensure stabilty. This means that the eigenvalues of $\mathbf{H}_K$ must be strictly negative, but lie within the unit circle, [12]. In addition, the infinity norm of $\mathbf{H}_K(s)$ must be sufficiently small [21], i.e. $\|\mathbf{H}_K(s)\|_\infty < \rho$.

For the discrete case, $\mathbf{H}_K$ is replaced with $\mathbf{H}_{K_d} = \mathbf{A} - \mathbf{K}_d\mathbf{C}$, where the discrete GNSS tuning matrix $\mathbf{K}_d$ is defined as $\mathbf{K}_d = \bar{T}_{obs} K_{gnss} \mathbf{K}$. Recall that $K_{gnss} = \frac{\bar{F}_{obs}}{F_{gnss}}$ from Table 4.1, which results in the following simplification $\mathbf{K}_d = \frac{1}{F_{gnss}}\mathbf{K}$. The stability matrix $\mathbf{H}_{K_d}$ is therefore highly dependent on the GNSS measurement rate F_{gnss}.

The Discrete Constant Gain

There exist different methods for choosing $\mathbf{K}$, e.g. pole placement, manual tuning and LMI-based methods, [20]. In this thesis an already chosen $\mathbf{K}$ will be used. The matrices contained in $\mathbf{K}$ will be held constant during simulations and experimental tests. From [20] they were found to be:

$$\mathbf{K}_{pp} = 0.6\mathbf{I}_{3\times3}, \ \mathbf{K}_{vp} = 0.11\mathbf{I}_{3\times3} \ \mathbf{K}_{\xi p} = 0.006\mathbf{I}_{3\times3} \tag{4.58}$$

Since the magnitude of the GNSS noise may vary, dependent on the satellite coverage, the need for an additional GNSS gain k_p is carried out. The k_p parameter is multiplied with each of the tuning matrices in (4.58):

$$\mathbf{K}_{pp} = k_p 0.6\mathbf{I}_{3\times3}, \ \mathbf{K}_{vp} = k_p 0.11\mathbf{I}_{3\times3} \ \mathbf{K}_{\xi p} = k_p 0.006\mathbf{I}_{3\times3} \tag{4.59}$$

The tuning matrix $\mathbf{K}_d$ gets redefined to $\mathbf{K}_d = \frac{k_p}{F_{gnss}}\mathbf{K}$, which says that for each different combination of k_p and F_{gnss}, there exist corresponding eigenvalues of $\mathbf{H}_{K_d(k_p,F_{gnss})}$ with an infinity norm $\|\mathbf{H}_{K_d(k_p,F_{gnss})}(s)\|_\infty = \rho(k_p, F_{gnss})$.

Different GNSS Sampling Rates

The complete set of eigenvalues of $\mathbf{H}_{K_d(k_p,F_{gnss})}$ for different combinations of k_p and F_{gnss} can be seen in Figure 4.2, more specifically called a root locus. To cover the domain of common GNSS frequencies, F_{gnss} is chosen from 0.25 $[Hz]$ to 10 $[Hz]$ with 0.25 $[Hz]$ increment. For each F_{gnss}, the tuning parameter k_p starts near zero with very small increments until the eigeinvalues go outside the unit circle.

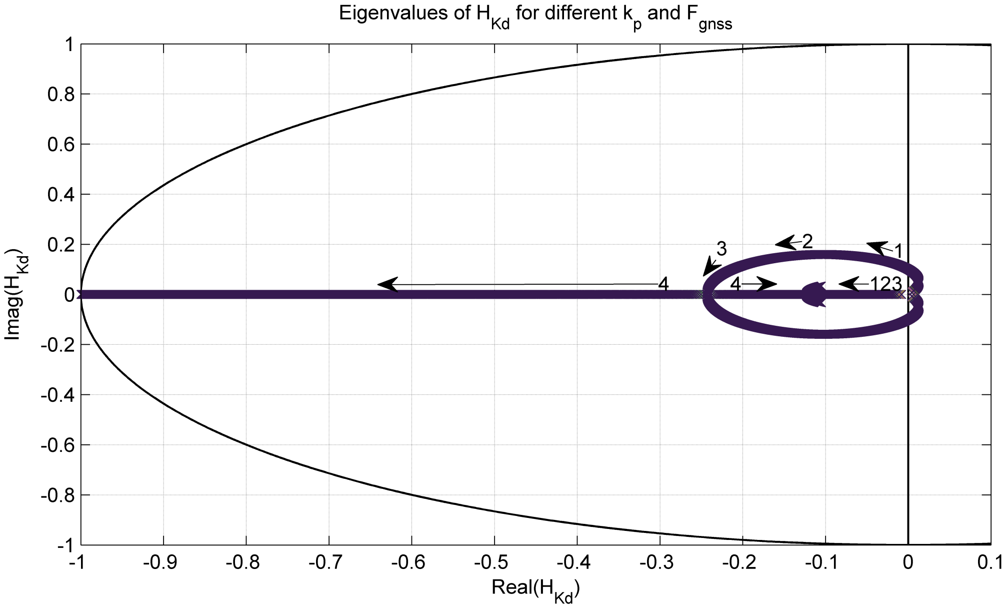


Figure 4.2: Eigenvalues of $\mathbf{H}_{K_d}$ for different k_p and F_{gnss}

The imaginary part of the eigenvalues followed the arrows denoted by 1, 2 and 3, whereas the corresponding real part follows the arrow denoted by 123. When the imaginery part gets close to the intersection between 3 and 4, the real values went rapidly to the left, see Figure 4.2.

Each unique F_{gnss} with the corresponding set of k_p-values give the same root locus, which make sense since the tuning matrix $\mathbf{K}$ is held constant. The difference lays in the set of k_p-values making $\mathbf{H}_{K_d(k_p,F_{gnss})}$ Hurwitz, where each F_{gnss} has a unique lower and upper boundary denoted by $k_{p,low}$ and $k_{p,high}$, respectively. See Figure 4.3 for illustration.

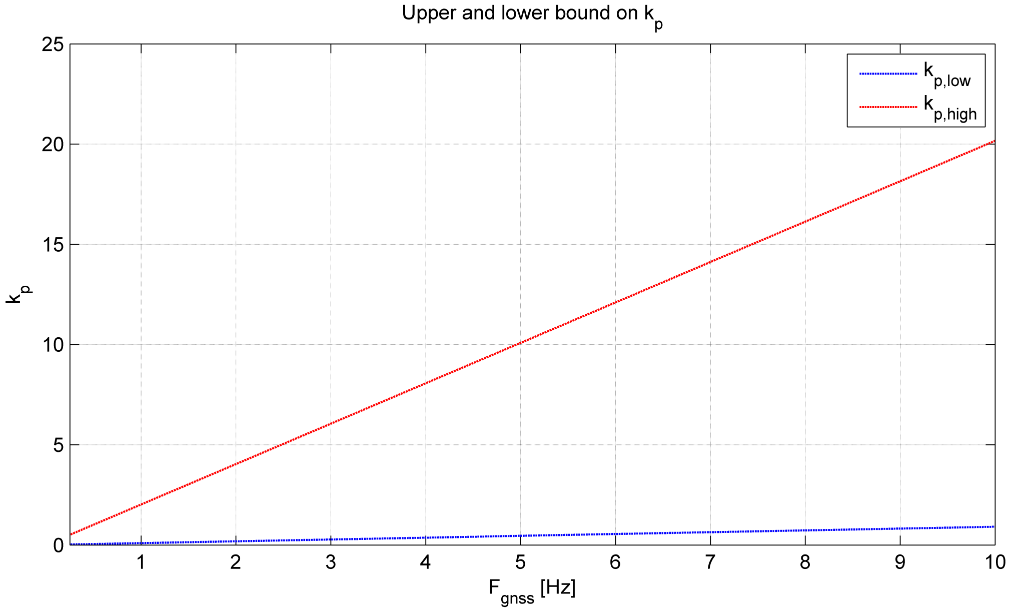


Figure 4.3: Upper and lower bound on k_p for each unique F_{gnss}

The corresponding $\|\mathbf{H}_{K_d(k_p,F_{gnss})}(s)\|_\infty$-values can be seen in Figure 4.4, where each line represents an unique frequency F_{gnss}, starting from the left at 0.25 $[Hz]$ and ends to the right at 10 $[Hz]$, with 0.25 $[Hz]$ increment.

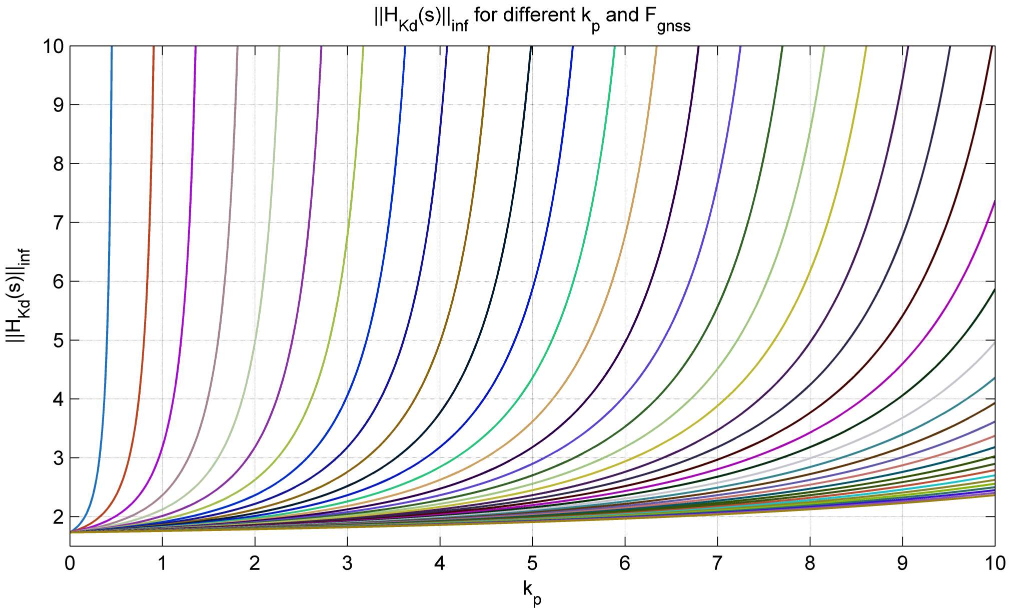


Figure 4.4: Infinity norm of $\mathbf{H}_{K_d}$ for different k_p and F_{gnss}

Preliminary Conclusion

The study of Figure 4.2, 4.3 and 4.4 result in the following preliminary conclusion:

- The set of k_p-values within the set $k_{p,low} \leq k_p \leq k_{p,high}$, which makes $\mathbf{H}_{K_d(k_p,F_{gnss})}$ Hurwitz, increases proportionally with the GNSS frequency F_{gnss}. This means that the system will be more robust in the presence of high GNSS frequencies, and more sensitive at low frequencies.
- Increasing k_p beyond a certain value, give a significant increase in the norm $\|\mathbf{H}_{K_d(k_p,F_{gnss})}(s)\|_\infty$. Since $\|\mathbf{H}_{K_d(k_p,F_{gnss})}(s)\|_\infty$ should be chosen sufficiently low [20], an upper bound $\rho_{high} \geq \|\mathbf{H}_{K_d(k_p,F_{gnss})}(s)\|_\infty$ should be defined to ensure stability.

The Dynamic GNSS Gain

When the value k_p makes the eigenvalues of $\mathbf{H}_{K_d(k_p,F_{gnss})}$ go beyond the intersection between 3 and 4 in Figure 4.2, a dramatic increase in $\|\mathbf{H}_{K_d(k_p,F_{gnss})}(s)\|_\infty$ is observed. From simulations in Chapter 6, any values of k_p beyond this intersection results in observer instability.

Two infinity norm values $\rho_{low} \leq \|\mathbf{H}_{K_d(k_p,F_{gnss})}(s)\|_\infty \leq \rho_{high}$, between 1 and the intersection in Figure 4.2, are chosen to define the region $k_{p,low} \leq k_p \leq k_{p,high}$ for different values of F_{gnss}.

The lower and upper bound on $\|\mathbf{H}_{K_d(k_p,F_{gnss})}(s)\|_\infty$ are defined as:

$$\rho_{low} = 1.7572, \ \rho_{high} = 2.3644 \tag{4.60}$$

with the corresponding eigenvalues:

$$eigval(\mathbf{H}_{K_d(\rho_{low})}) = \begin{bmatrix} -0.0021 + 0.1037i \\ -0.0021 - 0.1037i \\ -0.0021 + 0.1037i \\ -0.0021 - 0.1037i \\ -0.0021 + 0.1037i \\ -0.0021 - 0.1037i \\ -0.0557 + 0.0000i \\ -0.0557 + 0.0000i \\ -0.0557 + 0.0000i \end{bmatrix}, \ eigval(\mathbf{H}_{K_d(\rho_{high})}) = \begin{bmatrix} -0.2 \\ -0.1 \\ -0.3 \\ -0.3 \\ -0.3 \\ -0.2 \\ -0.2 \\ -0.1 \\ -0.1 \end{bmatrix} \tag{4.61}$$

Figure 4.5 shows the upper and lower bound $k_{p,low} \leq k_p \leq k_{p,high}$ for the chosen infinity norms of $\|\mathbf{H}_{K_d(k_p,F_{gnss})}(s)\|_\infty$, given by (4.60).

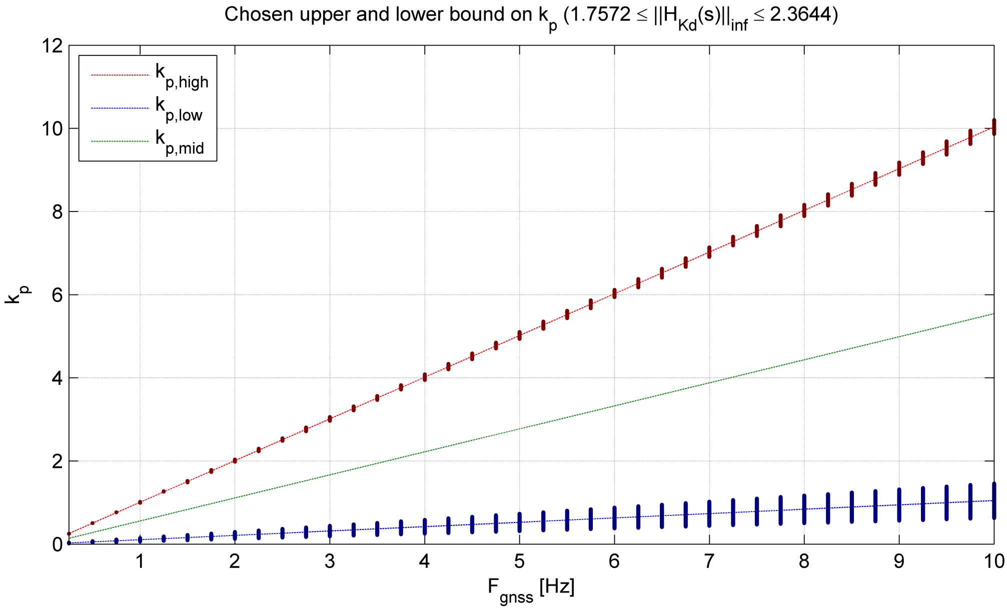


Figure 4.5: Chosen upper and lower bound on k_p

The red and blue dots in Figure 4.5 are found using:

$$||eigval(\mathbf{H}_{K_d(\rho_{low} \text{ or } \rho_{high})}) - eigval(\mathbf{H}_{K_d(k_p,F_{gnss})})|| \leq \epsilon_{eig} \tag{4.62}$$

where ϵ_{eig} defined the chosen error region and $eigval(\cdot)$ denotes a function which calculates the eigenvalues.

As the relationship between k_p and F_{gnss} was linear, the following functions were found using linear regression:

$$k_{p,low}(F_{gnss}) = 0.1F_{gnss} \tag{4.63}$$

$$k_{p,high}(F_{gnss}) = 1F_{gnss} \tag{4.64}$$

This results in the dynamic GNSS gain being defined as:

$$k_p(\chi, F_{gnss}) = \chi F_{gnss}, \; 0.1 \leq \chi \leq 1 \tag{4.65}$$

where χ is denoted as the GNSS weighting parameter, and should be chosen based on the quality of the GNSS measurements.

Using (4.65) with a constant χ will result in $\mathbf{H}_{K_d(k_p,F_{gnss})}$ always being Hurwitz, with $\|\mathbf{H}_{K_d(k_p,F_{gnss})}(s)\|_\infty \leq 2.3644$ and consistent eigenvalues for different GNSS frequencies. The dynamic GNSS gain $k_p(\chi, F_{gnss})$ would complement the corrector-predictor method for inconsistent GNSS rates, revised in 4.4.1, resulting in consistent error dynamics. Section 6 will include a case study to illustrate this.

4.4.3 The Discrete Observer

The position and velocity equations, (4.43) and (4.44), will be implemented using the combination of the forward and backward Euler discretization methods, which according to [15] is suited for a well-damped second-order system. The rest of the differential equations should be implemented using the forward Euler method. For more information about numerical integration the reader is referred to [15] and [37].

The final observer equations from 4.3.5 get the following discrete representation, using the corrector-predictor scheme for inconsistent GNSS rates:

Position

$$\begin{aligned}
\hat{\mathbf{p}}^e[k] &\overset{\text{corr}}{=} \bar{\hat{\mathbf{p}}}^e[k] + \frac{k_p(\chi, F_{gnss})}{F_{gnss}}\theta_{gnss}\mathbf{K}_{pp}(\mathbf{p}^e_{gnss}[k] - \bar{\hat{\mathbf{p}}}^e[k]) \\
\bar{\hat{\mathbf{p}}}^e[k+1] &\overset{\text{pred}}{=} \bar{\hat{\mathbf{p}}}^e[k] + \bar{T}_{obs}\bar{\hat{\mathbf{v}}}^e[k+1]
\end{aligned}$$

Velocity

$$\begin{aligned}
\hat{\mathbf{v}}^e[k] &\overset{\text{corr}}{=} \bar{\hat{\mathbf{v}}}^e[k] + \frac{k_p(\chi, F_{gnss})}{F_{gnss}}\theta^2_{gnss}\mathbf{K}_{vp}(\mathbf{p}^e_{gnss}[k] - \bar{\hat{\mathbf{p}}}^e[k]) \\
\bar{\hat{\mathbf{v}}}^e[k+1] &\overset{\text{pred}}{=} \hat{\mathbf{v}}^e[k] + \bar{T}_{obs}(-2\mathbf{S}(\boldsymbol{\omega}^e_{ie})\hat{\mathbf{v}}^e[k] + \hat{\mathbf{f}}^e[k] + \mathbf{g}^e(\bar{\hat{\mathbf{p}}}^e[k]))
\end{aligned}$$

Acceleration

$$\hat{\mathbf{f}}^e[k] = \mathbf{R}^e_b(\hat{\mathbf{q}}^e_b[k])(\mathbf{f}^b_{imu}[k] - \hat{\mathbf{b}}^b_a[k]) + \hat{\boldsymbol{\xi}}^e[k]$$

where

$$\begin{aligned}
\hat{\boldsymbol{\xi}}^e[k] &\overset{\text{corr}}{=} \overline{\hat{\boldsymbol{\xi}}}^e[k] + \frac{k_p(\chi, F_{gnss})}{F_{gnss}}\theta_{gnss}^3 \mathbf{K}_{\xi p}(\mathbf{p}_{gnss}^e[k] - \overline{\hat{\mathbf{p}}}^e[k]) \\
\overline{\hat{\boldsymbol{\xi}}}^e[k+1] &\overset{\text{pred}}{=} \hat{\boldsymbol{\xi}}^e[k] + \bar{T}_{obs}(-\mathbf{R}_b^e(\hat{\mathbf{q}}_b^e[k])\mathbf{S}(\hat{\boldsymbol{\sigma}}[k])(\mathbf{f}_{imu}^b[k] - \hat{\mathbf{b}}_a^b[k]) \\
&\quad +\omega_c \mathbf{R}_b^e(\hat{\mathbf{q}}_b^e[k])(\mathbf{f}_{imu}^b[k] + \hat{\mathbf{g}}_z^b[k] - \hat{\mathbf{b}}_a^b[k]))
\end{aligned}$$

Attitude

$$\begin{aligned}
\hat{\mathbf{q}}_b^e[k] &\overset{\text{corr}}{=} \overline{\hat{\mathbf{q}}}_b^e[k] \\
\overline{\hat{\mathbf{q}}}_b^e[k+1] &\overset{\text{pred}}{=} \hat{\mathbf{q}}_b^e[k] + \bar{T}_{obs}(\frac{1}{2}\hat{\mathbf{q}}_b^e[k] \otimes (\bar{\boldsymbol{\omega}}_{ib,imu}^b[k] - \bar{\hat{\mathbf{b}}}_g^b[k] + \bar{\hat{\boldsymbol{\sigma}}}[k]) \\
&\quad -\frac{1}{2}\bar{\boldsymbol{\omega}}_{ie}^e \otimes \hat{\mathbf{q}}_b^e[k]))
\end{aligned}$$

Gyroscope bias

$$\begin{aligned}
\hat{\mathbf{b}}_g^b[k] &\overset{\text{corr}}{=} \bar{\hat{\mathbf{b}}}_g^b[k] \\
\bar{\hat{\mathbf{b}}}_g^b[k+1] &\overset{\text{pred}}{=} \hat{\mathbf{b}}_g^b[k] + \bar{T}_{obs}(\mathbf{Proj}(\hat{\mathbf{b}}_g^b[k], -k_I\hat{\boldsymbol{\sigma}}[k]))
\end{aligned}$$

Accelerometer bias

$$\begin{aligned}
\hat{\mathbf{b}}_a^b[k] &\overset{\text{corr}}{=} \overline{\hat{\mathbf{b}}}_a^b[k] \\
\overline{\hat{\mathbf{b}}}_g^b[k+1] &\overset{\text{pred}}{=} \hat{\mathbf{b}}_a^b[k] + \bar{T}_{obs}\omega_c(\mathbf{f}_{imu}^b[k] - \hat{\mathbf{g}}_z^b[k] - \hat{\mathbf{b}}_a^b[k])
\end{aligned}$$

Injection term

$$\hat{\boldsymbol{\sigma}}[k] \quad = k_1\underline{\hat{\mathbf{f}}^b}[k] \times \mathbf{R}_b^e(\hat{\mathbf{q}}_b^e[k])^\top \underline{\hat{\mathbf{f}}^e}[k] + \quad k_2\underline{\mathbf{c}^b}[k] \times \mathbf{R}_b^e(\hat{\mathbf{q}}_b^e[k])^\top \underline{\mathbf{c}^e}[k] \tag{4.66}$$

where

$$\begin{aligned}
\underline{\mathbf{c}}^b[k] &= \frac{\mathbf{c}^b[k]}{||\mathbf{c}^b[k]||} \\
\underline{\mathbf{c}}^e[k] &= \frac{\mathbf{c}^e[k]}{||\mathbf{c}^e[k]||} \\
\underline{\mathbf{f}}^b[k] &= \frac{\mathbf{f}^b_{imu}[k] - \hat{\mathbf{b}}^b_a[k]}{max\{||\mathbf{f}^b_{imu}[k] - \hat{\mathbf{b}}^b_a[k]||, \delta\}} \\
\underline{\hat{\mathbf{f}}}^e[k] &= \frac{\hat{\mathbf{f}}^e[k]}{max\{||\hat{\mathbf{f}}^e[k]||, \delta\}}
\end{aligned}$$

In the discrete injection term, the scale factor K_{gyc} is added to compensate for the gyrocompass' lower sampling rate, compared to the IMU rate.

4.5 Observer Tuning

4.5.1 Sensor and System Parameters

The sensor and system parameters given in Table 4.2 should be found based on the knowledge of the specific system and sensors.

Description	Parameter	Condition
Upper bound of the specific force norm	M_f $[\frac{m}{s^2}]$	$M_f \geq ||\mathbf{f}^b||$
Lower bound of the specific force norm	m_f $[\frac{m}{s^2}]$	$m_f \leq ||\mathbf{f}^b||$
Ensure well-defined specific force	δ $[\frac{m}{s^2}]$	$0 < \delta < m_f$
Norm of the real gyroscope bias	M_{b_g} $[\frac{rad}{s}]$	$M_{b_g} = ||\mathbf{b}^b_g||$
Upper bound on the estimated gyroscope bias	$M_{\hat{b}_g}$ $[\frac{rad}{s}]$	$M_{\hat{b}_g} > M_{b_g}$

Table 4.2: Observer sensor parameters

The lower bounds $0 < \delta < m_f \leq ||\mathbf{f}^b||$ can be set using the fact that non-accelerated systems have $||\mathbf{f}^b|| = ||\mathbf{g}^e(\mathbf{p}^e)||$. The norm of the gyroscope bias $M_{\hat{b}_g}$ can be estimated using stationary measurements from the gyroscopes, in addtion to the sensor datasheet.

4.5.2 Tuning Parameters

After finding the appropriate sensor and system parameters defined in Table 4.2, the main tuning parameters for the observer are given in Table 4.3.

Description	Parameter	Condition
Parameter to guarantee stability	θ_{gnss}	$\theta_{gnss} \geq 1$
Dynamic or constant GNSS gain	$k_p(\chi, F_{gnss})$ or k_p	$\mathbf{A} - \mathbf{K}_d\mathbf{C}$ is Hurwitz
IMU parameter	k_1	$k_1 > 0$
Gyrocompass parameter	k_2	$k_2 > 0$
Gyroscope bias parameter	k_I	$k_I > 0$
Accelerometer bias cut-off frequency	$f_c(t)$	$f_{c,high} > f_{c,low} > 0$
Additional accelerometer bias parameters	T and β	$T > 0$, $1 > \beta > 0$

Table 4.3: Observer Tuning parameters

The observer tuning should be done in the following order, under the assumption that the system being estimated is stationary:

1. Tune the attitude and gyroscope bias observer in open-loop, i.e. the GNSS aiding should be turned off.
2. When acquiring good attitude estimates, turn on the GNSS aiding and find the appropriate GNSS gain.

Without doing any adjustments to the injection term under the first stage, this would result in poor estimates since the estimated specific force $\hat{\mathbf{f}}^e$ is dependent on GNSS measurements for convergence. This can be solved by replacing $\hat{\mathbf{f}}^e$ with the gravity $\mathbf{g}^e(\mathbf{p}^e)$ obtained from the gravity model. This should be valid since the system is assumed to be stationary at rest, i.e $\hat{\mathbf{f}}^e = -\mathbf{g}^e(\mathbf{p}^e)$. When the estimated attitude stops drifting, and the gyroscope measurements have a zero mean, the attitude observer is assumed to be correctly tuned.

Accelerometer Bias Gains

The accelerometer bias cut-off frequency $\omega_c = 2\pi f_c$ should be chosen low enough such that single waves don't affect the filtered mean. Choosing f_c to small, would make the filter's response too slow, which result in a long convergence time.

To solve this issue, an exponentially decaying $f_c(t)$ is chosen:

$$f_c(t) = f_{c,low} + (f_{c,high} - f_{c,low})\exp(\frac{-t}{\alpha}) \tag{4.67}$$

where α is the decay rate of $f_c(t)$, defined as:

$$\alpha = \frac{T}{\log(f_{c,high} - f_{c,low}) - \log(\beta f_{c,low})} \tag{4.68}$$

where $f_{c,low}$ is the lower cut-off frequency, s.t. filtered output doesn't react on single waves. The higher cut-off frequency $f_{c,high}$ shortens the initialization phase greatly. The decay rate α define when $f_c(t)$ reaches $(1+\beta)f_{c,low}$ at $t = T$. With this solution a quick initialization phase is achieved, defined by $f_{c,high}$, before it slowly converge to the mean values by decreasing the cut-off frequency over a time period T until it reaches $(1+\beta)f_{c,low}$.

Chapter 5

Experimental Setup

This chapter describes the setup utilized in the experimental tests, which is given in the next chapter. Marine Technologies LLC (MT) was the provider of the hardware equipment, as well as the base software for implementation of the nonlinear observer. The software included an empty 50 $[Hz]$ loop, where the observer equations was supposed to be implemented.

The first section revise the reference system used for comparison with the observer, whereas the second section gives a short overview on the GNSS receiver and the chosen IMU. The third section describes the observer implementation, the logging software, and the graphical user interface (GUI). Lastly, the final setup is given.

5.1 Reference System

The Teledyne TSS Orion INS was the availible reference system for comparison. The Orion provided accurate estimates of roll, pitch, heading, heave and position in LLA coordinates, see Table 5.1. The system had three solid-state accelerometers and three ring laser gyroscopes (RLGs) with high quality. More information about the Orion can be found in [6].

Figure 5.1: TSS Orion INS, Photo: [6]

The Orion had a relatively long initialization time period, i.e. approximately 30 $[min]$, before being fully operational with the accuarcy from Table 5.1. This was beacause of the fine alignment process using the inertial sensors to acquire precise heading estimates. The Orion hardware was provided with software which made possible for the operator to insert the location of the GNSS receiver, and to correct roll, pitch and heading misalignments.

Estimate	**Accuracy**
Roll and Pitch	0.01 $[deg]$
Heading	0.1 $[deg]$
Heave	0.05 $[m]$
Position drift	$< 9.26\ [km/hour]$

Table 5.1: TSS Orion INS accuracy, [6]

5.2 Sensors

This section shortly describes the sensors utilized in the experimental setup. A standalone gyrocompass was not availible during the tests. Instead, the heading from the Orion INS was considered as a valid substitute. The heading measurement rate from Orion was set to be 10 $[Hz]$, which according to MT was a reasonable choice.

5.2.1 GNSS

The GNSS receiver availible for testing was the Fugro SeaSTAR 9200-G2, which utilized both GPS and GLONASS satellite constellations. This receiver yielded position estimates in LLA coordinates at 1, 2, 5 or 10 $[Hz]$. Due to limits in the cable baudrate, the highest stable measurement rate achieved was 5 $[Hz]$.

Figure 5.2: Fugro SeaSTAR 9200-G2, Photo: [5]

5.2.2 IMU

The chosen inertial measurement unit was the ADIS16485, including a triaxial gyroscope and a triaxial accelerometer. The IMU was based on MEMS technology and had the ability to provide measurements at 2.4 $[kHz]$. The ADIS was able provide low-pass filtering and sensor bias correction, before transmitting the measurements. It was decided that the measured data from the sensors should be processed as little as possible before utilized by the nonlinear observer. This was realized by setting the proper registers in the device. The highest possible sampling rate for the ADIS16485 with the current setup was 500 $[Hz]$. The units for the gyroscopes and accelerometer were set to be $[deg/s]$ and $[mg]$, which was converted to $[rad/s]$ and $[m/s^2]$ in the observer, respectively. The specifications of the ADIS16485 can be found in the datasheet, [1].

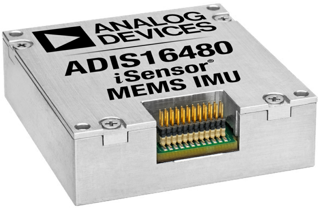


Figure 5.3: ADIS MEMS IMU, Photo: [7]

5.3 Implementation

The first section describes the observer implementation. Secondly, an overview of the logging system will be given. Lastly, the graphical user interface (GUI) used for supervision will be revised.

5.3.1 Nonlinear Observer

The discrete nonlinear observer was implemented using Microsoft Visual Studios 2008 on a control computer (CC[1]). This CC was provided by Marine Tehnologies LLC (MT), where the neccessary modules for communication with the GNSS, gyrocompass and IMU were included. The observer was implemented using the C++ programming language.

Inside the CC there was an observer loop running at 50 $[Hz]$. This loop included the discrete equations for the observer with an integration loop running at the IMU's measurement rate, i.e. 500 $[Hz]$.

Each sensor had its own thread inside the CC, which was triggered each time a new measurement was availible. The measurements were stored in a buffer in the case of a sensor thread being triggered more than once per observer loop iteration.

The IMU measurements were always stored in its respective buffer, with 10 measurements per observer iteration, since the IMU rate was 10 times faster than the observer loop. As the IMU had the fastest rate, an identification number (ID) was utilized for each of the 10 measurements in the IMU buffer, ranging from 1 to 10 in chronological order. Every time a new measurement arrived at the GNSS or gyrocompass thread, the arriving measurement was marked with the current IMU measurement ID. This made it possible to synchronize the measurements inside the observer integration loop, see Algorithm 4.1 in Chapter 4.

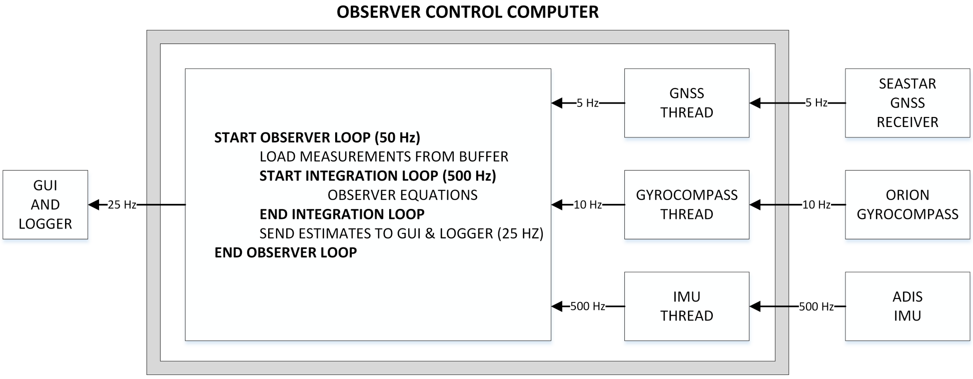


Figure 5.4: Nonlinear Observer Control Computer

To make the implementation intuitive and easy to troubleshoot, a matrix library was utilized, [3]. Even though calculation with matrices require more processing speed, no mentionable time delays were added. The matrix library should be removed in future applications to reduce computational overhead. Some modifications in the matrix library were done to improve the performance, such as a memory leak caused by an incorrect constructor.

[1] A CC runs on the Windows CE 7.0 Operating System

5.3.2 Logging System

It was important to log the raw data from all of the sensors, such that future improvements could be carried out with additional MATLAB simulations. The estimates from the nonlinear observer and the Orion INS were also logged, such that the two systems could be compared. MT made this possible by implementing proper logging software on a second control computer. The data was logged at different sampling rates, where the estimates from the nonlinear observer and the reference system were sampled at 25 $[Hz]$. The GNSS receiver, gyrocompass and IMU were logged at their respective frequencies, 5 $[Hz]$, 10 $[Hz]$ and 500 $[Hz]$.

During the first test runs in Egersund, it was discovered that the logs were degraded due to non-unique time samples in the datasets. The main idea was that the data should be written directly to file, when they arrived at the logging computer. But because of the high data rate, the received data was stored in buffers before being written to file. Each time the buffers were flushed, all of the data got the same time sample. This affected the IMU raw data logs the most. The magnitude of this issue was greatly reduced by further improvement of the logging software, but was never completely removed in the resulting data sets. With the use of post-processing methods offline, the time sample problem was further reduced.

5.3.3 Graphical User Interface

A graphical user interface (GUI) in C# was prepared by MT with the purpose of tuning observer parameters online. The GUI was modified to show the norm of different observer variables. These variables were the individual terms in the injection term, the injection term itself, the main observer error $\boldsymbol{\xi}$, and the position error. All of these variables should be small in the case of observer convergence and stability, and made it easy for the operator to see if everything worked properly.

In addition to the parameters and variables mentioned above, the GUI was further improved to plot the accelerometer bias, the gyroscope bias, the estimated position, and the GNSS measurements. This made the GUI a powerful tool, both in terms of real time tuning and supervision.

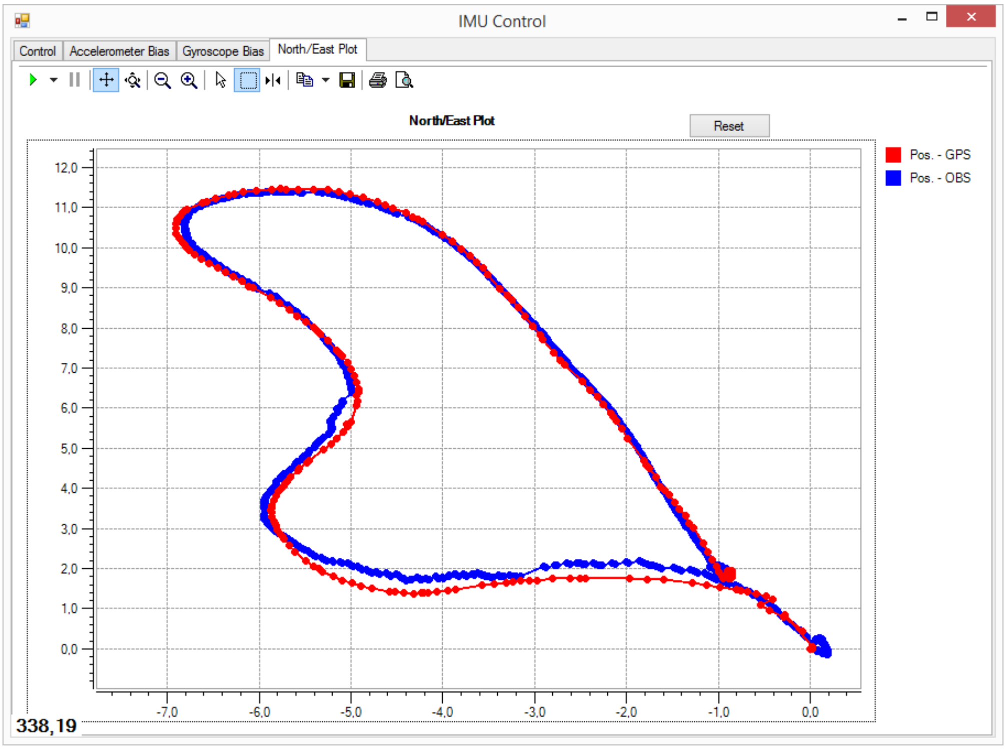


Figure 5.5: Graphical User Interface

5.4 Final Setup

The different sensors and systems communicated through an ethernet switch with power-over-ethernet (PoE). The PoE capability was necessary to power the IMU. The GNSS receiver and the TSS Orion INS used a serial interface to transfer measurements and estimates to the logging computer. The logging computer received additional data from the ethernet switch, such as estimates from the nonlinear observer. Furthermore, the heading from the Orion and the position measurement from the GNSS were converted for transmission over ethernet through a serial/ethernet interface. This made it possible for the nonlinear observer to receive the measurements. An analog to digital conversion was also utilized for the ADIS16485 IMU, where a Microsemi Smartfusion [4] board with an embedded FPGA[2] converted the data. The nonlinear observer received the measurements as illustrated in Figure 5.4. An extra CC (DPCC1) was included for additional plotting of data. A block diagram of the final setup can be seen Figure 5.6.

[2] Field-programmable gate array

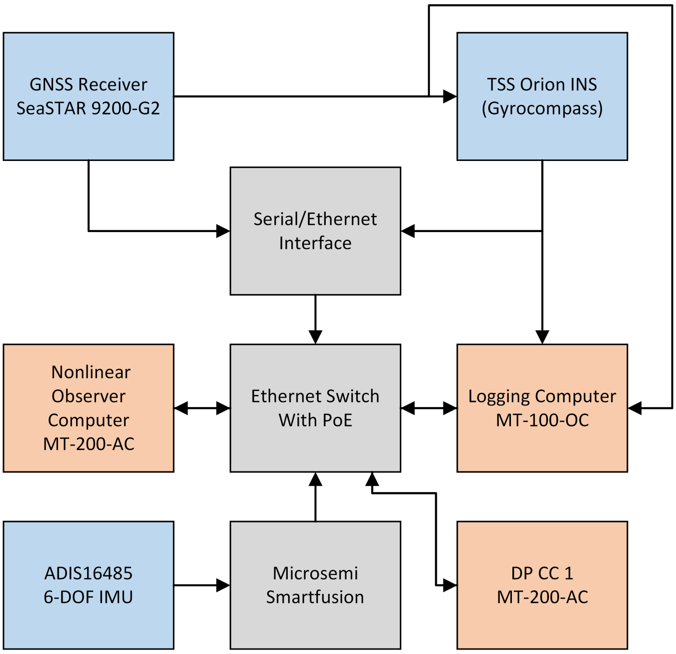

GNSS Receiver SeaSTAR 9200-G2
TSS Orion INS (Gyrocompass)
Serial/Ethernet Interface
Nonlinear Observer Computer MT-200-AC
Ethernet Switch With PoE
Logging Computer MT-100-OC
ADIS16485 6-DOF IMU
Microsemi Smartfusion
DP CC 1 MT-200-AC

Figure 5.6: Block Diagram

All of the devices were mounted on a plate and placed in a metal box, such that the equipment was shielded from the weather. The Orion INS and the IMU were placed such that their axis was aligned, where the distance between the sensors was reduced by placing the IMU on top of the Orion. The metal box was further placed in a wooden box and placed on a trolley, where the wooden box was made high, such that the GNSS antenna would get sufficient reception from the satellites. The GNSS antenna was placed directly above the reference system and the IMU, more specifically 1.3 $[m]$ above the sensors. This setup aligned the x- and y-axis of all sensors, such that $r^b_{imu} \approx r^b_{orion} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$ and $r^b_{gnss} = \begin{bmatrix} 0 & 0 & -1.3 \end{bmatrix}^\top$. See Figure 5.7 for the practical setup.

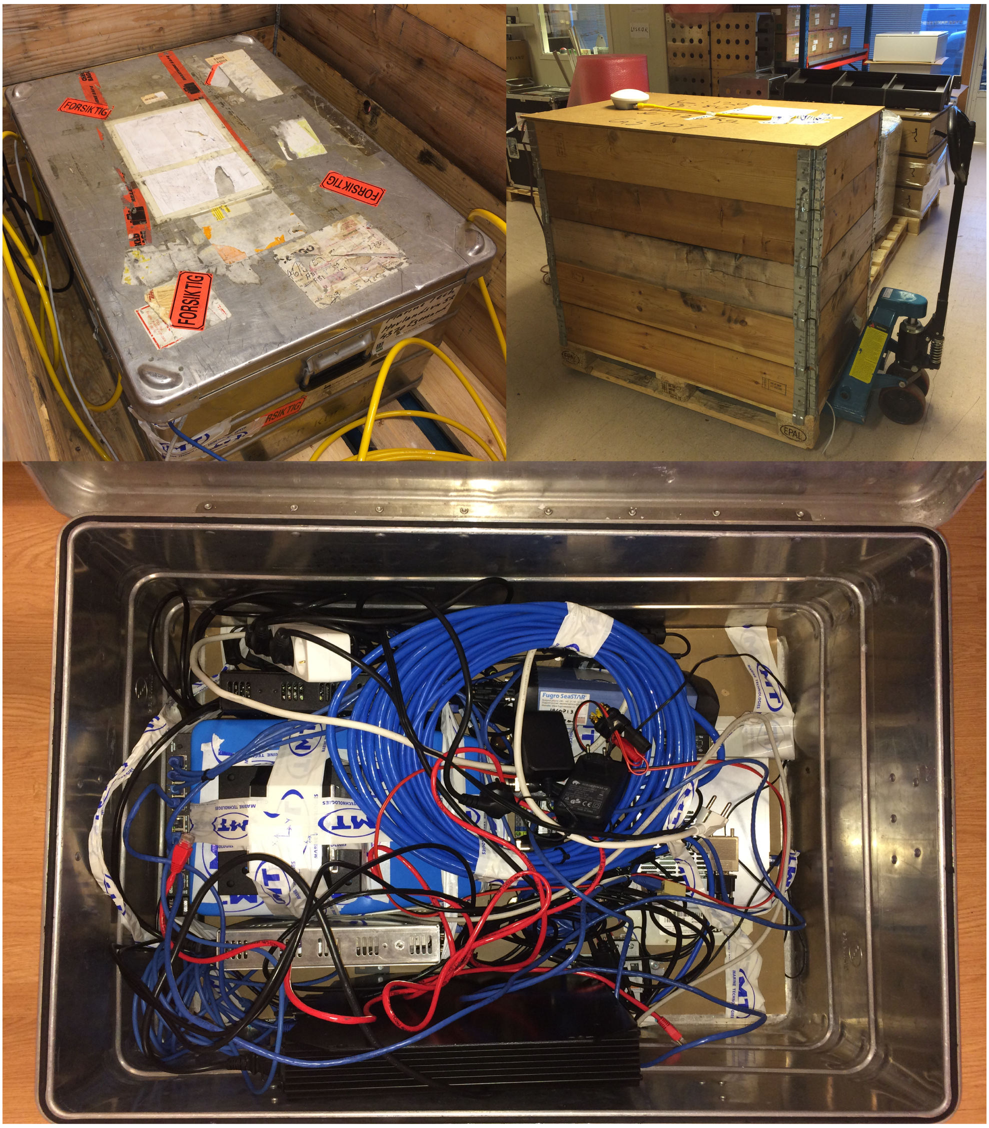


Figure 5.7: Practical Setup

Chapter 6

Simulation Results and Discussion

Simulations in MATLAB and Simulink were carried out to prepare the nonlinear observer for the experimental tests, revised in Chapter 7. The choice of vessel in the simulations was the semi-submersible rig with 6-DOF, which can be found in the mss toolbox[1] in MATLAB. The only environmental disturbance was the wave induced forces and moments, which was realized by a linear approximation of the JONSWAP spectrum.

The IMU measurements were added noise and bias, to make the simulations more realistic. In addition, the GNSS receiver was placed some distance above the IMU, where the latter was placed in CO. The sensor sampling rates and the observer loop frequency were set to be the same as in the experimental tests. The complete simulation setup and design can be found in Appendix A, whereas the main simulation parameters can be seen in Table 6.1. The observer sensor and system parameters are given in Table 6.2 and 6.3.

The nonlinear observer was tested in different cases. The first case compared the two accelerometer bias methods proposed in 4.3.4, and studied the effect of an incorrect estimated accelerometer bias. Secondly, the observer was tested with consistent and inconsistent GNSS rates, with and without the dynamic GNSS gain. The obsever was also tested in dead-reckoning, i.e. without GNSS aiding, to see how the observer performed in the presence of IMU errors. Lastly, an comparison between the nonlinear observer and the model-based Kalman filter was carried out. All of the cases were realized in open-loop[2].

This chapter is organized in the following way:

[1] The mss toolbox can be found at http://www.marinecontrol.org/

[2] Open-loop implies that the estimates of the observer are not utilized by the controller

- Section 6.1: Case 1: Accelerometer Bias Estimation
- Section 6.2: Case 2: Consistent GNSS Measurement Rate
- Section 6.3: Case 3: Inconsistent GNSS Measurement Rate
- Section 6.4: Case 4: Dead-Reckoning
- Section 6.5: Case 5: Nonlinear Observer versus Kalman Filter
- Section 6.6: Discussion

Remarks

The setup for all of the case studies were represented by the parameters in Table 6.1, 6.2 and 6.3, unless otherwise stated. To ensure a faster convergence of the observer, the attitude observer gains k_1, k_2 and k_I were multiplied by 20 during the initialization phase, which ended at $t = T_{init}$,[20].

The NED frame was chosen for intuitive interpretation of the position and velocity estimates, where the attitude was represented with Euler angles. This was achieved with the alternative representation given in Section 4.3.5. The position was also represented in LLA coordinates, where the initial LLA coordinate was subtracted from the date before being plotted. This was carried out to see the variation in latitude and longitude, due to very small increments for LLA coordinates in a small area.

Description	**Value**
Wave peak frequency (JONSWAP)	$\omega_0 = 1.2\ [\frac{rad}{s}]$
Significant wave height (JONSWAP)	$H_s = 1.0\ [m]$
Placement of the GNSS relative to CO	$\mathbf{r}^b_{gnss} = \begin{bmatrix} 0 & 0 & -30 \end{bmatrix}^\top\ [m]$
Placement of the IMU relative to CO	$\mathbf{r}^b_m = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top\ [m]$
Accelerometer noise (RMS)	$A_{acc,n} = 0.0121\ [\frac{m}{s^2}]$
Gyrosopenoise noise (RMS)	$A_{gyr,n} = 0.0026\ [\frac{rad}{s}]$
Accelerometer bias	$\mathbf{b}^b_a = 10^{-1} \begin{bmatrix} 2.30 & -3.10 & -4.15 \end{bmatrix}^\top\ [\frac{m}{s^2}]$
Gyrosope bias	$\mathbf{b}^b_g = 10^{-3} \begin{bmatrix} 4.00 & 2.50 & -3.00 \end{bmatrix}^\top\ [\frac{rad}{s}]$
GNSS frequency	$F_{gnss} = 5\ [Hz]$
Gyrocompass frequency	$F_{gyc} = 10\ [Hz]$
IMU frequency	$F_{imu} = 500\ [Hz]$
Observer frequency	$F_{obs} = 50\ [Hz]$

Table 6.1: Main Simulation Parameters

Description	Value
Upper bound of the specific force norm	$M_f = 9.81\ [\frac{m}{s^2}]$
Lower bound of the specific force norm	$m_f = 0.98M_f\ [\frac{m}{s^2}]$
Ensure well-defined specific force	$\delta = 0.98m_f\ [\frac{m}{s^2}]$
Norm of the real gyroscope bias	$M_{b_g} = 5.6 \cdot 10^{-3}\ [\frac{rad}{s}]$
Upper bound on the estimated gyroscope bias	$M_{\hat{b}_g} = 1.1M_{b_g}\ [\frac{rad}{s}]$

Table 6.2: Observer Sensor Parameters

Description	Value
Parameter to guarantee stability	$\theta_{gnss} = 2$
Dynamic GNSS gain	$k_p(0.5, F_{gnss}) = 0.5F_{gnss}$
IMU parameter	$k_1 = 1.5$
Gyrocompass parameter	$k_2 = 5.0$
Gyroscope bias parameter	$k_I = 0.005$
Accelerometer bias cut-off frequency (high)	$f_{c,high} = 5 \cdot 10^{-2}\ [Hz]$
Accelerometer bias cut-off frequency (low)	$f_{c,low} = 5 \cdot 10^{-6}\ [Hz]$,
Additional accelerometer bias parameters	$T = 1000\ [s],\ \beta = 0.1$
Observer initialization time	$T_{init} = 360\ [s]$

Table 6.3: Observer Tuning Parameters

6.1 Case 1: Accelerometer Bias Estimation

To see which accelerometer bias method was best suited for the remaining case studies, a comparison study between the mean filtering (MF) method and the vector bias (VB) method were carried out, see Section 4.3.4 for more information. The vessel was stationary during the whole simulation, with constant position and heading references. The VB method was implemented with the recursive least-squares method, with the parameters seen in Table 6.4.

Description	Value
Upper bound on the estimated accelerometer bias	$M_{\hat{w}} = 1.2\ [\frac{m}{s^2}]$
Initial gain matrix	$\boldsymbol{\Upsilon}(0) = 0.01\mathbf{I}_{4\times 4}$
Forgetting factor	$\alpha = 0$

Table 6.4: Parameters for the VB Method

Since the VB method utilized the estimated specific force $\hat{\mathbf{f}}^e$ as the bias free reference vector, the bias estimation started at $t = T_{init}$ $[s]$. This would give the estimate $\hat{\mathbf{f}}^e$ some time to converge, as recommendend in [20]. The same yielded for the MF method, since the estimated bias $\hat{b}^b_{a,z}$ utilized the estimated gravity $\hat{g}^b_z$ obtained from $\hat{\mathbf{g}}^b = \mathbf{R}^e_b(\hat{\mathbf{q}}^e_b)^\top \mathbf{g}^e(\hat{\mathbf{p}}^e)$. As this only affected the z-axis estimate, the bias estimation in the two other axes started at $t = 0$ $[s]$.

6.1.1 Results

Figure 6.1 and 6.2 illustrate that the MF method rapidly converged to the true values, whereas the VB method used a very long time, considering the x- and y-estimate. This was as expected, since the VB method highly relies on the PE condition to be fulfilled. The current sea-state realized by the JONSWAP spectrum seemed to be insufficient.

The z-axis estimate for the VB method was observed to converge faster than the MF method, but with a small constant error. This sudden transient was a result of the z-axis containing the gravity constant, which resulted in a higher energy density at low frequencies.

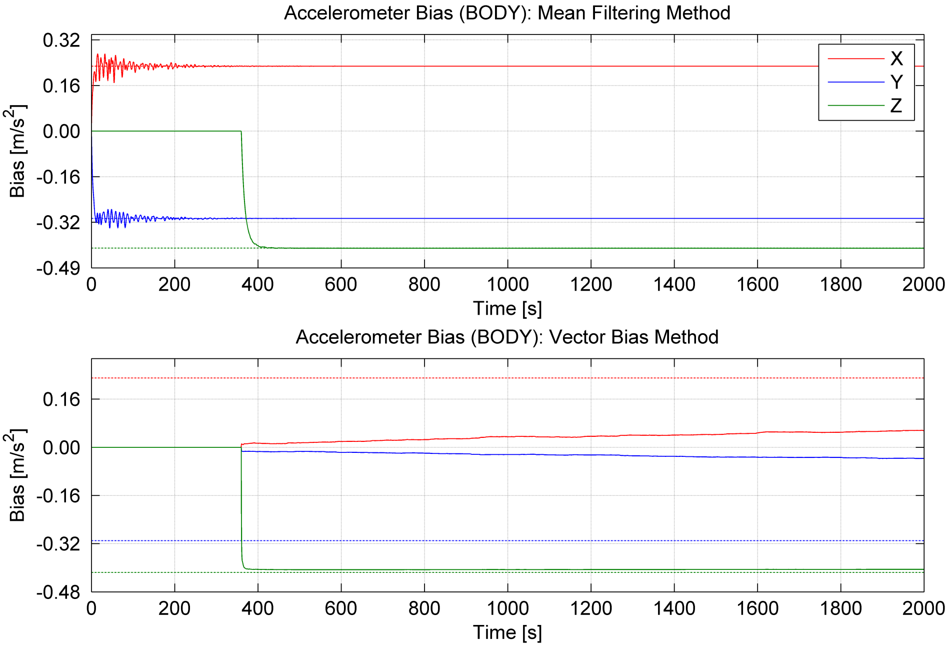


Figure 6.1: Case 1: Accelerometer Bias

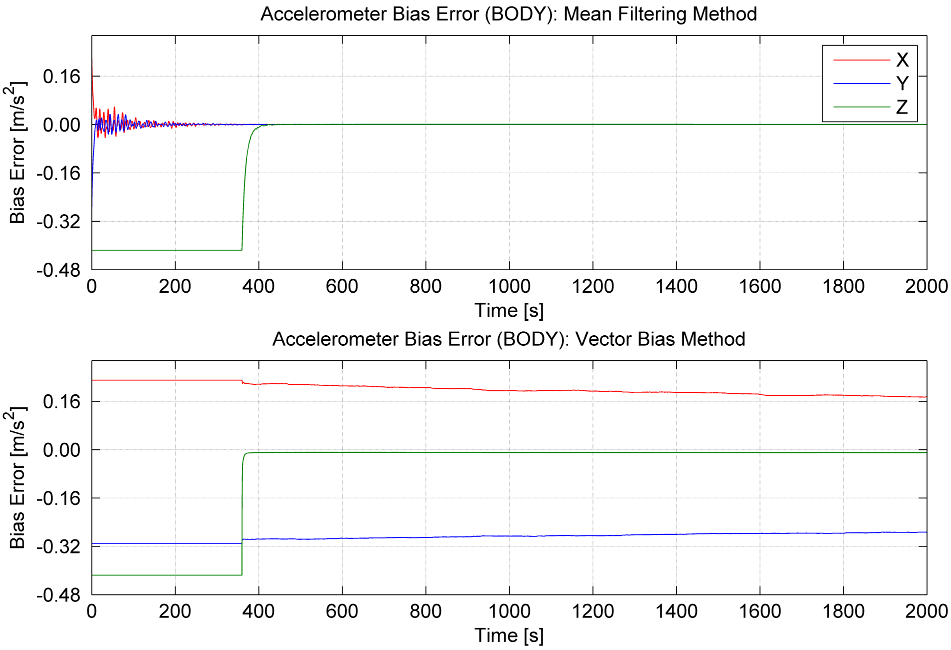


Figure 6.2: Case 1: Accelerometer Bias Error

By studying the main observer error $\boldsymbol{\xi}$ in Figure 6.3, it was easy to observe the effect of the bias from the accelerometers. Since the MF method started to estimate the bias in the x- and y-axis at $t = 0$ $[s]$, $\boldsymbol{\xi}$ experienced a lower value than the VB method. The most dominating change occured at $t = 360$ $[s]$, when both methods removed the bias in the z-axis, which resulted in $\boldsymbol{\xi} \approx \mathbf{0}_{3\times1}$.

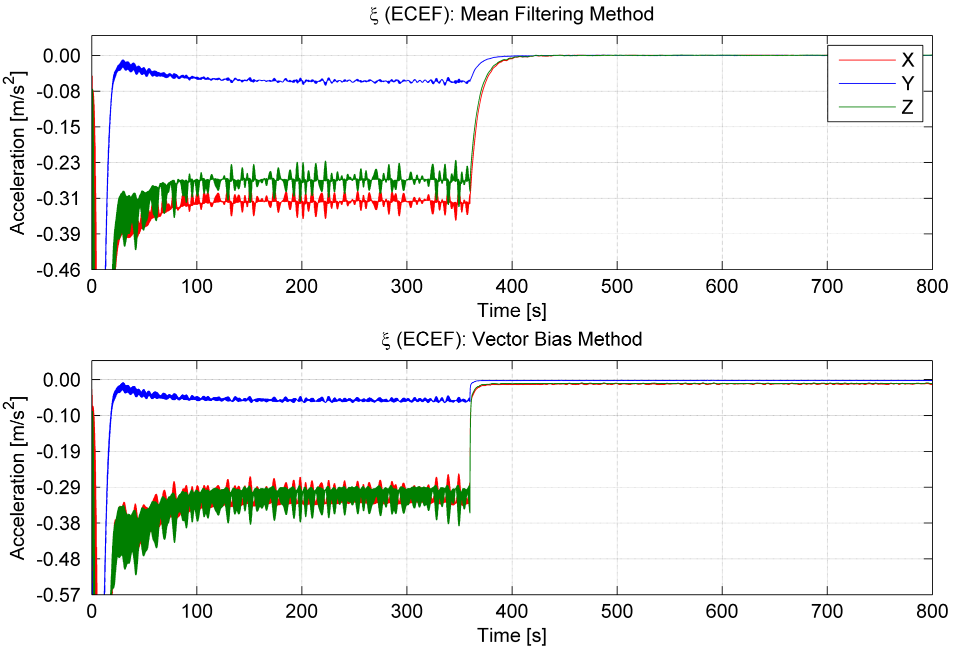


Figure 6.3: Case 1: Main Observer Error ($\boldsymbol{\xi}$)

From Figure 6.4 and 6.5, the estimated attitude with the VB method was observed to have a constant error proportional to the remaining accelerometer bias. This was due to the biased vector pair in the attitude observer's injection term, which concerned the specific force. The MF method had succesfully removed most of the bias, and resulted in an error close to zero. The deviations from zero were caused by the IMU sensor noise.

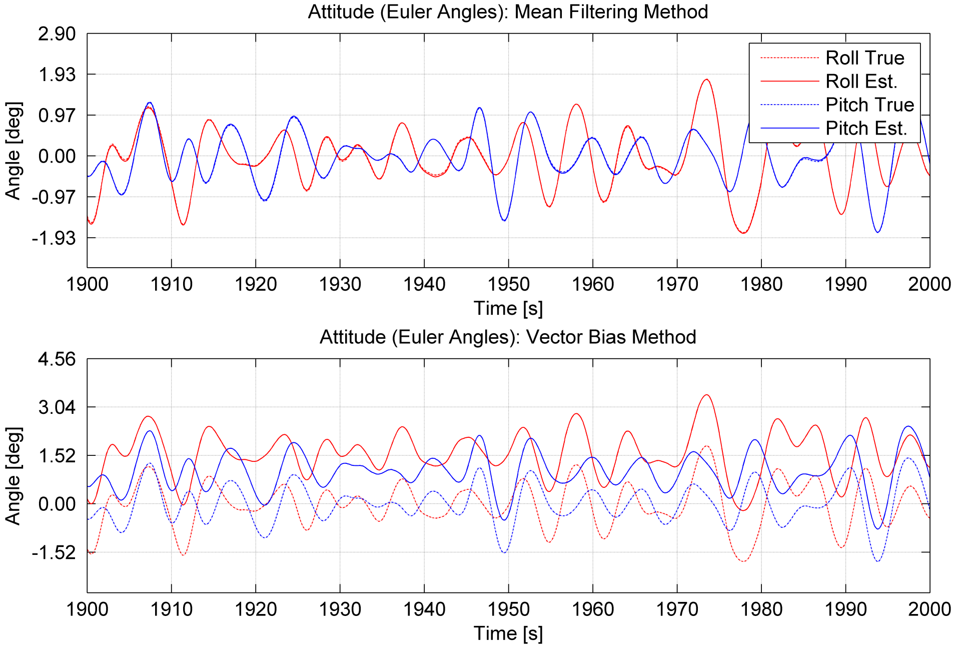


Figure 6.4: Case 1: Attitude (Euler angles)

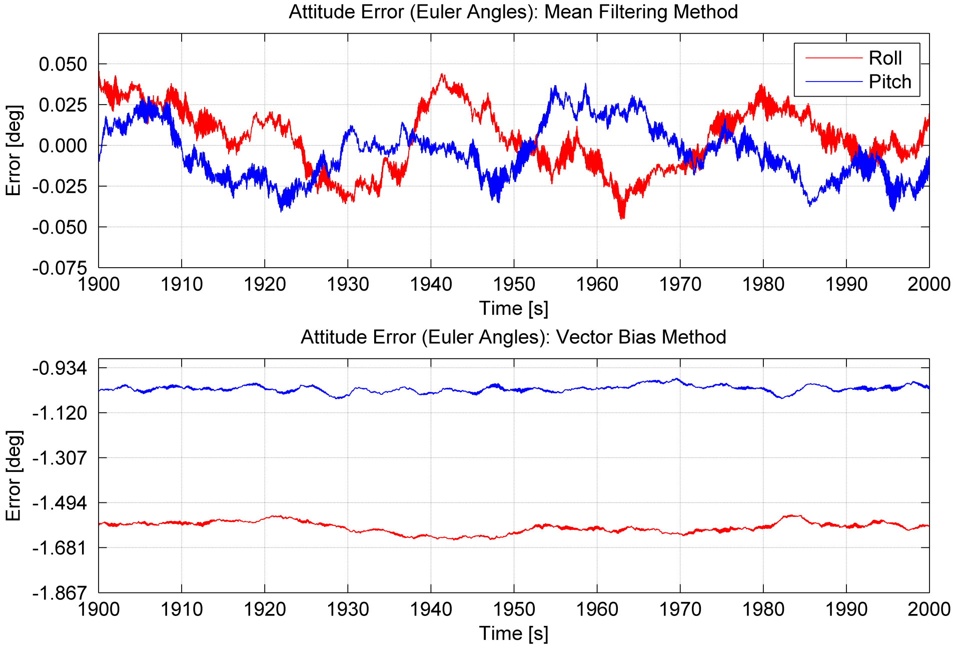


Figure 6.5: Case 1: Attitude Error (Euler angles)

Estimating the position, under the presence of a nearly constant attitude error, results in the GNSS being incorrectly transformed from $\{gnss\}$ to $\{m\}$. The VB method illustrates this in Figure 6.6, 6.7 and 6.8.

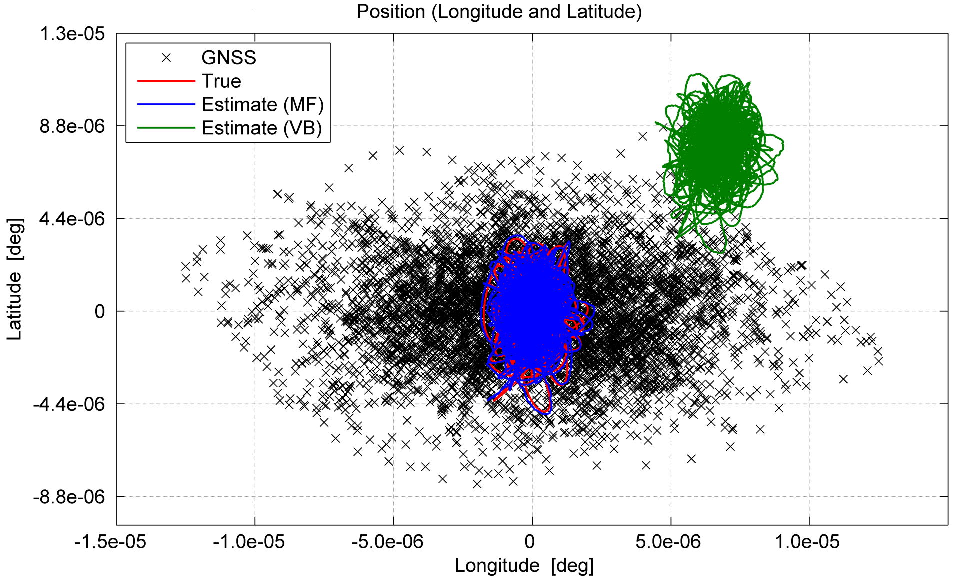


Figure 6.6: Case 1: Position (Latitude and Longitude)

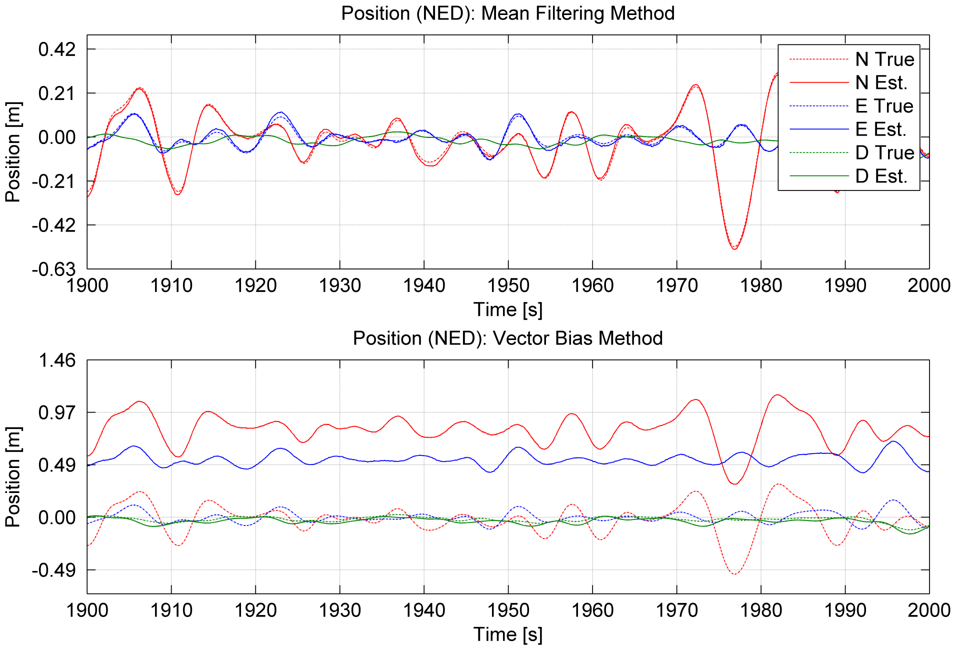


Figure 6.7: Case 1: Position (NED)

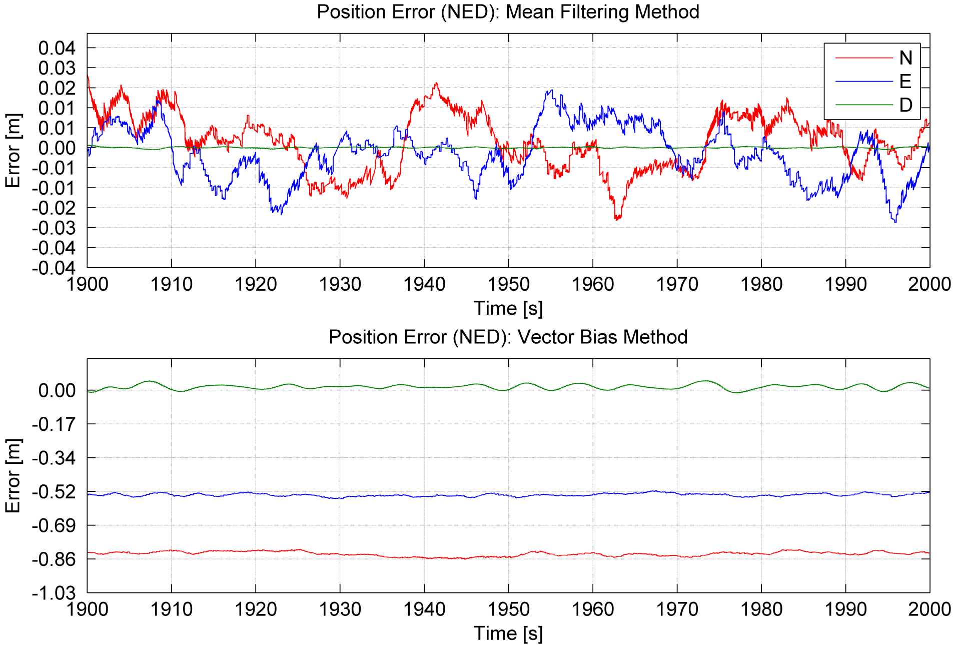


Figure 6.8: Case 1: Position Error (NED)

Figure 6.9 and 6.10 show the estimated velocities in the NED frame. The velocities were not affected by the accelerometer bias in the same manner as the attitude, due to the GNSS corrections.

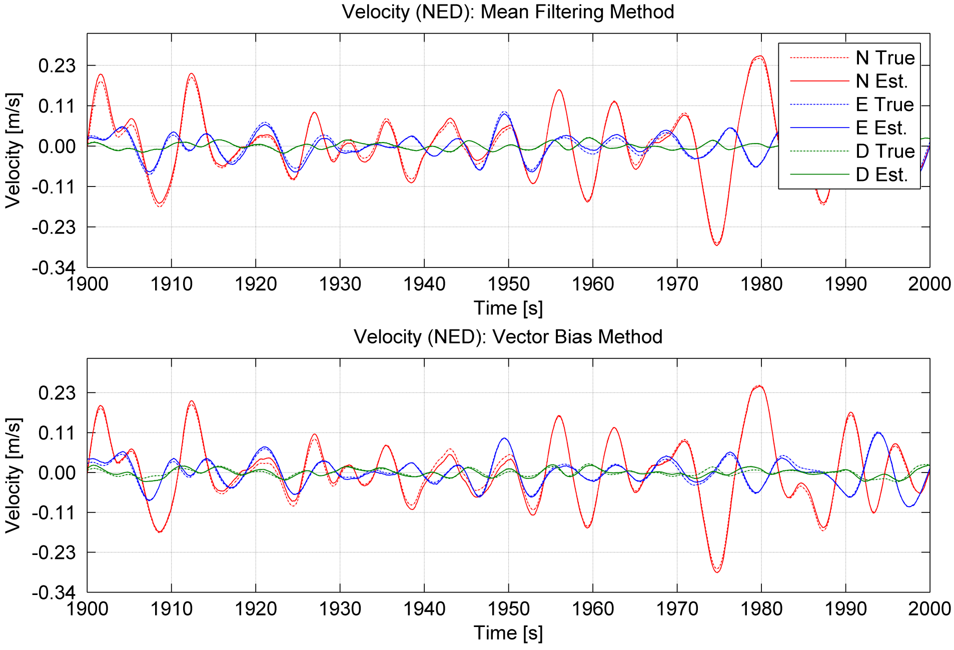


Figure 6.9: Case 1: Velocity

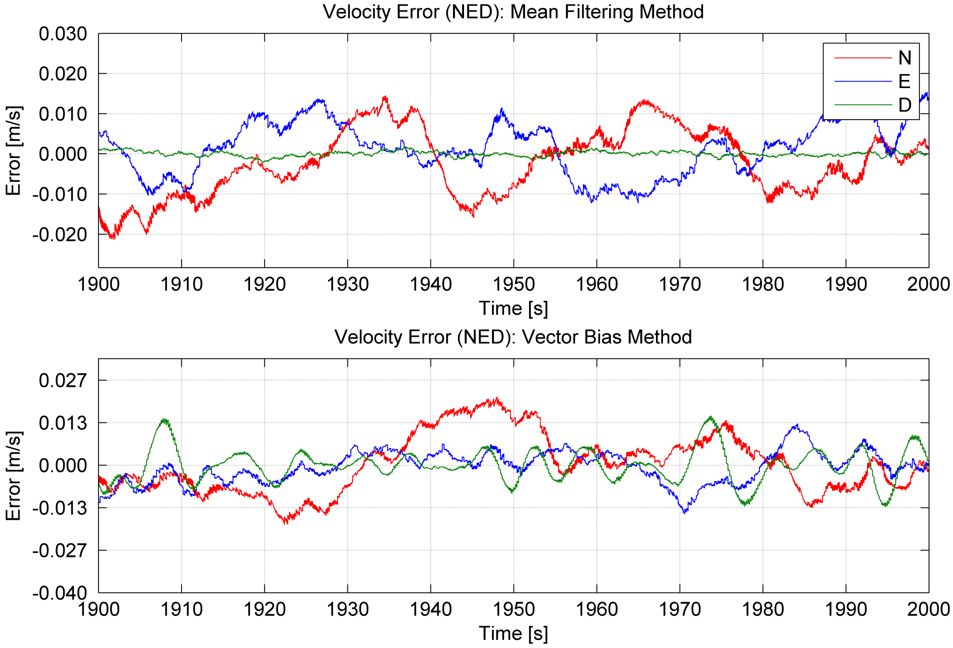


Figure 6.10: Case 1: Velocity Error

The estimated gyroscope bias was not affected by the accelerometer bias, which can be seen in Figure 6.11. Although the attitude observer utilized the biased vector pair in the injection term $\hat{\boldsymbol{\sigma}}$, it still provided information whether the vessel was rotating or not.

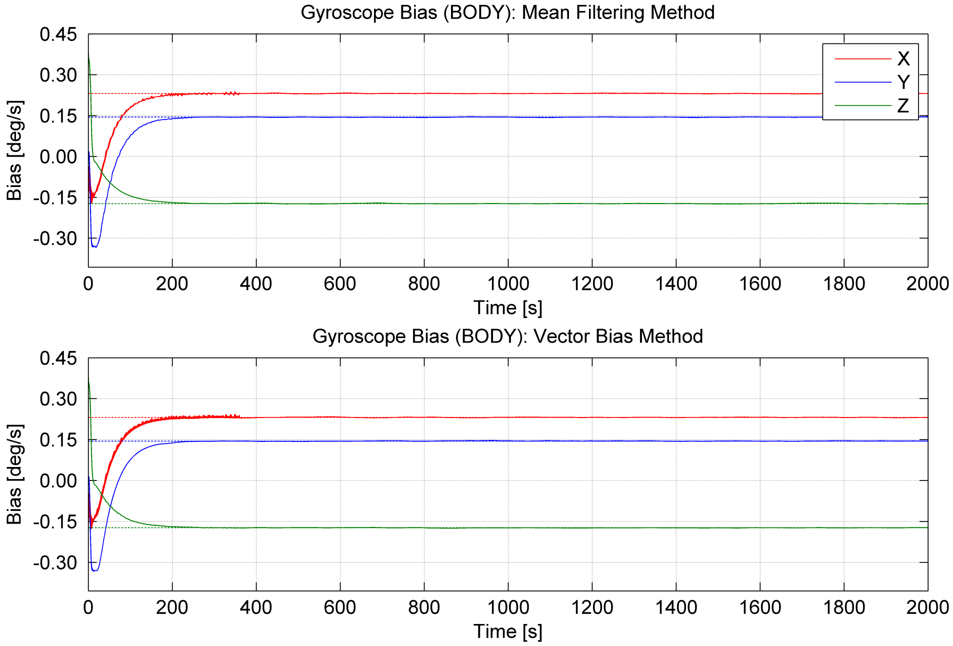


Figure 6.11: Case 1: Gyroscope Bias

6.2 Case 2: Consistent GNSS Measurement Rate

The reference values to the controller were set to be constant during the initialization phase. At $t = 1000$ $[s]$ the reference values changed and caused the vessel move in a diagonal fashion, approximately 20 $[m]$ south and 30 $[m]$ east. During the transition, the vessel rotated approximately 70 $[deg]$ about the z-axis. At $t = 1400$ $[s]$ the vessel stopped and held its position for the rest of the simulation. The main agenda with this case study was to test the performance of the observer during transit and change of heading. Since the mean filtering method provided the best estimates during the previous case study, this method was utilized in the remaining case studies.

6.2.1 Results

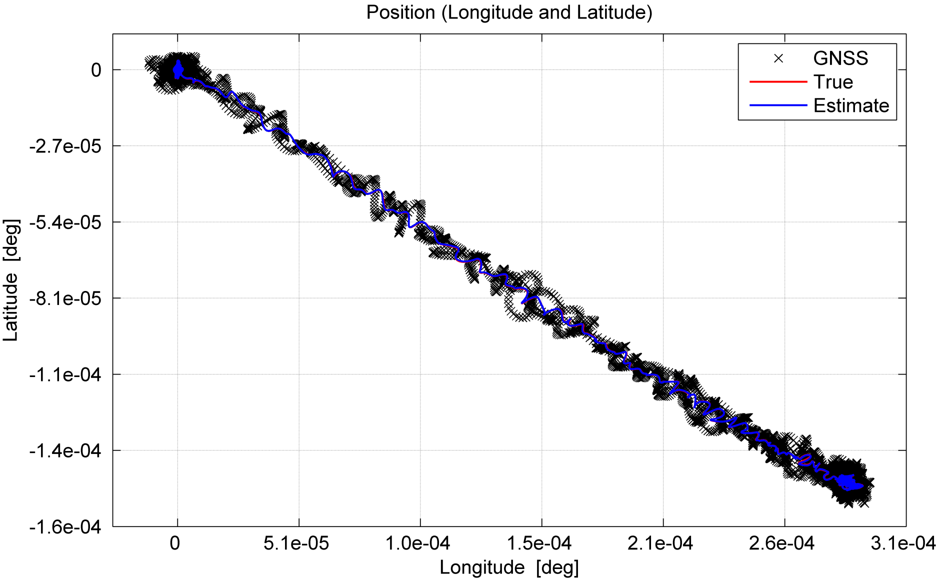


Figure 6.12: Case 2: Position (Latitude and Longitude)

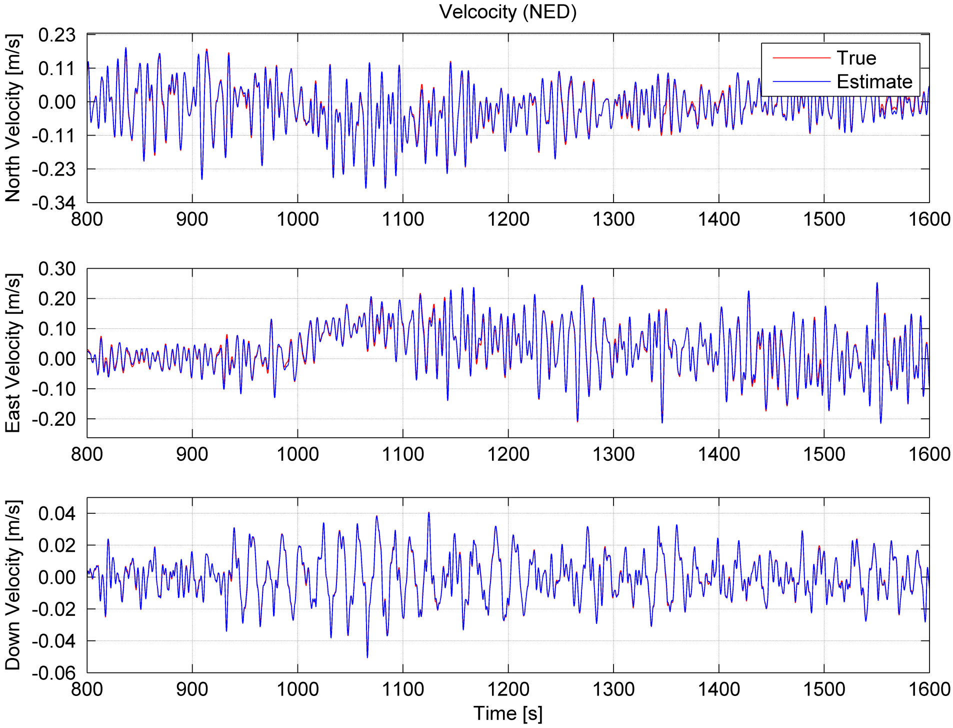


Figure 6.13: Case 2: Velocity

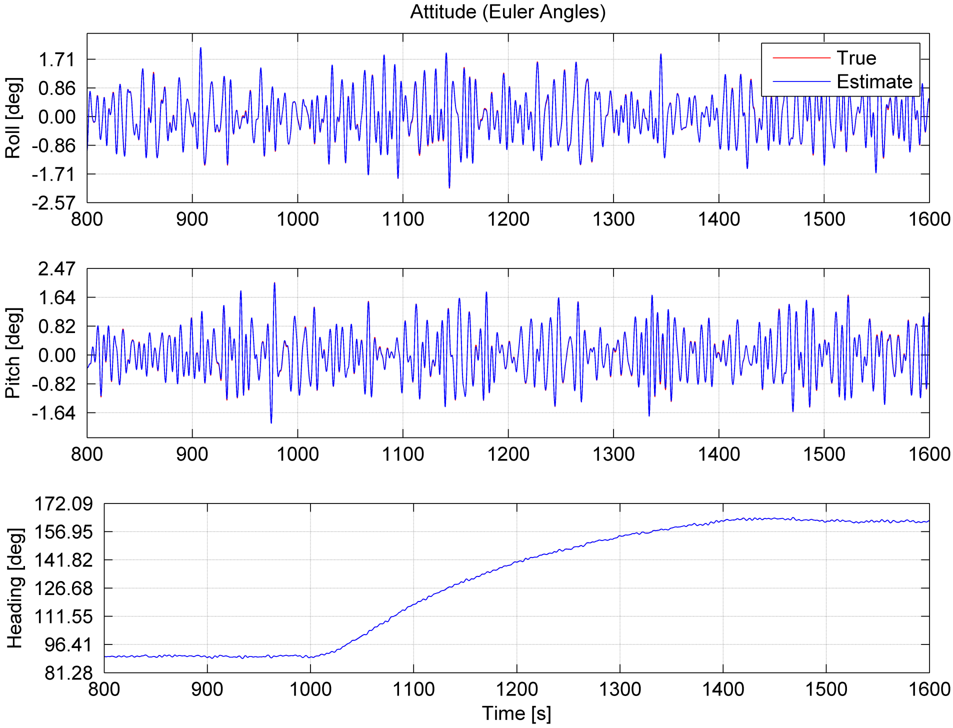


Figure 6.14: Case 2: Attitude

The position, velocity and attitude estimates can be seen in Figure 6.12, 6.13 and 6.14. The observer provided good estimates with relative small errors, i.e. position errors under 0.03 $[m]$, velocity errors under 0.02 $[m/s]$ and attitude errors under 0.025 $[deg]$, compared to the true values. The noisy errors were due to the inflicted measurement noise on the IMU. The more aggressive observer gains can be observed during the initialization phase. See Figure 6.15 for illustration.

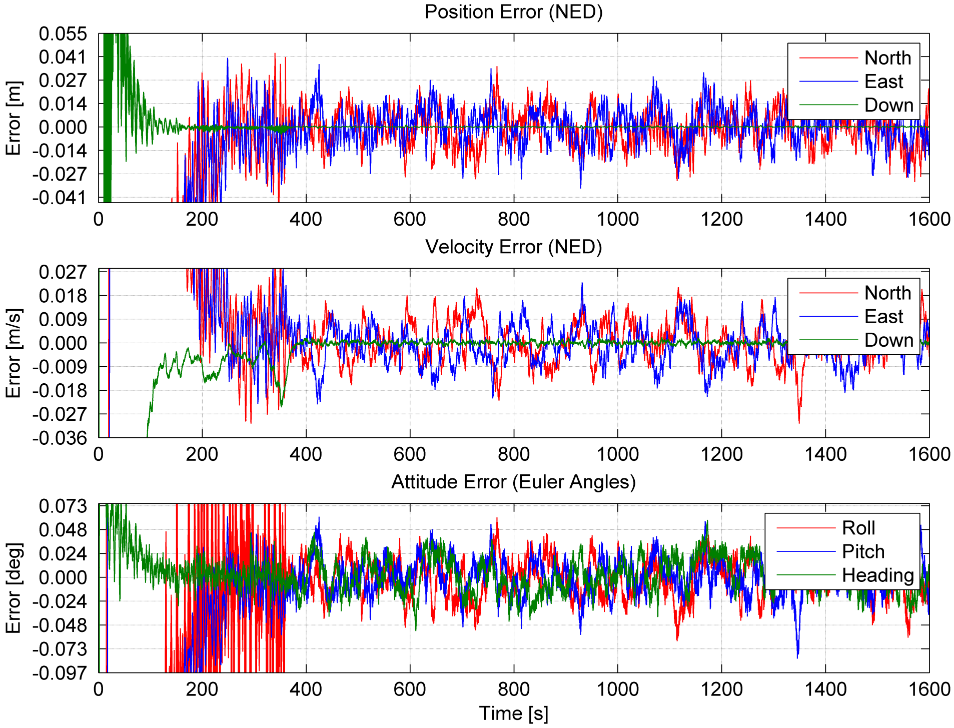


Figure 6.15: Case 2: Position, Velocity and Attitude Errors

Both the gyroscope and accelerometer biases were estimated with small errors, i.e. under $1 \cdot 10^{-3}$ $[deg/s]$ for the gyroscope biases and under $7 \cdot 10^{-4}$ $[m/s^2]$ for the accelerometer biases. It was difficult to remove these error residuals completely due to the aggressive IMU measurement noise, see Figure 6.16 and 6.17.

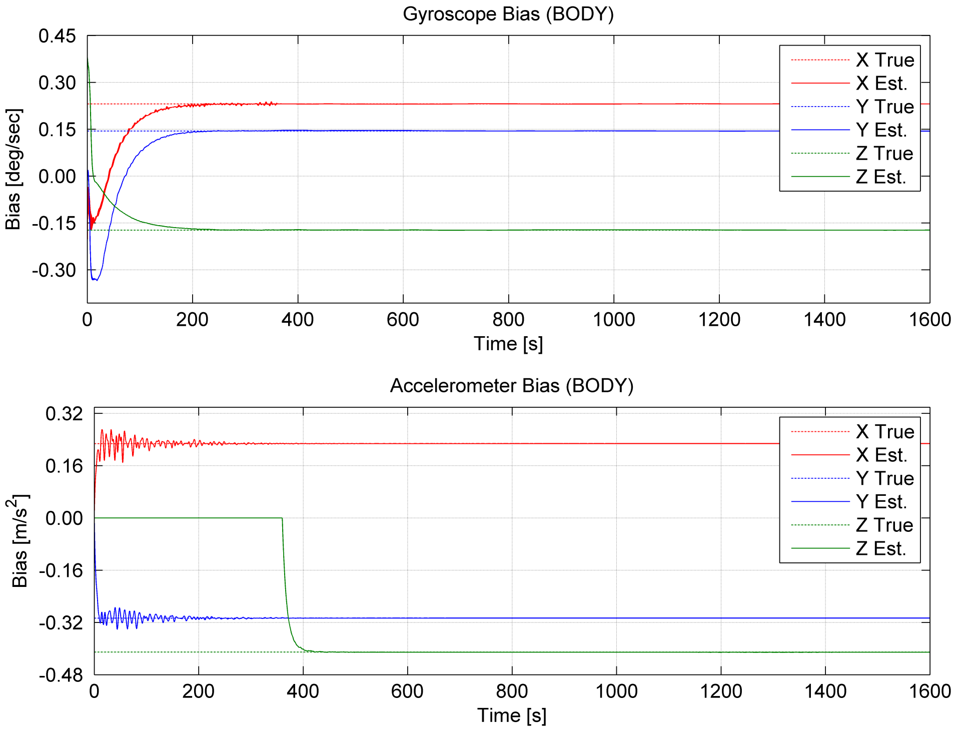


Figure 6.16: Case 2: Accelerometer and Gyroscope Bias

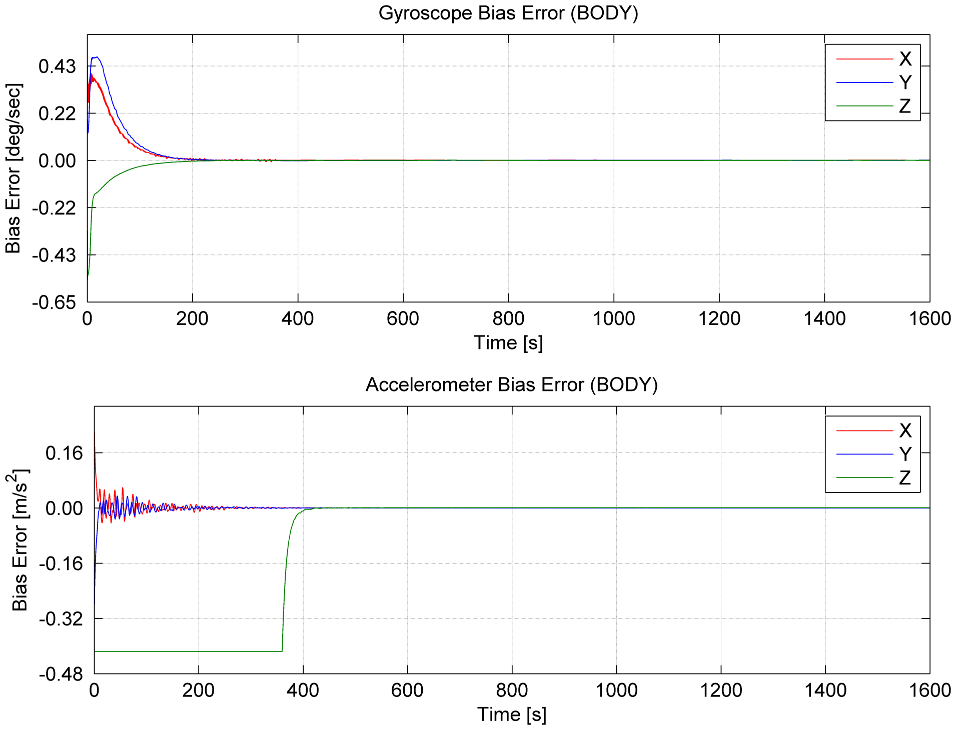


Figure 6.17: Case 2: Accelerometer and Gyroscope Bias Error

The most interesting result from this case study can be seen in 6.18, where the mean of the main observer error $\boldsymbol{\xi}$ was observed to slightly lie over the zero error line. This was most likely caused by small remaining errors in the estimated acceleration bias, which in a larger scale can be seen in Figure 6.3, in the previous case study. The black line in the figure represent the norm of $\boldsymbol{\xi}$, which is a useful variable when studying the performance of the observer.

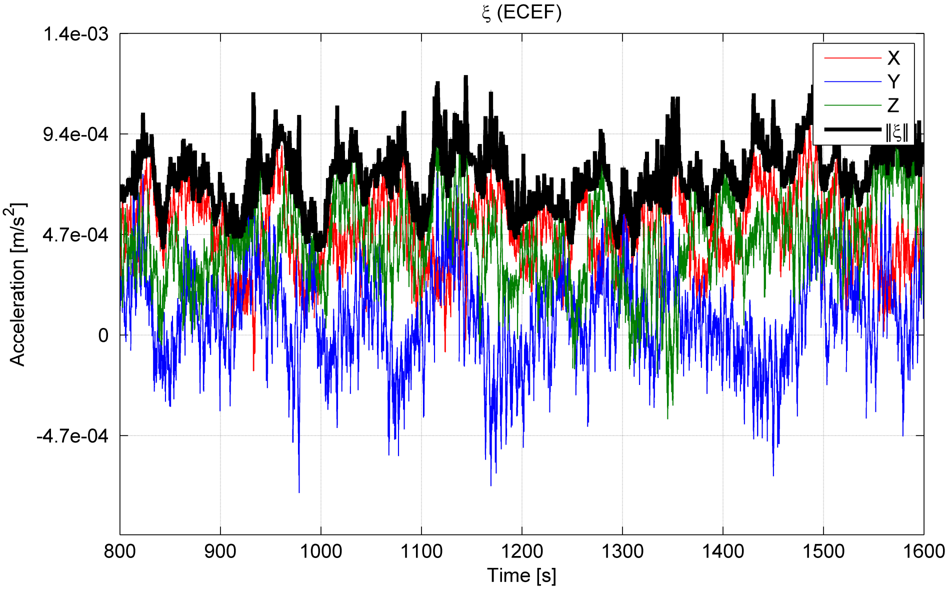


Figure 6.18: Case 2: Main Observer Error ($\boldsymbol{\xi}$)

6.3 Case 3: Inconsistent GNSS Measurement Rate

This case study will show that the dynamic GNSS gain $k_p(\chi, F_{gnss})$ makes the error dynamics consistent, under the influence of a varying GNSS rate. The GNSS frequency was chosen to vary aribtrary between 0.5, 1, 2, 5 and 10 $[Hz]$, holding each frequency for 40 $[s]$. The results from the dynamic gain will be compared to a constant GNSS gain k_p.

The dynamic GNSS gain was chosen as $k_p(1, F_{gnss}) = F_{gnss}$, whereas the constant weighting gain was chosen to be $k_p = 1$. This constant gain made the error dynamics stable for all frequencies. Choosing this value too big would make $\|\mathbf{H}_{K_d}(s)\|_\infty$ not sufficiently low, whereas a too low value would result in $\mathbf{H}_{K_d}$ not being Hurwitz. These conditions had to be fulfilled to achieve stable error dynamics.

The GNSS gains intersected at 1 $[Hz]$, which should result in the same behaviour for both systems. When the GNSS frequency went below 1 $[Hz]$, the constant GNSS gain system should provide estimates with lower accuracy, since the $\|\mathbf{H}_{K_d}(s)\|_\infty$ would be at a higher value. The vessel followed the same path as in the previous case study.

6.3.1 Results

Figure 6.19 shows that both methods provided good position estimates. This was as expected, since both methods provided stable error dynamics at the different GNSS frequencies.

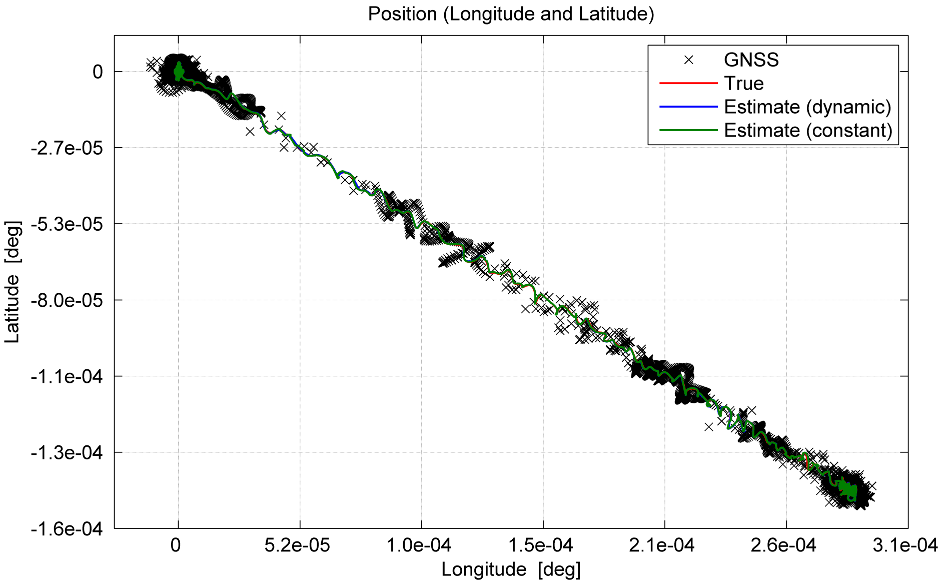


Figure 6.19: Case 3: Position

The comparison of the PVA errors in Figure 6.20 with the errors in Figure 6.21, revieled a slightly bigger error in the position estimates for the system with the constant gain. This was especially noticeable at the lowest frequency (0.5 $[Hz]$). The system with constant gain had a position error under 0.051 $[m]$, where the dynamic gain system had an error under 0.042 $[m]$. This was due to $\|\mathbf{H}_{K_d}(s)\|_\infty$ having a slightly higher value under 1 $[Hz]$ for the system with a constant gain. See Figure 6.22 and 6.23 for the different frequencies at the different time intervals.

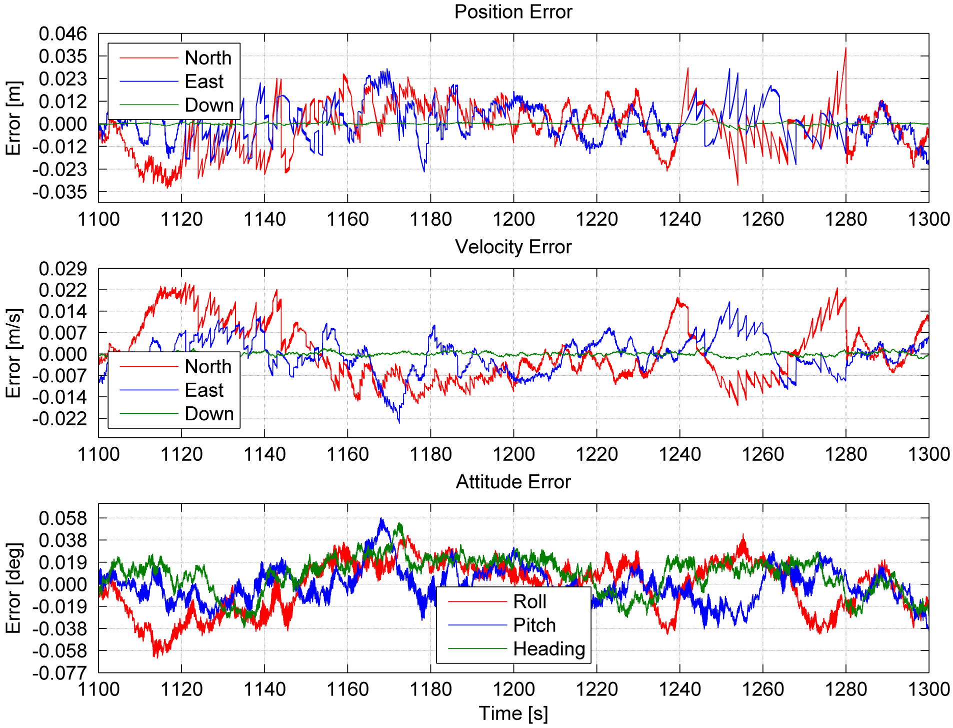


Figure 6.20: Case 3: PVA Errors (Dynamic GNSS Gain)

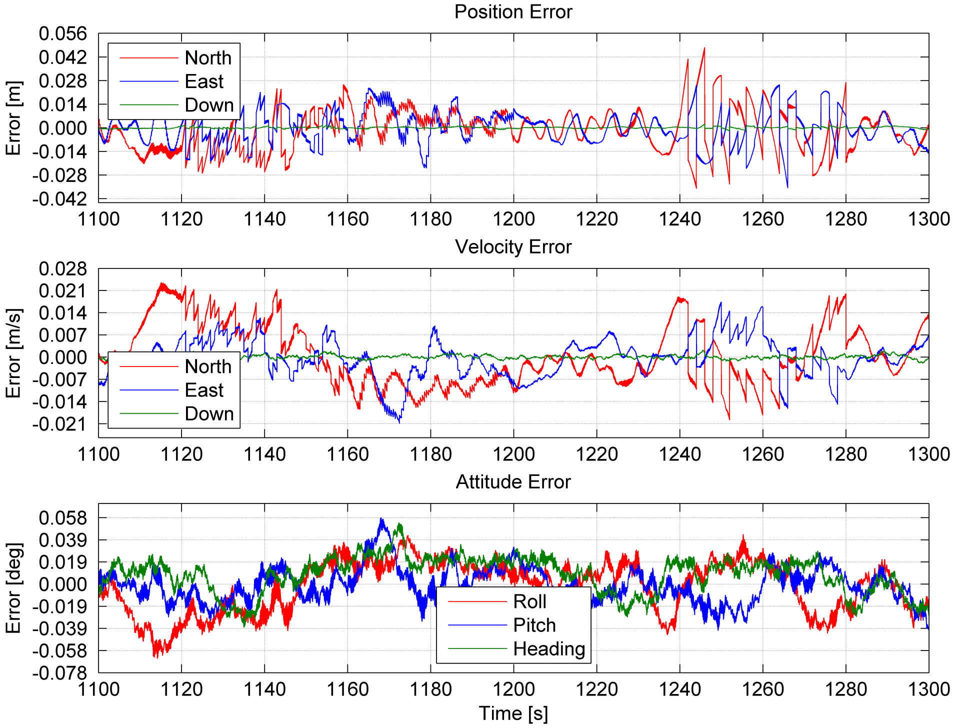


Figure 6.21: Case 3: PVA Errors (Constant GNSS Gain)

Figure 6.22 and 6.23 illustrate the main observer error $\boldsymbol{\xi}$ for both systems. The system with the dynamic GNSS gain seemed to have a more consistent behaviour, in terms of variance and aggresivity on each GNSS measurement. This coincided with the assertion of having consistent error dynamics.

The system with the constant gain had the opposite behaviour, where each frequency resulted in $\mathbf{H}_{K_d}$ having different poles, which again results in a system with uniqe error dynamics at each particular frequency. At 1 $[Hz]$, both systems were observed to have the same error. This was because both systems had the same gain at this particular frequency.

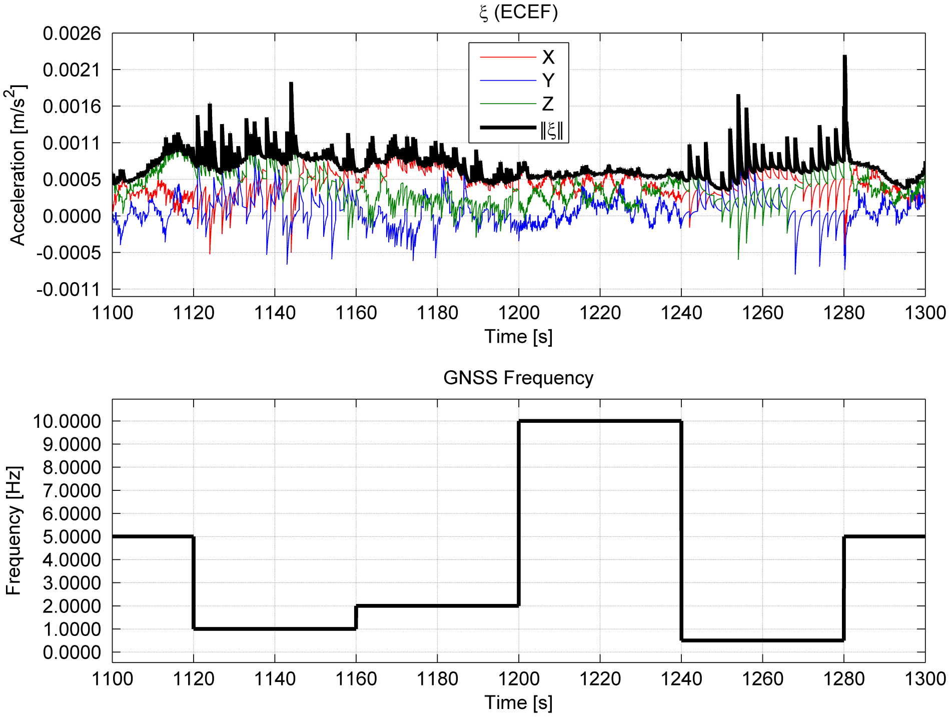


Figure 6.22: Case 3: Main Observer Error ($\boldsymbol{\xi}$) (Dynamic GNSS Gain)

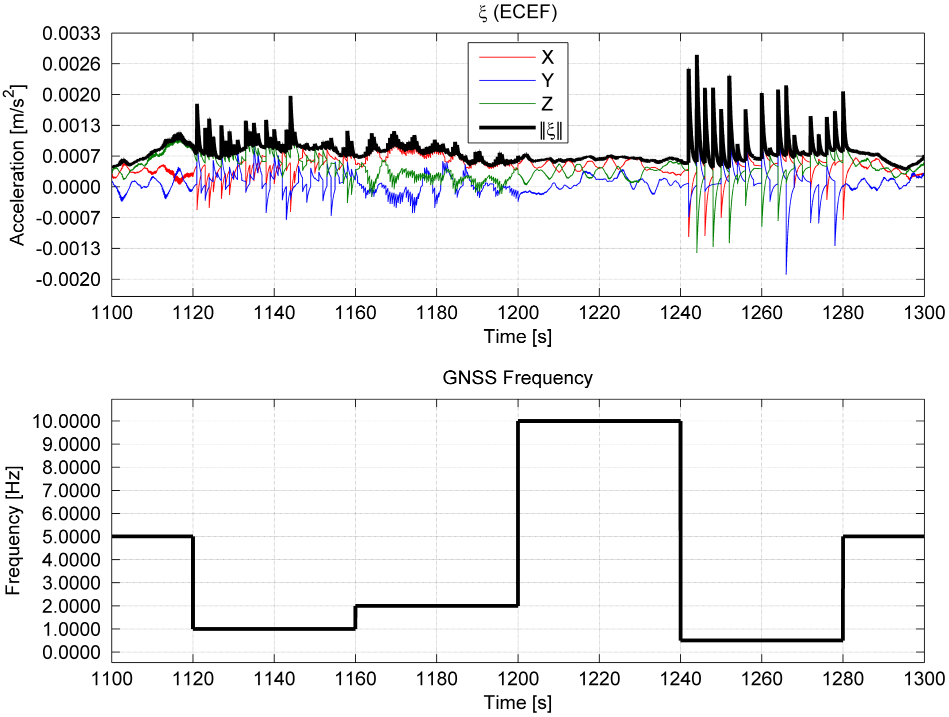


Figure 6.23: Case 3: Main Observer Error ($\boldsymbol{\xi}$) (Constant GNSS Gain)

6.4 Case 4: Dead-Reckoning

This case will demonstrate the performance of the nonlinear observer when the GNSS was absent. Without GNSS aiding the observer would be sensitive to IMU sensor errors, such as bias and white noise. The latter was found to be the hardest error to handle, and caused the estimates to drift away rapidly. Two simulations were therfore carried out: With and without noise. The reason for not including a low-pass filter will be explained under the discussion in Section 6.6. The GNSS aiding was turned off between $t = 1050$ $[s]$ and $t = 1250$ $[s]$, where the vessel followed the same position and heading references as the previous case studies.

6.4.1 Results

Figure 6.24 shows the position estimates without measurement noise. The slight deviation from the vessel's true position was due to small errors in the accelerometer bias estimates, which had an error under $7{\cdot}10^{-4}$ $[m/s^2]$. Figure 6.25 illustrates the substantial effect of adding white gaussion noise to the IMU mesaurements.

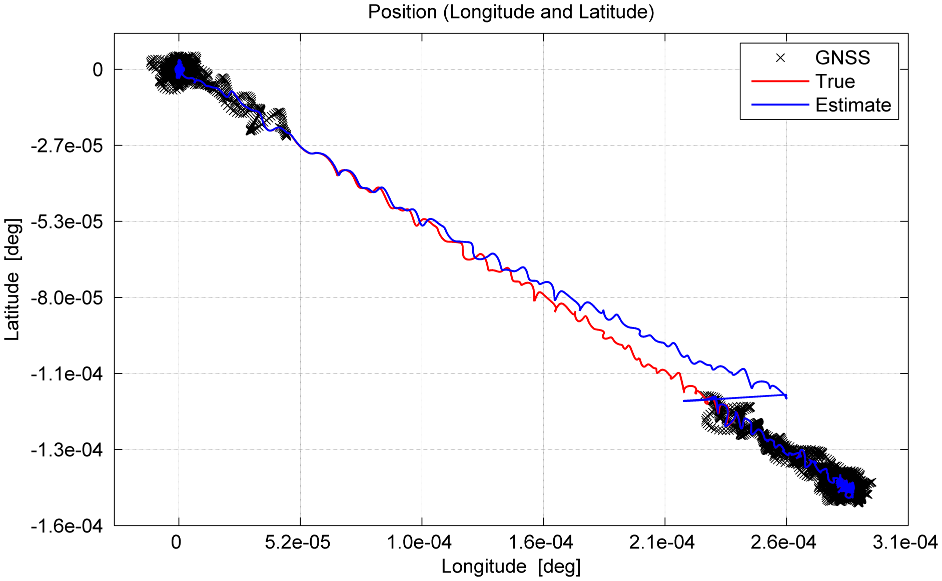


Figure 6.24: Case 4: Position (Without Noise)

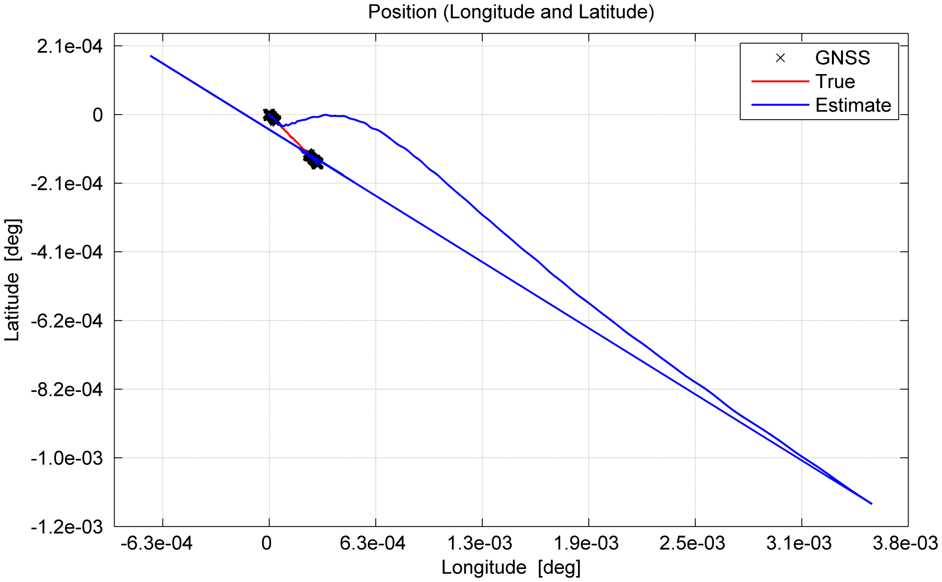


Figure 6.25: Case 4: Position (With Noise)

The velocity estimates can be seen by comparing Figure 6.26 with Figure 6.27. It can clearly be observed that the integration of white aggressive noise would give a randomly walking velocity estimate. A second integration would lead to a rapidly drifting position estimate, as illustrated in Figure 6.25.

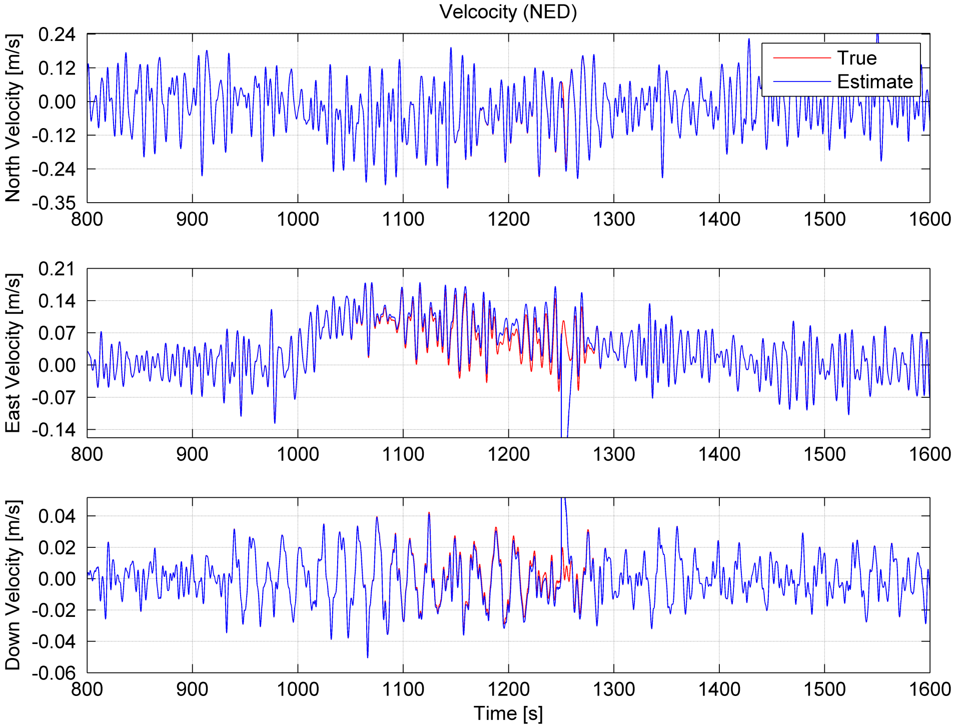


Figure 6.26: Case 4: Velocity (Without Noise)

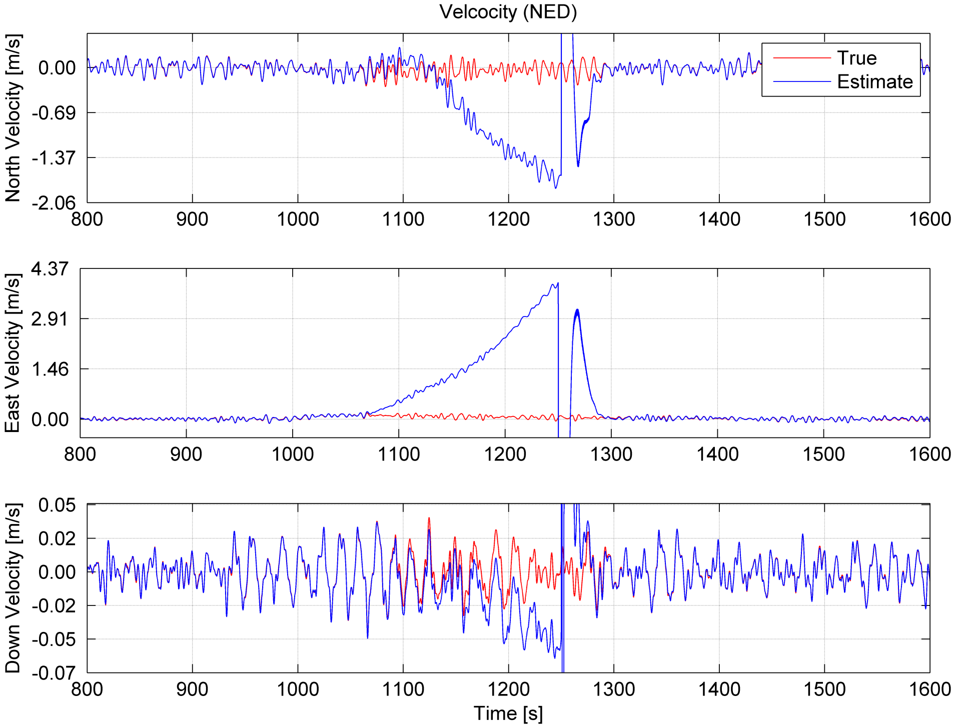


Figure 6.27: Case 4: Velocity (With Noise)

Furthermore, it can be seen that the rapid drift in the position estimates affected the Euler angle estimates in a negative manner, see Figure 6.29 and 6.27. This was a result of the nonlinear observer which utilized the position estimate for calculation of the attitude quaternion between BODY and NED. This calculation was done with the alternative representation in Section 4.3.5.

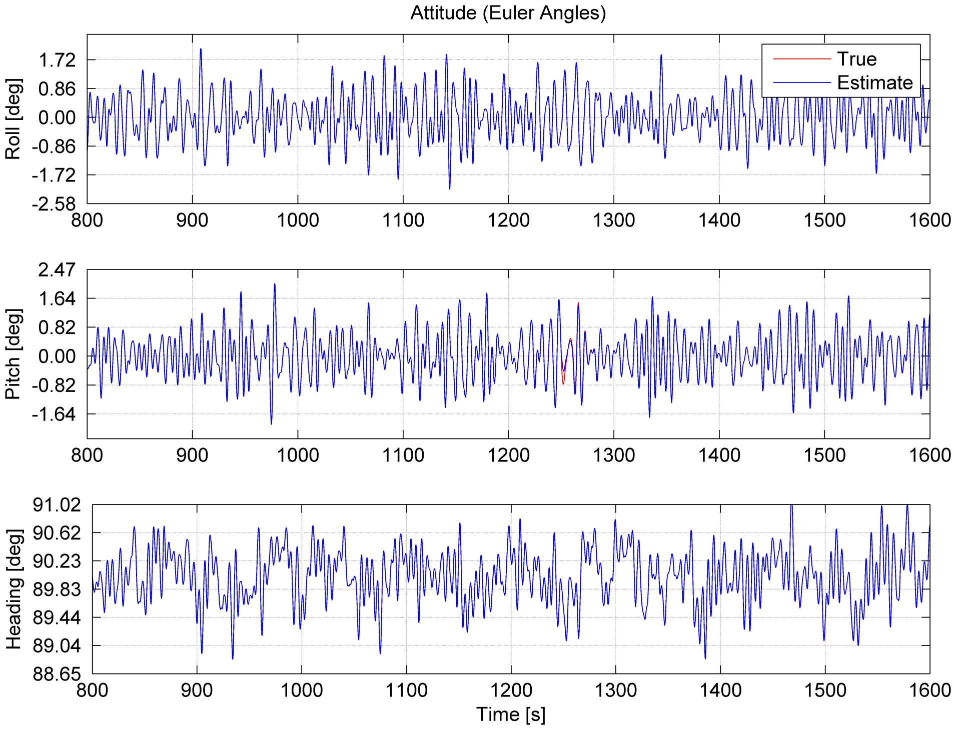


Figure 6.28: Case 4: Attitude (Without Noise)

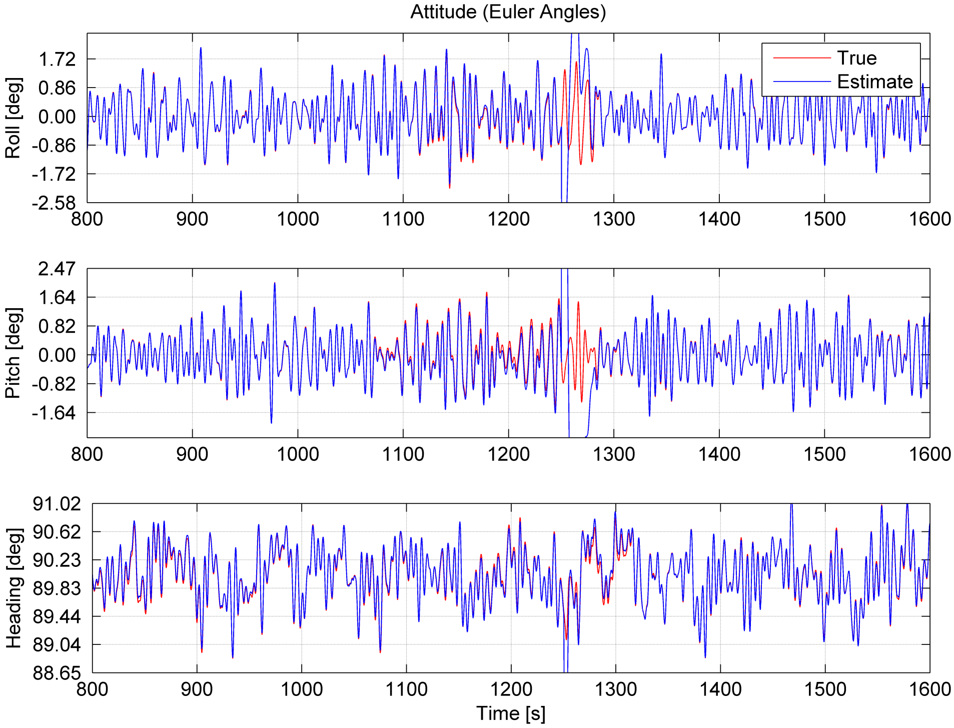


Figure 6.29: Case 4: Attitude (With Noise)

The PVA errors in the noise free simulation illustrated the effect of the small residuals between the true and estimated accelerometer biases, see Figure 6.30. Figure 6.31 showed the PVA errors under the influence of the additional sensor noise. Considering the east axis, the noise free case almost reached -3.0 $[m]$, whereas the noise inflicted case almost reached -300.0 $[m]$. The velocity ended up at with an error of -0.028 $[m/s]$ without noise, and -4.080 $[m/s]$ with noise. The noise free attitude estimate had an error under -0.02 $[deg]$, compared to the noise affected case with an error of nearly -0.23 $[deg]$. The noise affected case resulted in the estimates being approximately 100 times worse than the noise free case.

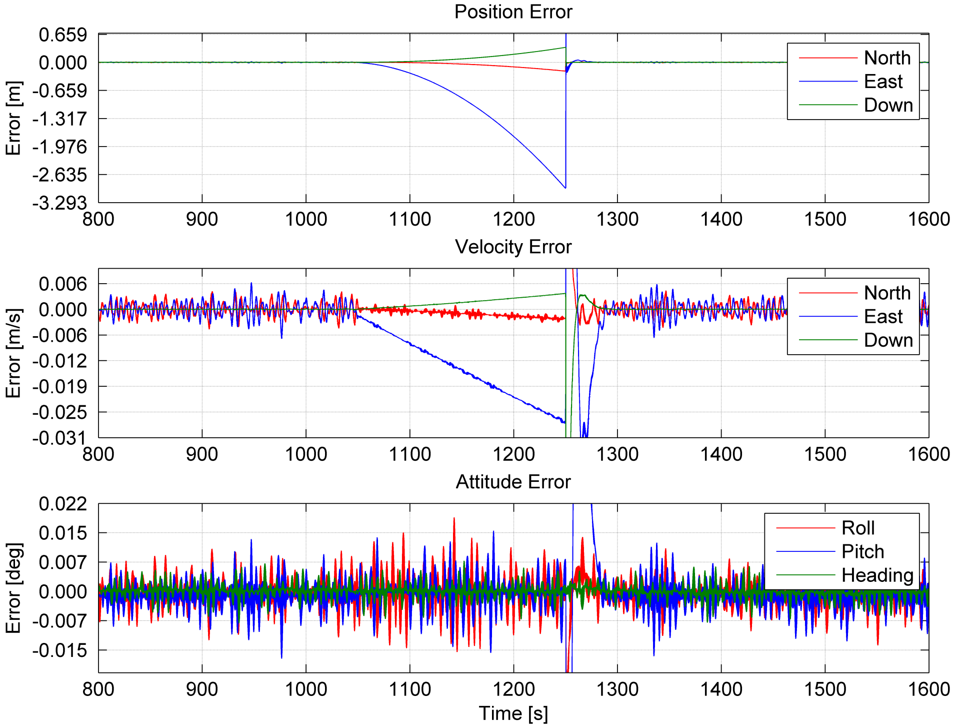


Figure 6.30: Case 4: Position, Velocity and Attitude Errors (Without Noise)

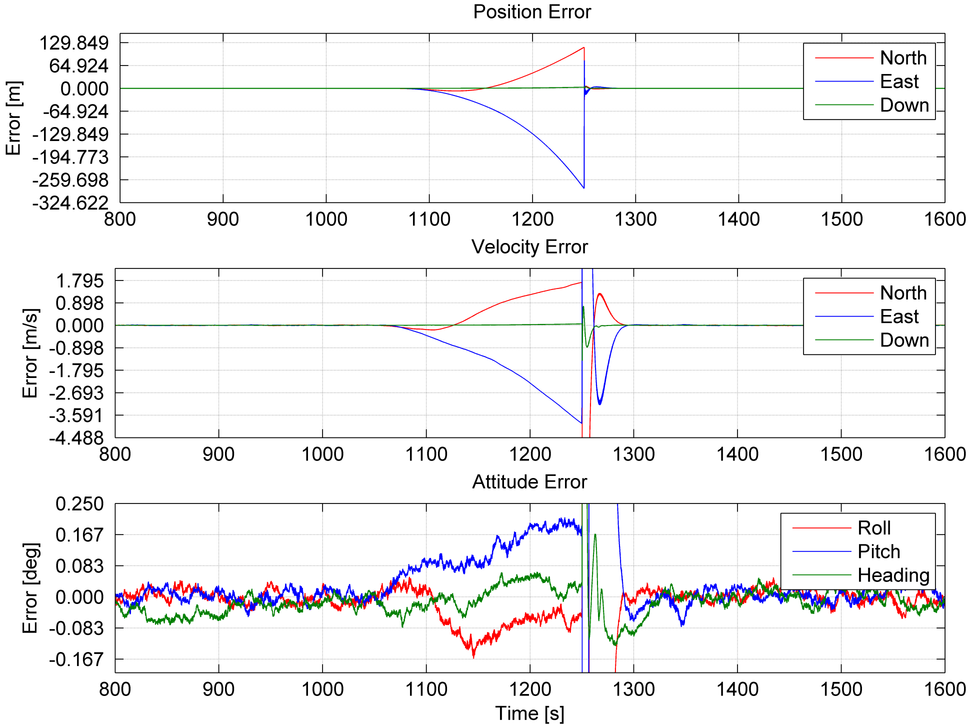


Figure 6.31: Case 4: Position, Velocity and Attitude Errors (With Noise)

6.5 Case 5: Nonlinear Observer versus Kalman Filter

This case study will compare the nonlinear observer with a Kalman filter (KF) in open-loop. The simulation was conducted under the presence of an unknown constant external force, using the same semi-submersible rig as in the previous case studies. The comparison was carried out by comparing the estimated position from both estimators. The constant external force was chosen to simulate ice gathering around the hull of the ship, moving the vessel in the direction of the moving ice. The force's magnitude was found by testing, such that the vessel was noticeably moved by the external force.

The external force was modeled by a fraction of the following vector:

$$\boldsymbol{\tau}_{ext} = \mathbf{M}_{RB}\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}^\top \tag{6.1}$$

which would mainly affect the vessel from the east, when the rig has the heading $\psi = 0$. The force was inflicted using a unit time step. The value of $\mathbf{M}_{RB}$ can be found in Appendix A.

Both state estimators were for simplicity expressed in the NED frame, see Appendix B for the setup of both estimators, with the corresponding design matrices for the KF. The KF was linearized under a constant heading $\psi = 0$, and implemented to only estimate the north and east position, in addition to the heading. The wave induced forces, previously included by the JONSWAP spectrum, were not included in this case. Instead, a small white noise process noise was included, such that the KF criteria of having process noise was fulfilled. The nonlinear observer was implemented using the same parameters as in the previous case studies.

The GNSS measurements was not affected by noise, due to the assumption of having a good satellite covarge. With a good coverage, the GNSS receiver would give noise free measurements, due to the assumption of being implemented by another KF. This would defy one of the criterias when implementing a KF, but would be realistic since this could occur in real applications under good conditions. In addition, the location of the GNSS was set to be same as the location of both state estimators, i.e. $\{gnss\} = \{m\} = \{b\}$. This would result in the lever arms $\mathbf{r}^b_{gnss} = \mathbf{r}^b_m = \mathbf{0}_{3\times1}$. The GNSS and gyrocompass measurement rates were set to 1 $[Hz]$ and 10 $[Hz]$, respectively. The IMU rate was set to 100 $[Hz]$. Both observers were implemented at the same rate as the IMU.

The simulation was divided in four phases: Between $t = 0$ $[s]$ and $t = 300$ $[s]$ the vessel was stationary; At $t = 300$ $[s]$, the vessel started moving diagonally to a new position reference, approximately 20 $[m]$ south and 30 $[m]$ east, while always keeping a constant heading $\psi = 0$; At $t = 500$ $[s]$, during the transit, the vessel was exposed to the unknown external force from the east, with a magnitude equal to $\frac{5}{100}\boldsymbol{\tau}_{ext}$; At $t = 950$ $[s]$, the external force increased to $\frac{1}{10}\boldsymbol{\tau}_{ext}$; Lastly, at $t = 1400$ $[s]$, both state estimators lost the GNSS aiding. The KF was tuned to approximately give same errors as the nonlinear observer, when the vessel wasn't affected by the external forces. The complete setup for this case can be found in Appendix B.

6.5.1 Results

From Figure 6.32 and 6.33, the effect of the external force can easily be observed. The yellow and green dot denote when the first and second unknown forces were applied, respectively. Since the forces were applied from the east, the north position was nearly unaffected. It is possible to see that the position estimates from the KF starts to struggle when the forces were inflicted. The cyan colored dot denoted when the GNSS aiding was lost.

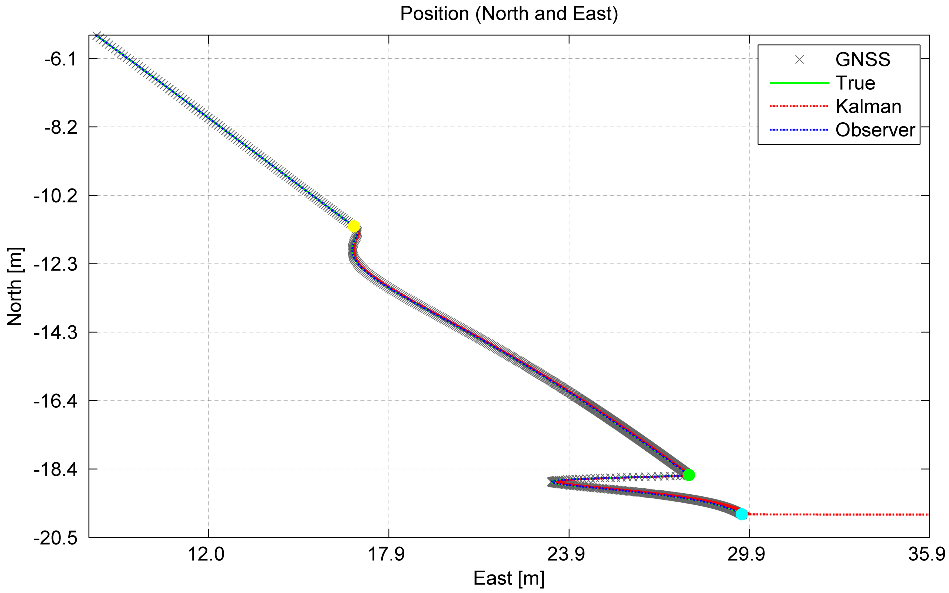


Figure 6.32: Case 5: Position (North vs. East)

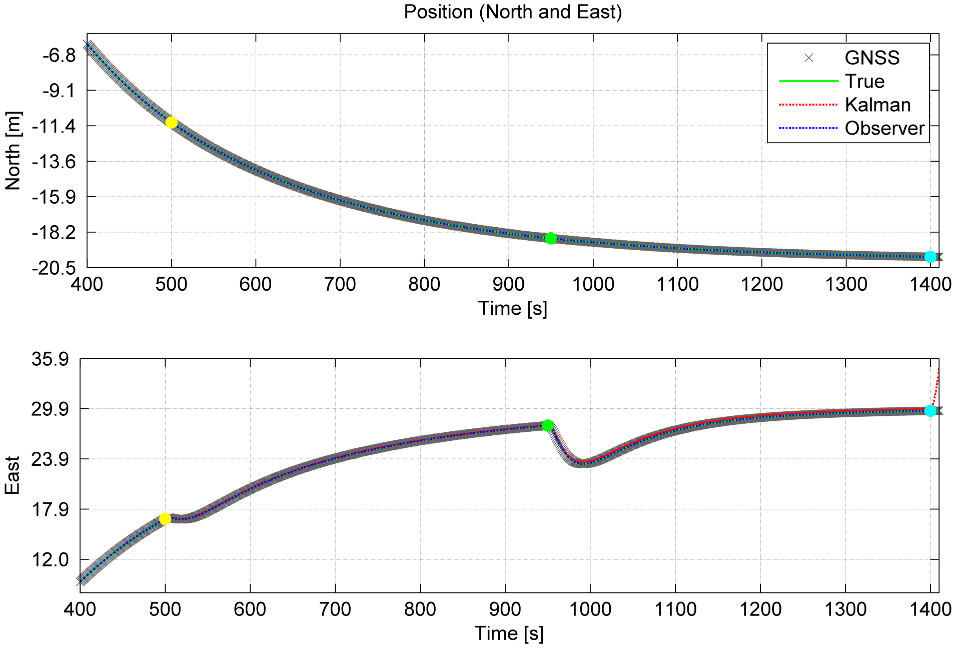


Figure 6.33: Case 5: Position (North and East)

Figure 6.34 shows the position error during the first 500 seconds. Both state estimators had an error equal to zero when being stationary. At $t = 300$ $[s]$, the vessel started moving, which can be reflected in the position error. Both navigation systems provided relatively small errors, whereas the KF estimates were somewhat larger. This was due to the bias term $\mathbf{b}$ needing time to converge to its new values, because of small nonlinearities in the model. The bias term $\mathbf{b}$ estimates the unmodeled dynamics in the KF, see Appendix B.

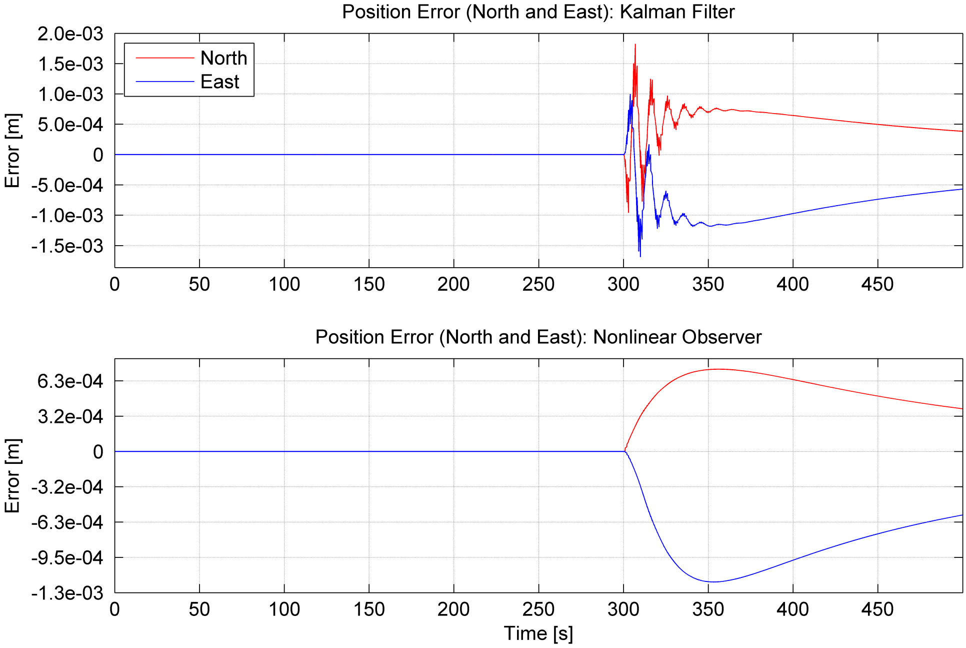


Figure 6.34: Case 5: Position Error (Without External Forces)

When the vessel was affected by the external forces, the KF clearly deviated from the true position, which can be seen in Figure 6.35. The first external force resulted in the KF having a constant estimation error between -0.030 $[m]$ and -0.073 $[m]$ to the east, whereas the increased force gave an error between -0.157 $[m]$ and -0.290 $[m]$. The estimates from the nonlinear obseserver was nearly unaffected by the external forces, with an error smaller than $2 \cdot 10^{-3}$ $[m]$. The KF error seemed to increased proportional with the size of the external force.

Figure 6.36 illustrated the position error between two GNSS measurements, when the KF was affected by the second force. This was clearly a result of the external force not being properly handled by the KF.

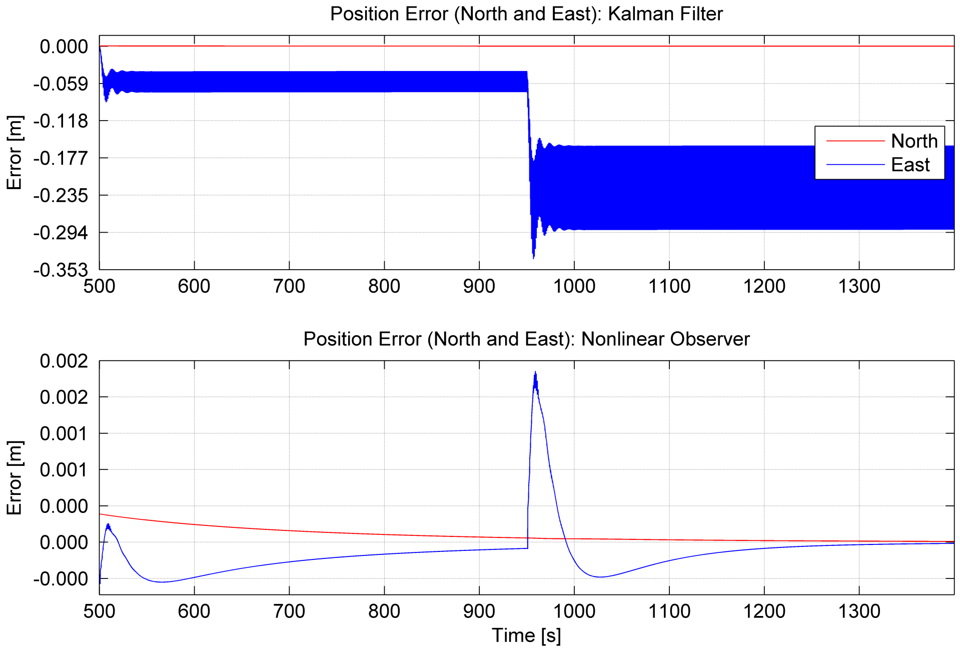


Figure 6.35: Case 5: Position Error (With External Forces)

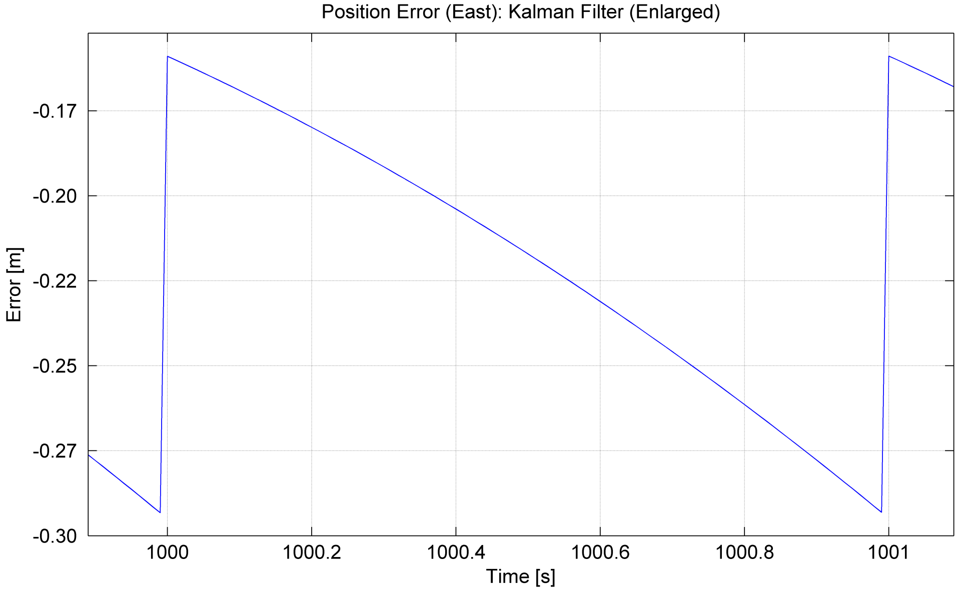


Figure 6.36: Case 5: Position Error for the KF between two GNSS measurements (With External Forces)

At $t = 1400$ $[s]$, the GNSS aiding was turned off. The resulting effect of the external force without GNSS measurements can be observed in Figure 6.37, where the KF drifted almost 1227 $[m]$ in 200 $[s]$. The nonlinear observer drifted approximately $2.3 \cdot 10^{-4}$ $[m]$

in the same time period, probably due to numerical errors.

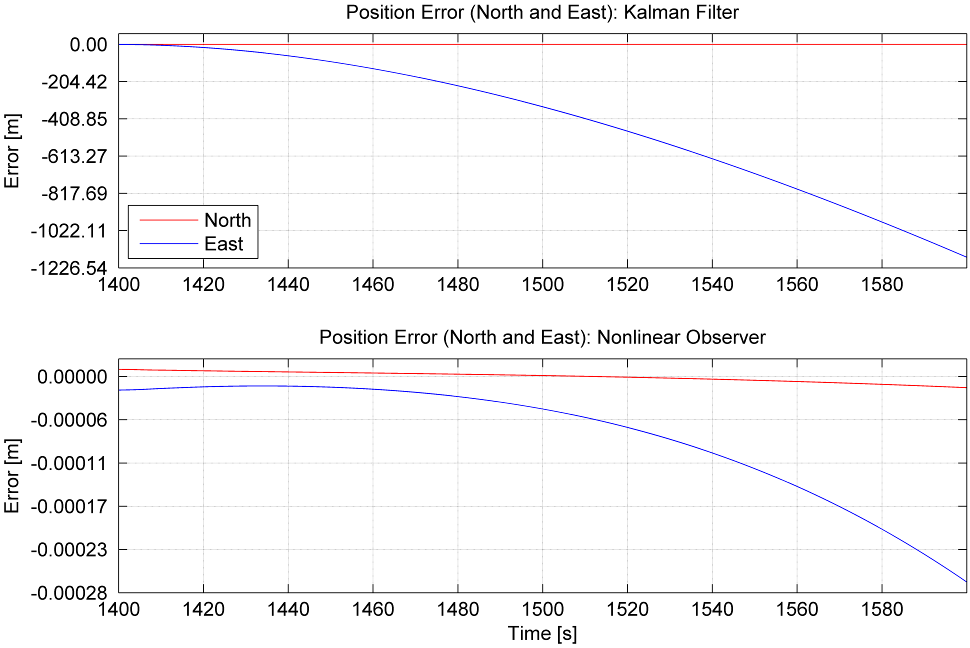


Figure 6.37: Case 5: Position Error (With External Forces and Dead-Reckoning)

6.6 Discussion

The mean filtering method gave the best accelerometer bias estimates. This was due to the assumption of having an evenly loaded vessel with roll and pitch angles with zero mean, which was the case for the simulations. In the case of having an unevenly loaded vessel, the mean filtering method would result in incorrect attitude and position estimates. This problem could be solved by having some apriori information about the vessel's angle offset during different loads, which could be used to correct this particular estimation method.

The vector bias method relied on having an independent and sufficient varying input vector for convergence, due to the PE condition. Under the current seastate, provided by the JONSWAP spectrum with $H_s = 1$ $[m]$ and $\omega_0 = 1.2$ $[rad/s]$, the input vector $\boldsymbol{\phi}$ resulted in slowly converging bias estimates. The vector bias method was also sensitive to sensor noise, which according to [19] would lead to biased estimates. In the case of having a sea-state with sufficient frequencies and wave amplitudes, the vector bias method should still be considered for future testing, where any sensor noise can be filtered to prevent any bias in the estimates.

The solution to the accelerometer bias problem can be carried out with a combination of the two proposed methods. Some logic could be created and implemented to achieve the best possible accelerometer bias estimates, by utilizing additional knowledge of the ship's angle offset during different loads, measurements from VRUs/MRUs, and the IMU's sensor characteristics.

The nonlinear observer provided great estimates when it was consitently aided by the GNSS, both at constant and varying GNSS frequencies. This was realized by the dynamic GNSS gain, where the results showed that this gain provided consistent error dynamics during inconsistent GNSS measurement rates.

It was also discovered that the nonlinear observer was robust on the choice of the GNSS gain, especially at higher GNSS frequencies. As long as $\mathbf{H}_{K_d}$ was Hurwitz and $\|\mathbf{H}_{K_d}(s)\|_\infty$ was sufficiently low, the observer resulted in estimates with small errors. At lower GNSS frequencies, the observer was found to be more sensitive to the choice of GNSS gain, see 4.4.2. Any additional time delays in the GNSS measurements could therefore cause instability in the error dynamics. In this particular case, the dynamic gain should be utilized for stability.

The presence of error residuals, between the true and estimated biases, clearly affected the estimates when the GNSS aiding was absent. To reduce these errors, the nonlinear observer should be further tuned. But at a certain point, these residuals will be limited by the sensor characteristics of the chosen IMU.

The noise was discovered to affect the estimates with a substantial impact during the dead-reckoning phase. To solve this problem a first order digital low-pass filter with different cut-off frequencies was implimented to remove the noise from the IMU. This was carried out without any mentionable success, due to the aggressive sensor noise. A third order Butterworth filter was also tested without any luck.

As the cut-off frequency for the filters got lower, the noise diminished at the cost of larger phase-delay. Before the noise was satisfactionally removed, the time delay of the filtered measurements was so big that it resulted in a synchronization mismatch between the IMU and the GNSS measurements. This made the position transformation from $\{gnss\}$ to the $\{m\}$ too erroneous.

Fortunately, it was later discovered that the real IMU measurement noise was far more gracious than the simulated noise. Both the simulated and the real measurement noise had approximately the same power density, but it was discovered that the real measurement noise was more correlated with the real signal. This increased the signal-to-noise ratio (SNR) and resulted in better estimates. It should be mentioned that the use of a low-pass filter did improve the performance of the nonlinear observer at some degree, but not sufficiently due to the simulated noise characteristics.

The results from the state estimator comparison illustrated a weakness in the model-based estimation approach. Unless all of the dynamics were modeled or sufficiently accounted for, deviations from the true states would occur. To handle this unmodeled dynamics, the Kalman filter tried to estimate the nonlinear dynamics in real-time. The design matrices were designed such that the unmodeled dynamics were sufficiently handled under the current sea-state. When the external forces were applied to the vessel, the Kalman filter didn't manage to adapt to the new sea-state. This resulted in inaccurate estimates. In practical cases, the Kalman filter will be designed such that it handles a variety of different sea states, [15]. But at some point, when the forces are big enough, deviations will occur. The nonlinear observer measured the inflicted forces using the IMU, and was unaffected by the external forces.

The position estimate was found to negatively affect the Euler angle estimate in the observer, which was a result of the position dependancy when calculating the unit quaternion between BODY and NED, see Section 4.3.5. Since the nonlinear observer was expressed using the ECEF frame, the attitude observer's main task was to estimate the attitude quaternion between BODY and ECEF. This attitude estimate was not affected by the posisition estimate in the same manner.

Since the heading was calculated from the Euler angle ψ, and because it's crucial to have a good heading estimate for marine navigation, the nonlinear observer should in future applications be concidered in the NED frame. This would make the roll, pitch and heading estimates independent of erroneous position estimates. Complete independancy was difficult to achieve, since the injection term utilized the estimated specific force, which was dependent on GNSS measurements for convergence. An alternative approach could be to replace the estimated force with the gravity vector obtained from the gravity model. This should be sufficient when the vessel is non-accelerating, i.e. dynamic positioning.

The gyrocompass substitute in the injection term seemed to result in great heading estimates from the observer. Since the substitute assumed $\phi = 0$ and $\theta = 0$, only the z-axis in the corresponding term was utilized in the attitude observer's injection term. This modification should remove any errors when $\phi \neq 0$ and $\theta \neq 0$.

The redefined injection term is:

$$\hat{\boldsymbol{\sigma}} := k_1 \underline{\mathbf{f}^b} \times \mathbf{R}_b^e(\hat{\mathbf{q}}_b^e)^\top \underline{\hat{\mathbf{f}}^e} + k_2 \mathbf{I}_{gyc} \underline{\mathbf{c}^b} \times \mathbf{R}_b^e(\hat{\mathbf{q}}_b^e)^\top \underline{\mathbf{c}^e} \tag{6.2}$$

where

$$\mathbf{I}_{gyc} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.3}$$

Chapter 7

Experimental Results and Discussion

The experimental tests were executed in Egersund, Norway, in collaboration with Marine Technologies LLC. The tests were carried out on a trolley, using the ADIS16485 MEMS based IMU. The original plan was to test the nonlinear observer on a supply boat, but this wasn't realized due to restrictions when accessing the boat's GNSS and gyrocompass. The setup for the tests can be reviewed in Chapter 5.

The mean filtering method, chosen to estimate the accelerometer bias, assumed that the roll and pitch angles had zero mean. To make this assumption valid, the Orion INS was used as a reference to level the trolley during the initialization phase. The Orion had a roll and pitch accuracy at 0.01 $[deg]$, see 5.1 in Chapter 5. The gyroscope bias was assumed correct estimated when the mean values of all three axis were zero over some time period. The trolley was kept stationary and level for about 1000 $[s]$ to ensure that all estimates had converged, using the GUI as aidance. This exact initialization procedure was executed for each test.

It was observed that the nonlinear observer missed a considerable number of GNSS measurements during the experimental tests. This was due to some unfortunate logic when handling and matching new GNSS measurements with the corresponding IMU measurement. This resulted in a degraded performance of the estimates, especially the position estimates. For this reason, an offline MATLAB simulation was carried out for each experimental test, using the improved version of the nonlinear observer which included the dynamic GNSS gain. This was made possible since the logging software successfully logged the raw data from the sensors.

In the first case, ideal measurement rates were considered, see Table 7.1 for the rates. Due to the GNSS matching problem mentioned above, this was hard to realize in the experimental test. Secondly, the gyrocompass was turned off to see how much the heading drifted. Lastly, the performance during GNSS outage was tested. The latter case was divided in three parts and tested the dead-reckoning performance of both the nonlinear observer and the reference system.

This chapter is organized in the following way:

- Section 7.1: Case 1: Ideal Measurement Rates
- Section 7.2: Case 2: Loss of Gyrocompass
- Section 7.3: Case 3: Loss of GNSS
- Section 7.4: Discussion

The sensor and observer configuration setup can be seen in Table 7.1. Table 7.2 shows the sensor parameters for the online[1] experimental tests and the additional offline[2] simulation. The upper bound on the estimated bias was found by taking the mean of the gyroscopes during the initalization phase. Further, the observer tuning parameters for the experimental tests can be seen in Table 7.3, where the tuning parameters for the offline simulation can be found in Table 6.3 in Chapter 6.

Remarks

The Orion INS wasn't able to correctly interprete the altitude from the GNSS receiver, where the estimated altitude consistently gave estimates around 0 $[m]$. This was troubleshooted for a long time, without any luck. Despite this error, the estimated position in the horizontal plane seemed to be correct and precise. Beacuse of this problem, the GNSS location wasn't taken into account, due to fear of introducing additional errors. The lever arms for the state estimators and the GNSS can be seen in Table 7.1.

To ensure a faster convergence of the observer, the attitude gains k_1, k_2 and k_I were multiplied by 20 during the initialization phase, which ended at $t = T_{init}$. The NED frame was chosen for intuitive interpretation of the position estimates from both the nonlinear observer and the Orion INS. Since the logging software logged the estimated positions in LLA coordinates, the coordinates were transformed to the NED frame using `geodetic2ned` in MATLAB. This would introduce neglible errors, since the navigation

[1] Denotes the actual experimental test

[2] Denotes the offline simulation based on the logged raw data

area was small. The original LLA data was additionally plotted in the same manner as in Chapter 6, where the initial LLA coordinate was subtracted from the original data. Additionaly, the IMU measurements in the experimental tests were filtered by a first order low-pass filter with cut-off frequency $f_{c,lp} = 5\ [Hz]$. This cut-off frequency did not add any noticeable time delay to the measurements.

Description	Value
Placement of the GNSS relative to CO	$\mathbf{r}^b_{gnss} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top\ [m]$
Placement of the IMU relative to CO	$\mathbf{r}^b_{m} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top\ [m]$
GNSS frequency	$F_{gnss} = 5\ [Hz]$
Gyrocompass frequency	$F_{gyc} = 10\ [Hz]$
IMU frequency	$F_{imu} = 500\ [Hz]$
Observer frequency	$F_{obs} = 50\ [Hz]$

Table 7.1: Test Configuration (Online and Offline)

Description	Value
Upper bound of the specific force norm	$M_f = 9.81\ [\frac{m}{s^2}]$
Lower bound of the specific force norm	$m_f = 0.98M_f\ [\frac{m}{s^2}]$
Ensure well-defined specific force	$\delta = 0.98m_f\ [\frac{m}{s^2}]$
Upper bound on the estimated gyroscope bias	$M_{\hat{b}_g} = 1.6 \cdot 10^{-3}\ [\frac{rad}{s}]$

Table 7.2: Observer Sensor Parameters (Online and Offline)

Description	Value
Parameter to guarantee stability	$\theta_{gnss} = 2$
Constant GNSS gain	$k_p = 1$
IMU parameter	$k_1 = 0.5$
Gyrocompass parameter	$k_2 = 5.0$
Gyroscope bias parameter	$k_I = 0.001$
Accelerometer bias cut-off frequency (high)	$f_{c,high} = 5 \cdot 10^{-2}\ [Hz]$
Accelerometer bias cut-off frequency (low)	$f_{c,low} = 5 \cdot 10^{-6}\ [Hz],$
Additional accelerometer bias parameters	$T = 1000\ [s],\ \beta = 0.1$
Observer initialization time	$T_{init} = 550\ [s]$
Cut-off frequency First order low-pass filter	$f_{c,lp} = 5\ [Hz]$

Table 7.3: Observer Tuning Parameters (Online)

7.1 Case 1: Ideal Measurement Rates

This section will consider the performance of the nonlinear observer when it was ideally aided by the GNSS at 5 $[Hz]$. As earlier mentioned, the observer randomly missed out some measurements, making the GNSS rate somewhat inconsistent with a mean below the ideal rate. The trolley was moved at a low speed in a pattern as illustrated in Figure 7.1.

7.1.1 Results

From Figure 7.1, 7.2, 7.3 and 7.4, the GNSS measurements suddenly jumped away from its initial position almost immidiately after moving the trolley. This was caused by satellite interference. Both the Orion and the nonlinear observer experienced the same behaviour during the first seconds of the transit, before once again converging to position measurements. When studying the offline simulations, the nonlinear observer performed better than the corresponding online test due to a consistent GNSS rate.

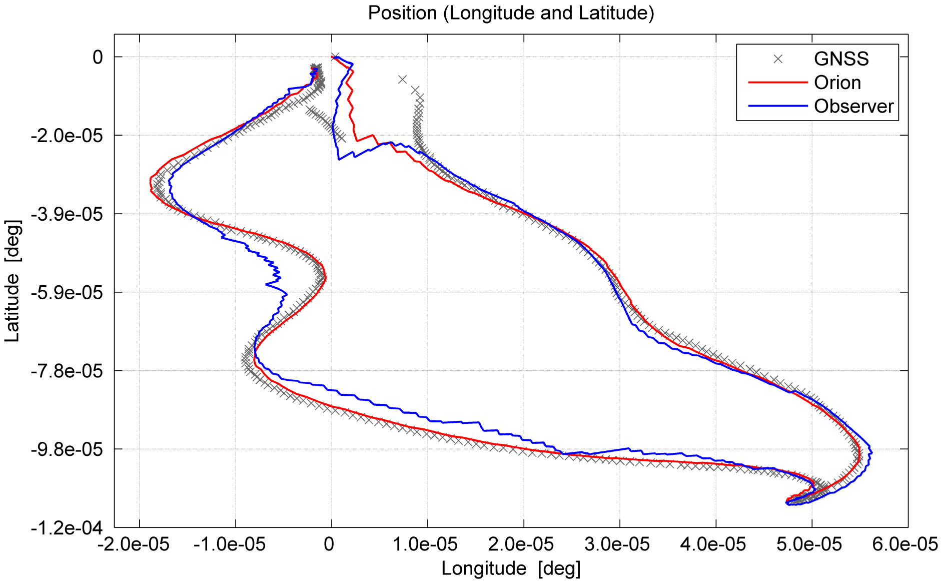


Figure 7.1: Case 1 (Online): Position (Latitude and Longitude)

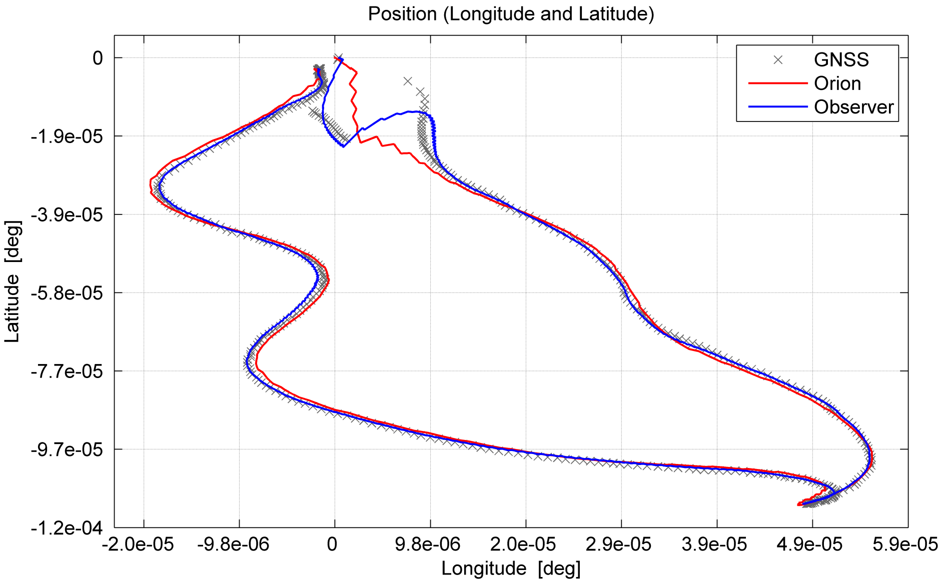


Figure 7.2: Case 1 (Offline): Position (Latitude and Longitude)

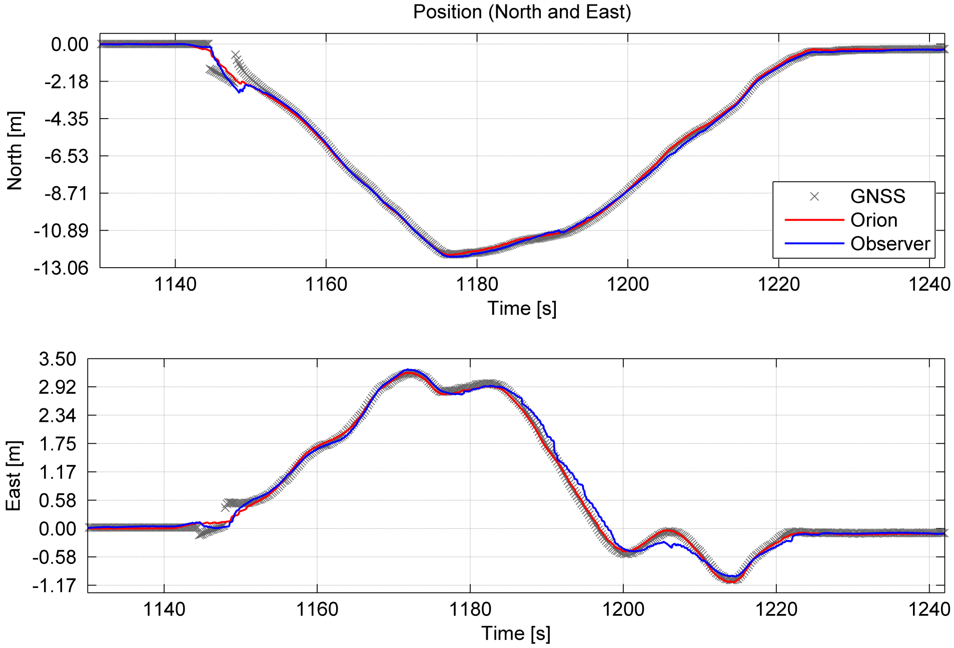


Figure 7.3: Case 1 (Online): Position (North and East)

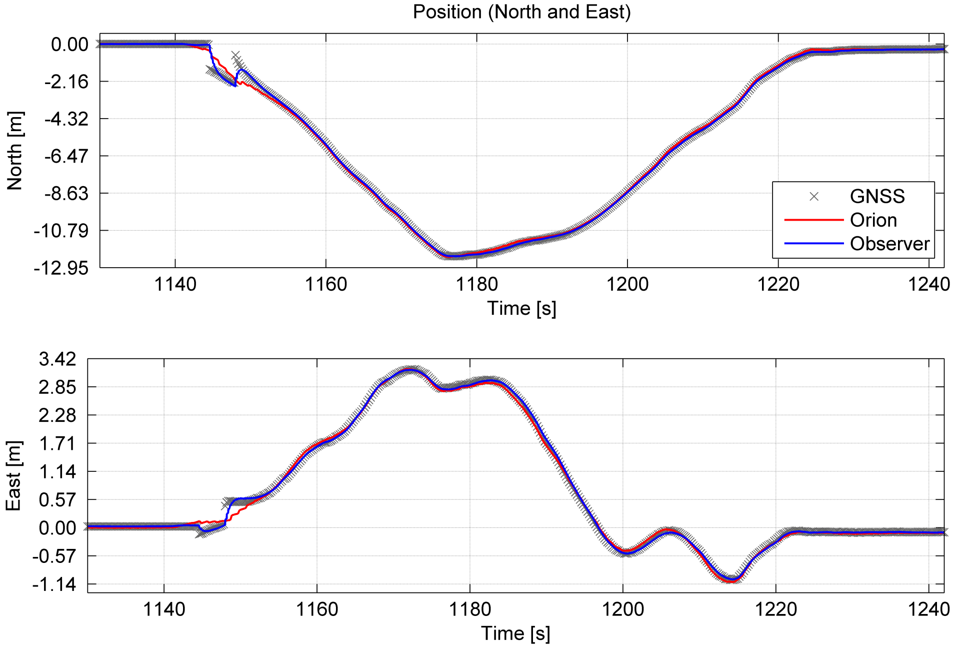


Figure 7.4: Case 1 (Offline): Position (North and East)

The position error in Figure 7.5 and 7.6 show the deviation between the Orion and the nonlinear observer during the experimental test and the offline simulation, respectively. Without considering the GNSS position jump, the online test yielded position errors under 0.45 $[m]$ and 0.36 $[m]$ in north and east, respectively. For comparison, the offline test resulted in an position error under 0.19 $[m]$ and 0.15 $[m]$ in the same axes. This gave an improvement of approximately 58 [%] for the offline simulation, compared to the online test.

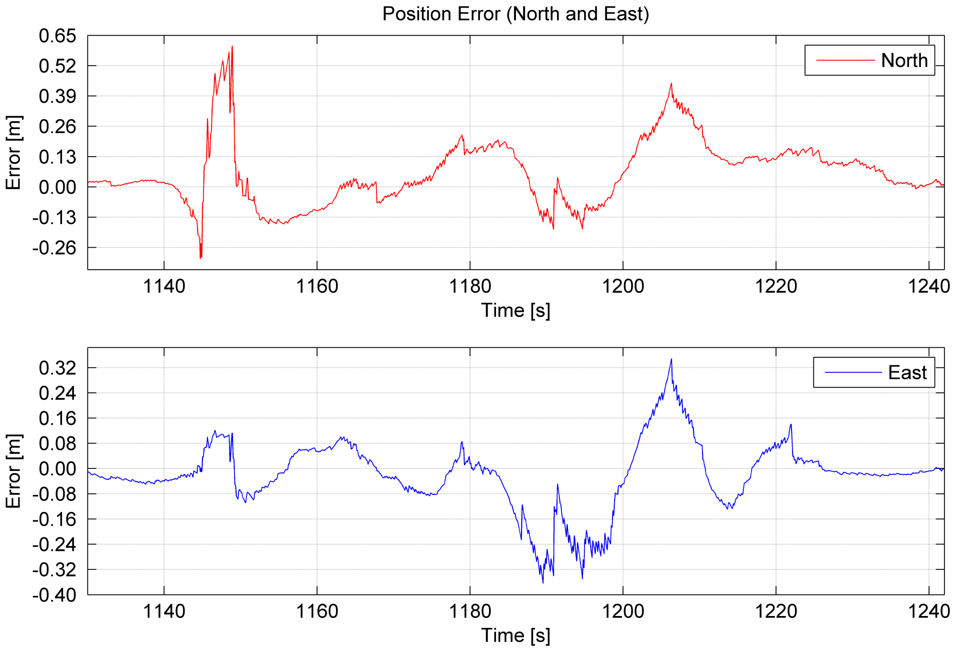


Figure 7.5: Case 1 (Online): Position Error (North and East)

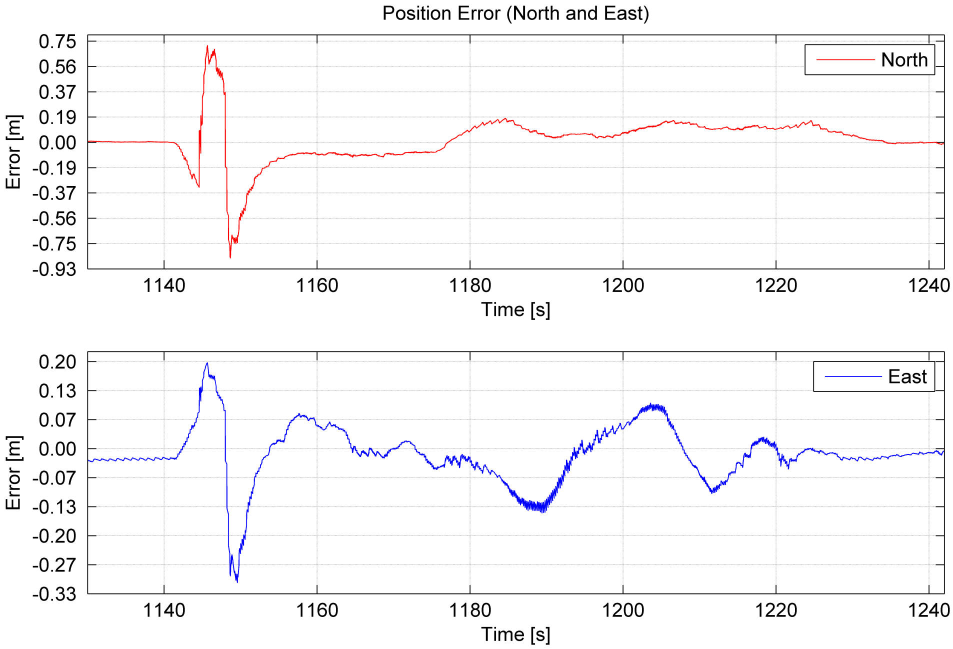


Figure 7.6: Case 1 (Offline): Position Error (North and East)

Due to the lower k_1 gain from Table 7.3 and the loss of GNSS measurements, the Euler angle attitude was observered to have a delay behind the reference system, see Figure

7.7. This was a result of the alternative attitude representation given in Section 4.3.5, which was dependent on the estimated LLA coordinate. As the position estimates in the offline simulations were better, this gave rise to better attiude estimates. See Figure 7.8 for illustration.

The attitude errors, for both the online test and the offline simulation, can be seen in Figure 7.9 and 7.10. The errors were reduced in the offline simulation, compared to the online test. The offline simulation resulted in a roll error under 0.22 [*deg*], pitch error under 0.27 [*deg*] and a heading error under 0.17 [*deg*]. These errors were most likely due to sensor errors and misalignments of the trolley during the initialization phase.

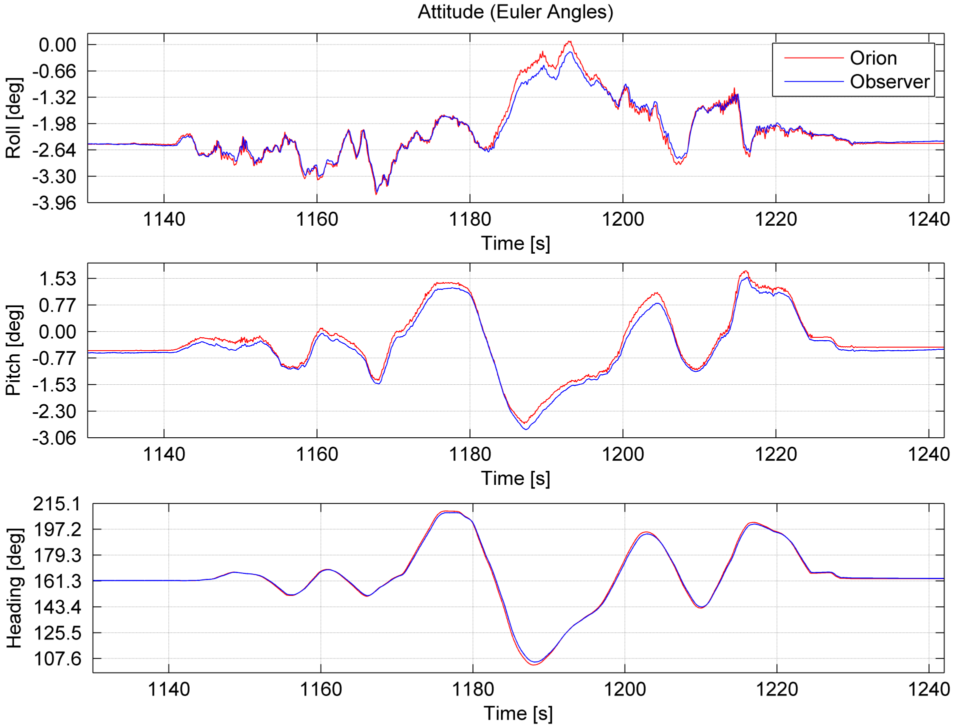


Figure 7.7: Case 1 (Online): Attitude (Euler Angles)

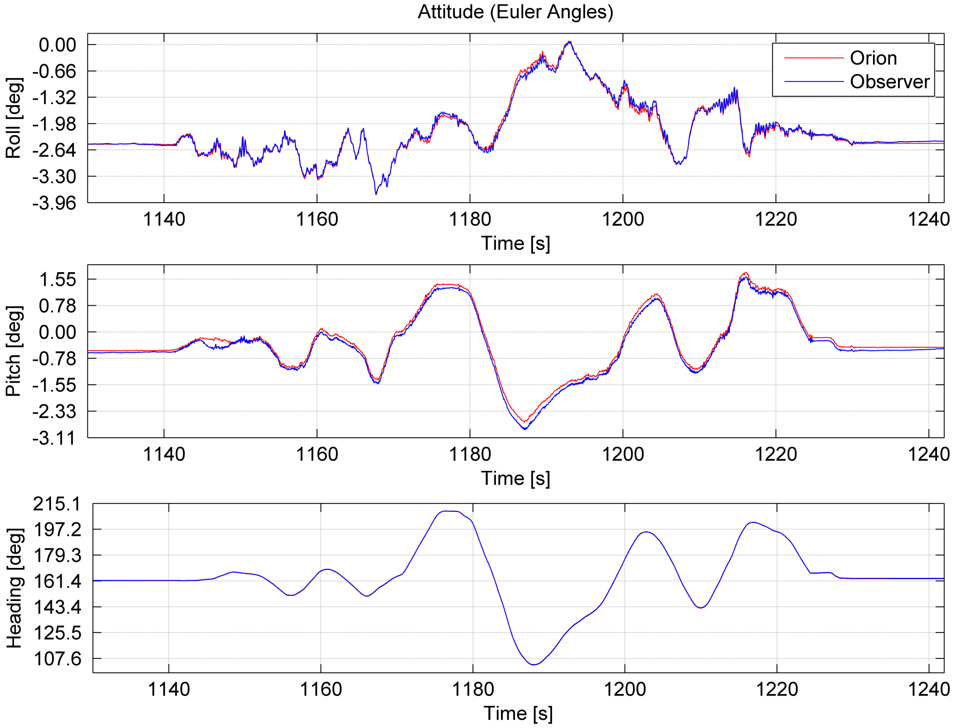


Figure 7.8: Case 1 (Offline): Attitude (Euler Angles)

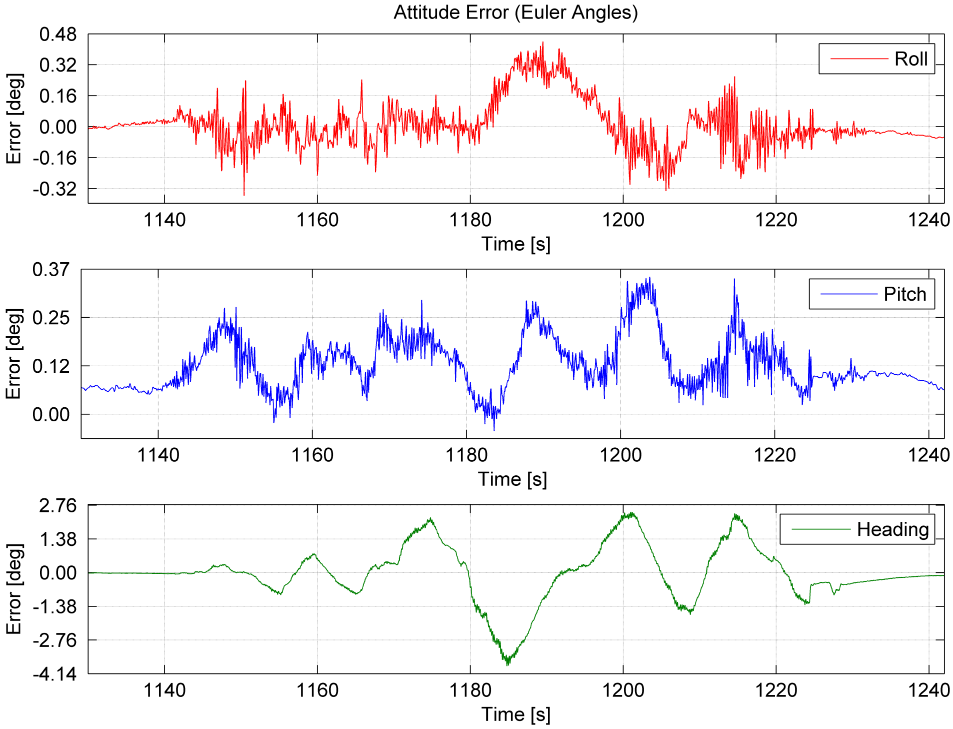


Figure 7.9: Case 1 (Online): Attitude Error (Euler Angles)

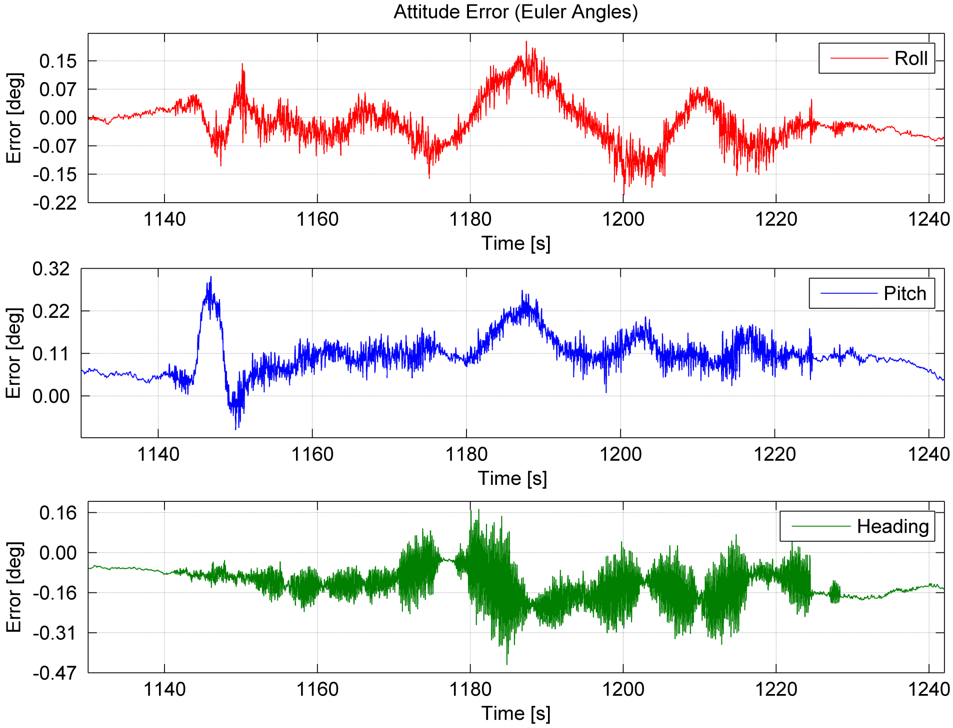


Figure 7.10: Case 1 (Offline): Attitude Error (Euler Angles)

Both the estimated gyroscope and accelerometer bias seemed converge in both the experimental test and the offline simulation, see Figure 7.11 and 7.12. The stippled lines indicated the offline estimate of the true sensor bias, and were calculated by taking the mean of measurements when the trolley was level and stationary during the initialization phase. Due to a higher gyroscope gain k_I in the offline simulation, the bias was estimated more aggressively.

The deviation from the indicated sensor bias and the estimated values were most likely due to a random walk in the biases. This random walk will be considered in more detail in Section 7.3.3, where the trolley was stationary and level during the whole test. The spikes observed right after $t = 600$ $[s]$ was due to a bug in the logging software.

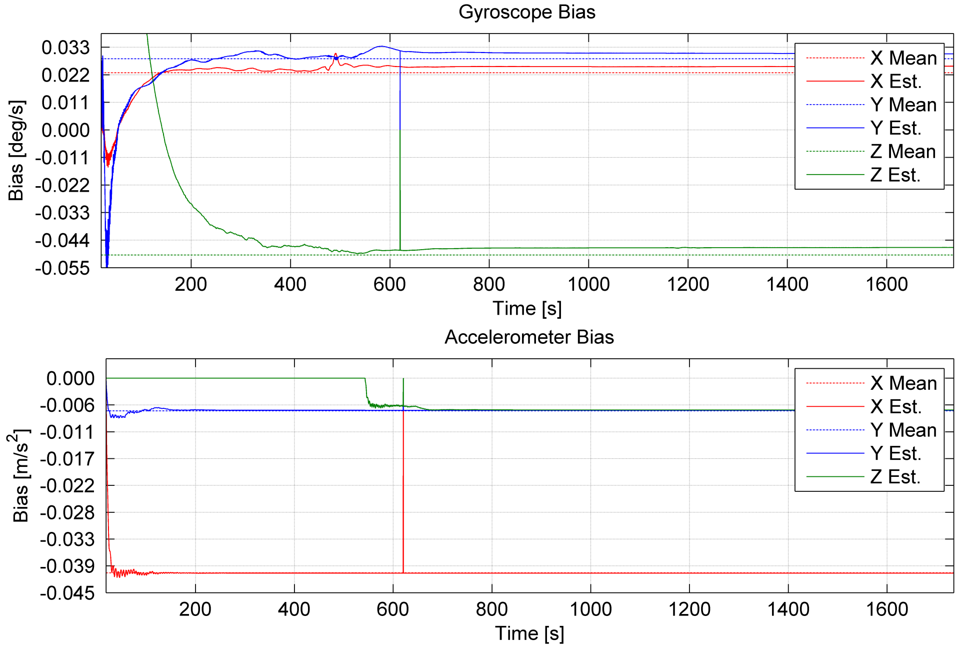


Figure 7.11: Case 1 (Online): Gyroscope Bias and Accelerometer Bias

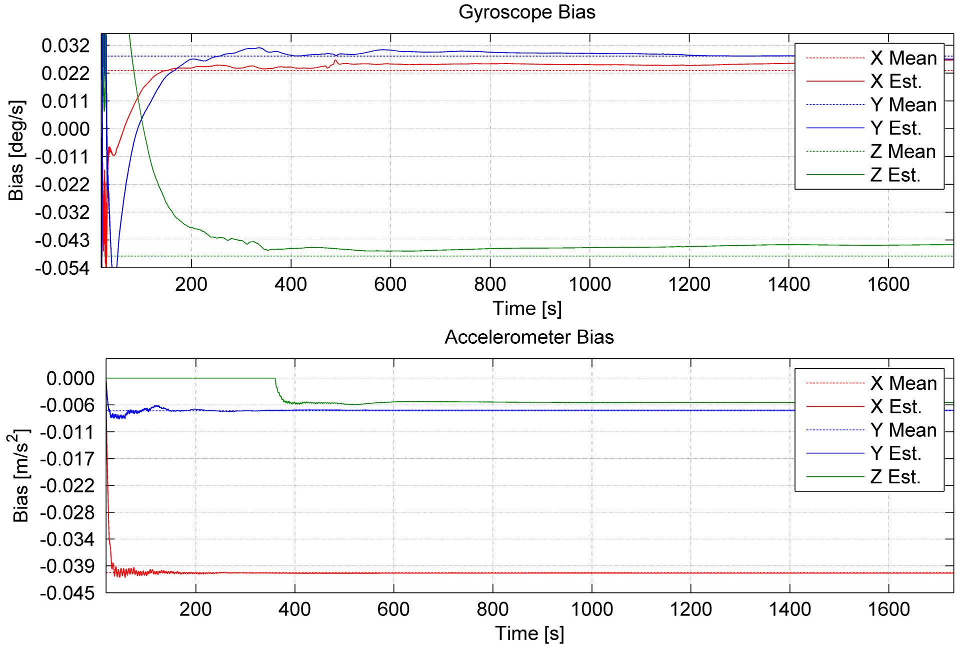


Figure 7.12: Case 1 (Offline): Gyroscope Bias and Accelerometer Bias

The main observer error $\boldsymbol{\xi}$ was observed to be sensitive to the GNSS deviation, which occured during the first seconds of the transit, see Figure 7.13 and 7.14. The offline simulation resulted in a higher spike during this GNSS offset, due to the more aggressive dynamic gain. This more aggressive behaviour can also be reflected in the position estimates in Figure 7.2 and 7.4, where the estimates followed the GNSS measurements more closely. During the offline simulation, $\boldsymbol{\xi}$ was observed to have an overall lower error, compared to the online test result.

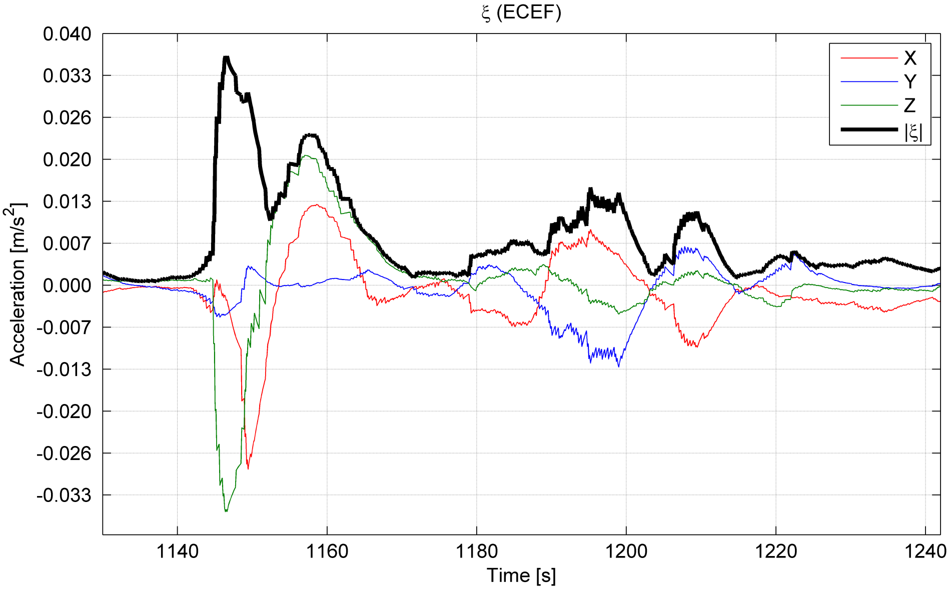


Figure 7.13: Case 1 (Online): Main Observer Error (ξ)

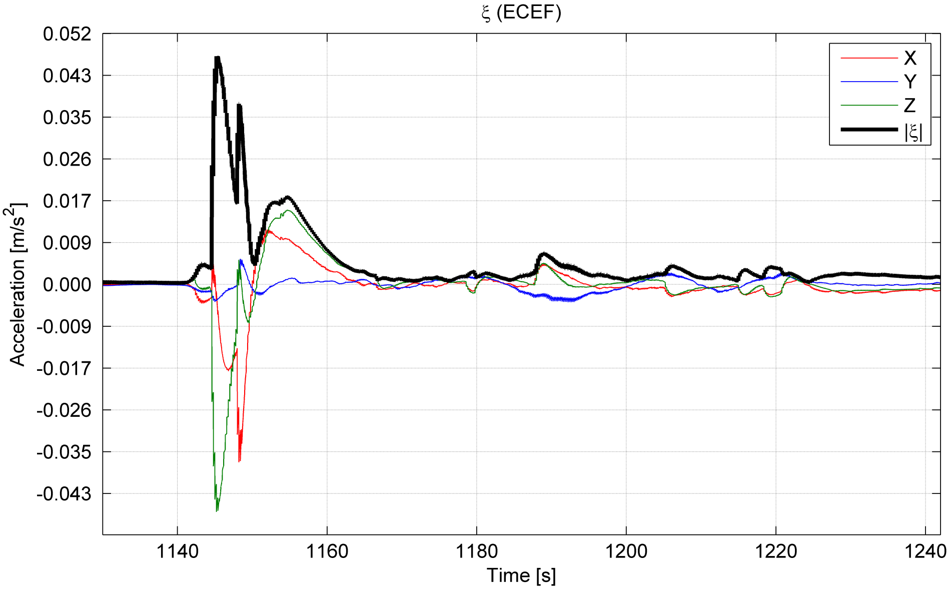


Figure 7.14: Case 1 (Offline): Main Observer Error (ξ)

7.2 Case 2: Loss of Gyrocompass

This section will consider the drift of the heading estimate obtained from the nonlinear observer, when the gyrocompass was unavailible for a certain time period. During the test the trolley was moved in a pattern as illustrated in Figure 7.15, starting and ending at the initial position. The gyrocompass aiding was turned off during the whole transit, i.e. between $t = 1530$ $[s]$ and $t = 1620$ $[s]$. This was the same time period used to plot the data. When the gyrocompass was absent, the observer completely relied on the integration of gyroscope measurements to provide heading estimates.

7.2.1 Results

The position estimates were satisfactory for both the online and offline results, under the absense of gyrocompass aiding. See Figure 7.15, 7.16, 7.17 and 7.18 for illustrations. The offline simulation performed better for reasons mentioned in the previous section. At $t = 1593$ $[s]$, the GNSS measurements experienced a jump in the position, similar to the previous case. The corresponding position errors can be seen in Figure 7.19 and 7.20.

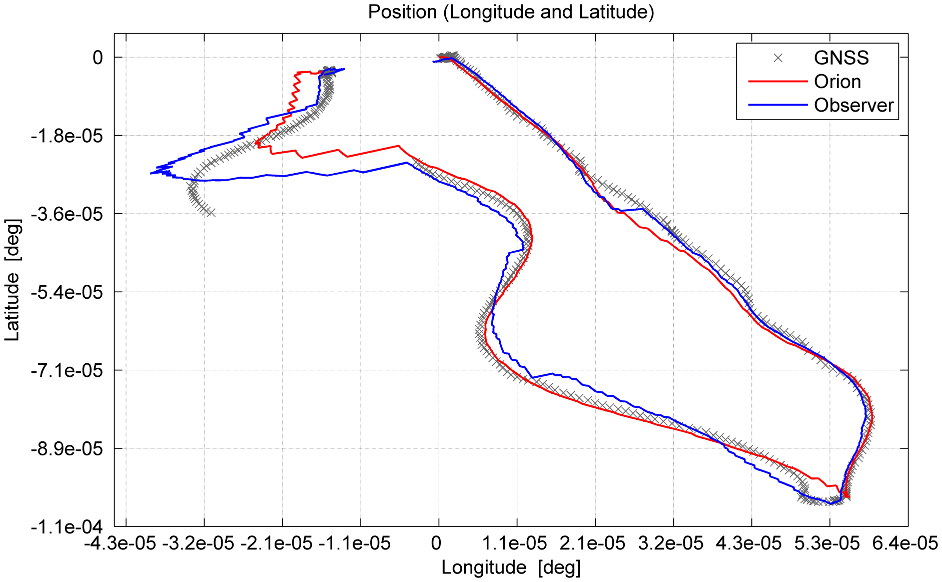


Figure 7.15: Case 2 (Online): Position (Latitude and Longitude)

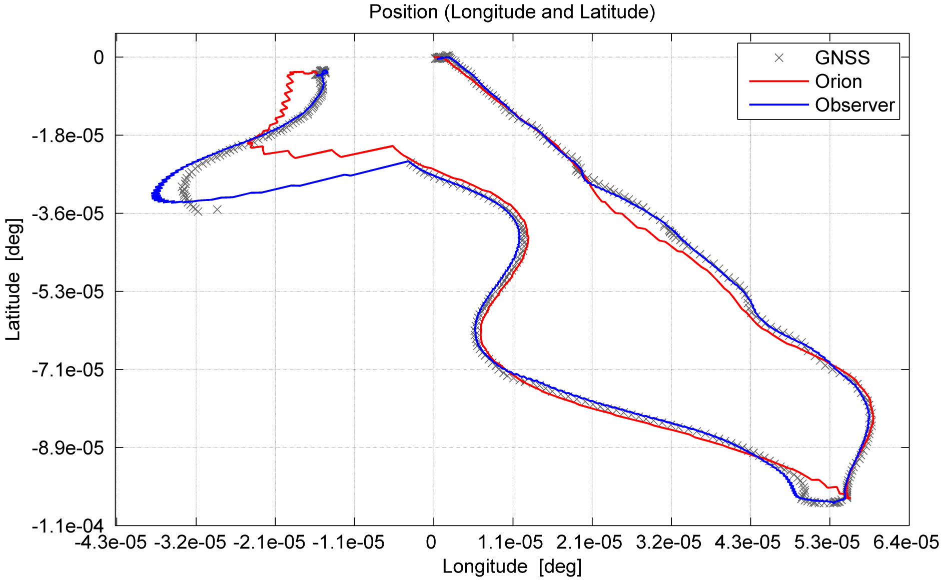


Figure 7.16: Case 2 (Offline): Position (Latitude and Longitude)

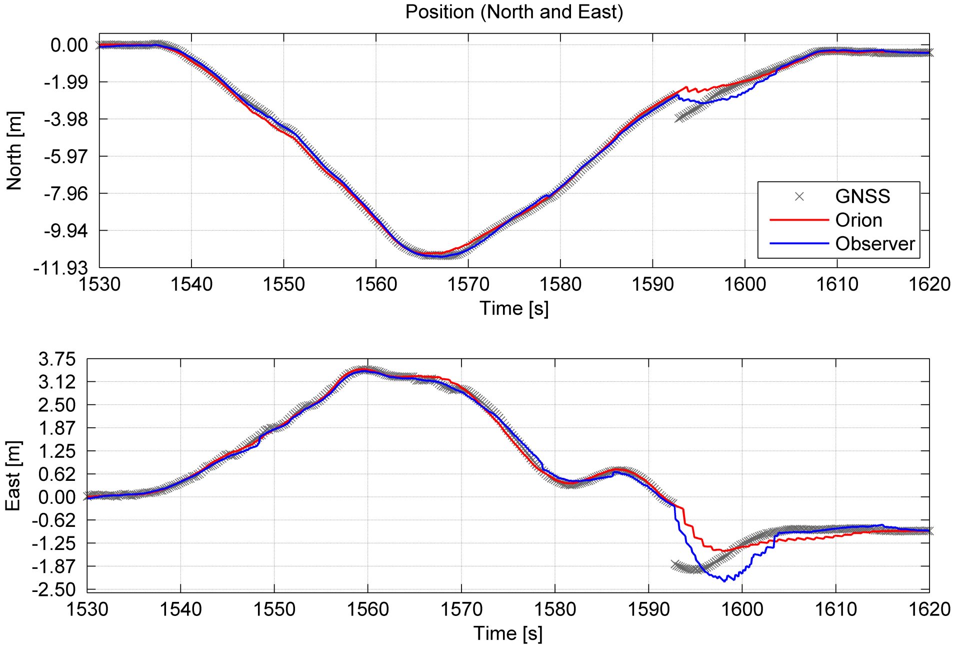


Figure 7.17: Case 2 (Online): Position (North and East)

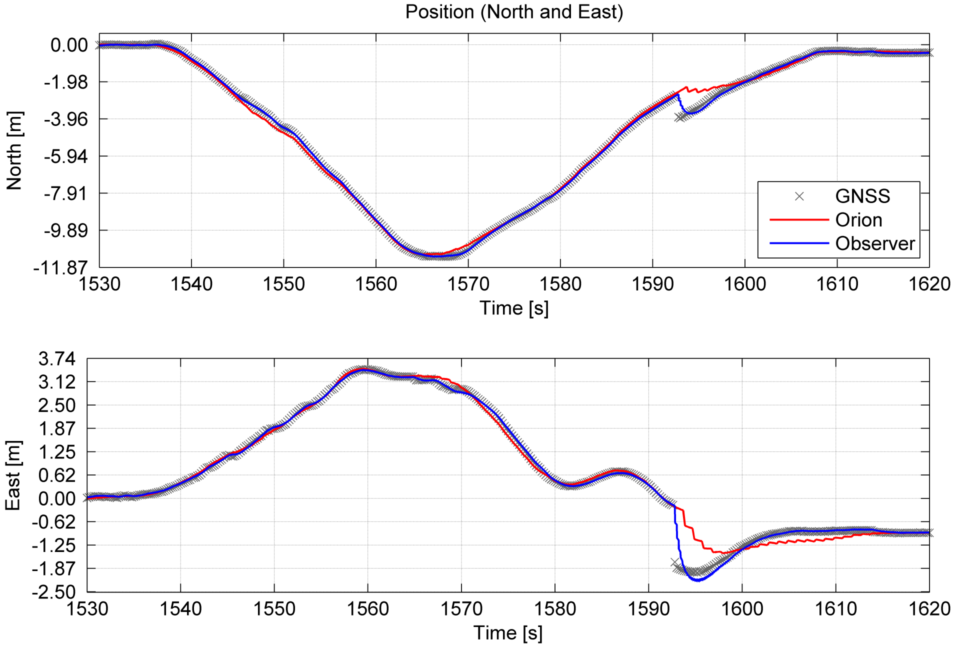


Figure 7.18: Case 2 (Offline): Position (North and East)

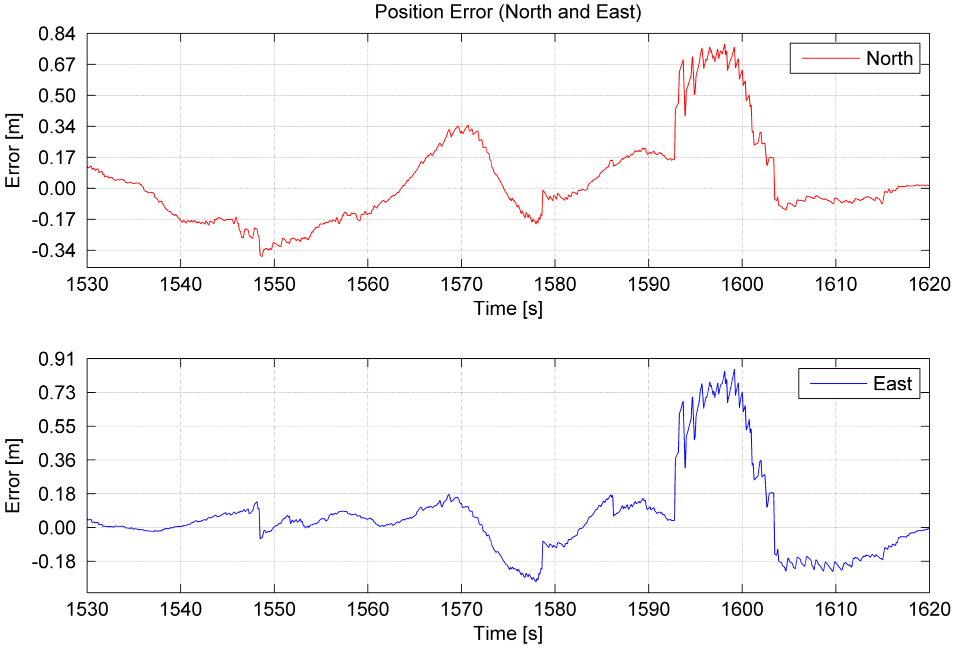


Figure 7.19: Case 2 (Online): Position Error (North and East)

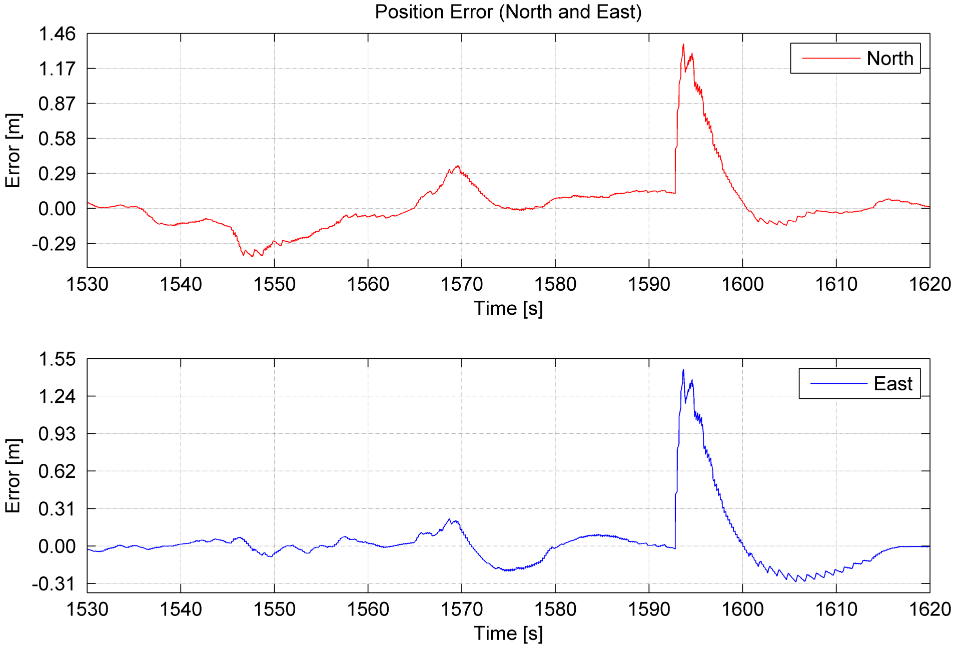


Figure 7.20: Case 2 (Offline): Position Error (North and East)

The heading estimate managed to keep track of the true heading with a low error during the whole transit, both for the online test and the offline simiulation. See Figure 7.21, 7.22, 7.23 and 7.24 for illustrations. The online test performed worse during the transit compared to the offline simulation, due to a lower k_1 gain and a higher position error. The heding error was during the transit under 3.36 [deg] for the online test. For comparison, the offline simulation yielded a transit error under 0.45 [deg]. This resulted in an improvement of 87 [%] for the offline simulation. Both cases ended up at approximately the same error at the end of the test, with a heading error under 0.30 [deg].

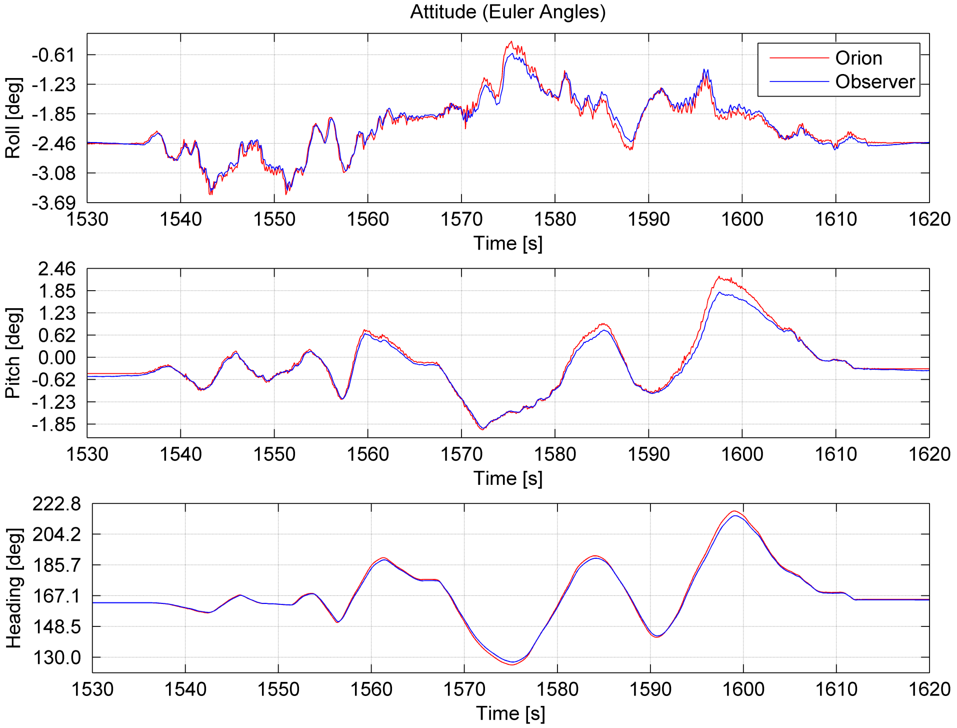


Figure 7.21: Case 2 (Online): Attitude (Euler Angles)

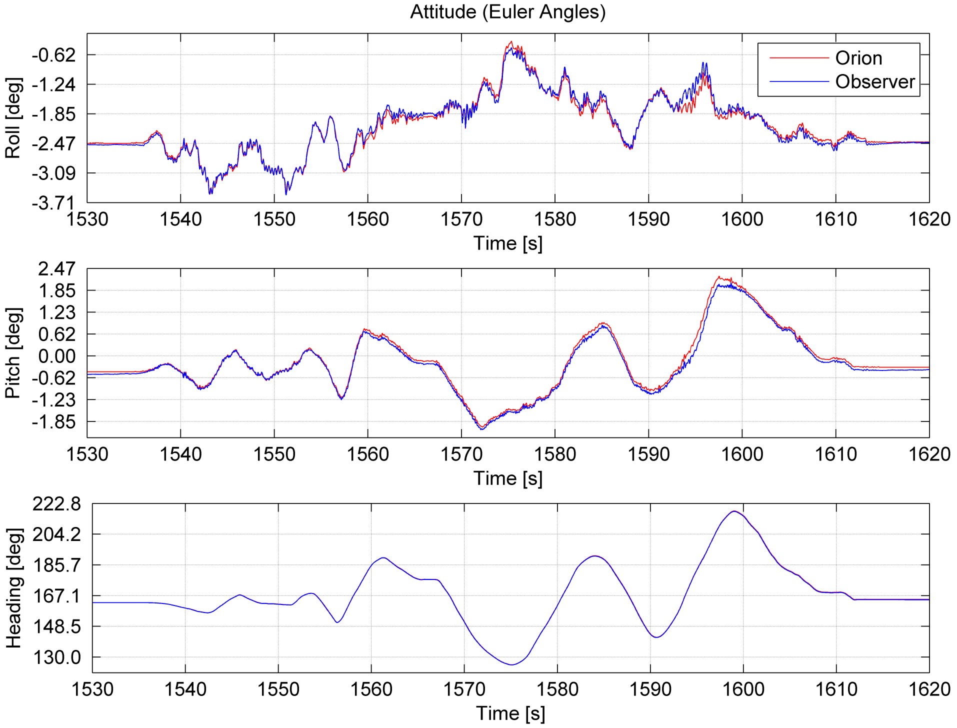


Figure 7.22: Case 2 (Offline): Attitude (Euler Angles)

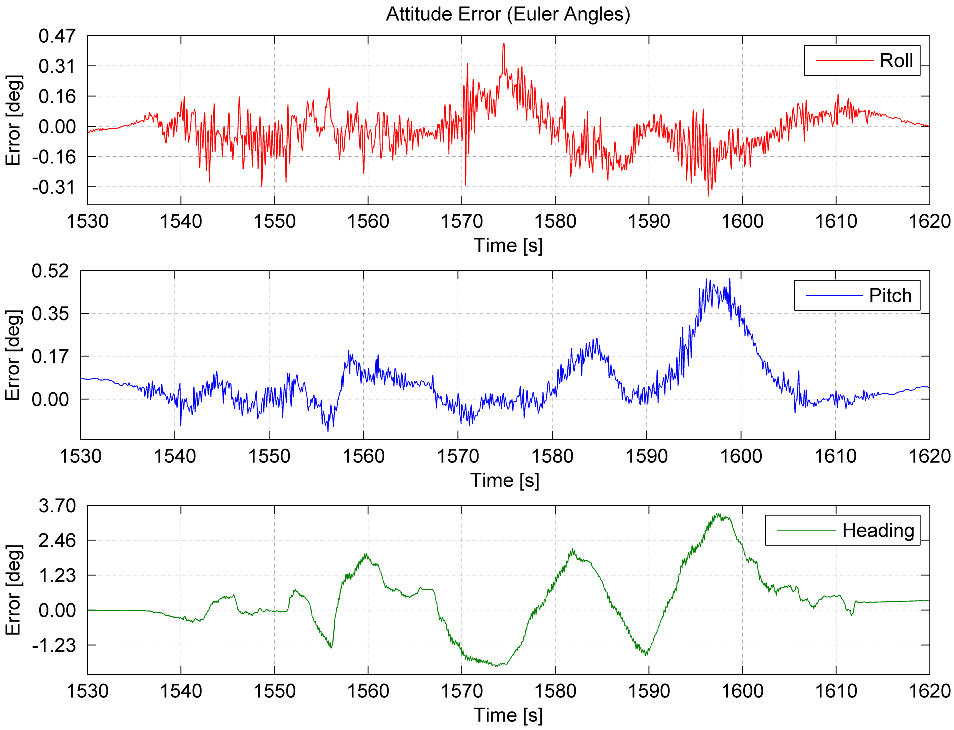


Figure 7.23: Case 2 (Online): Attitude Error (Euler Angles)

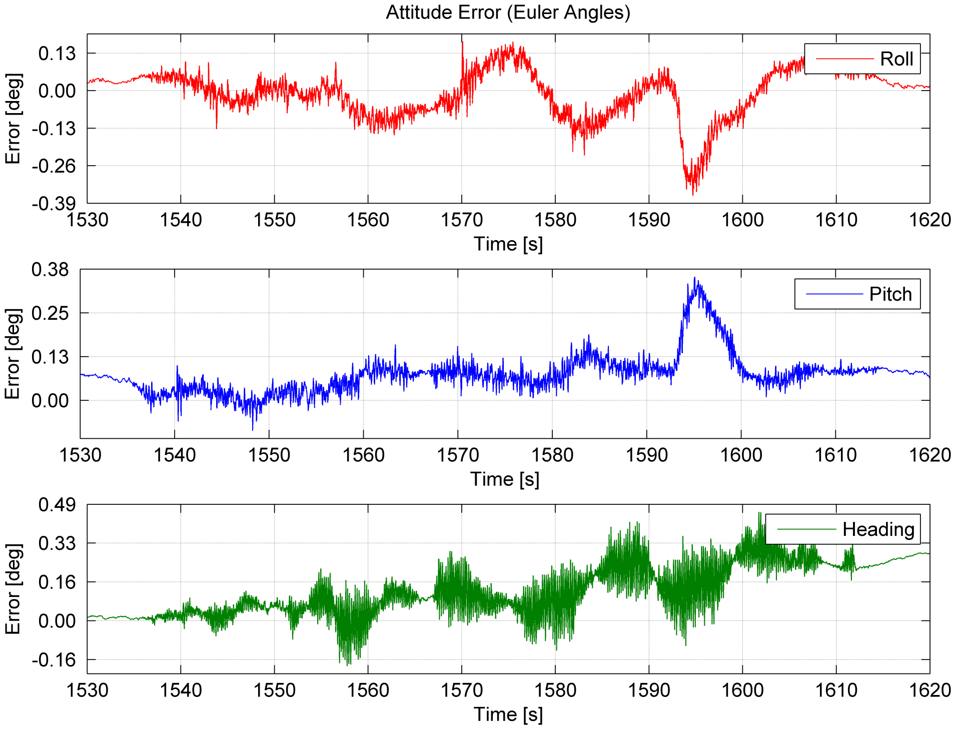


Figure 7.24: Case 2 (Offline): Attitude Error (Euler Angles)

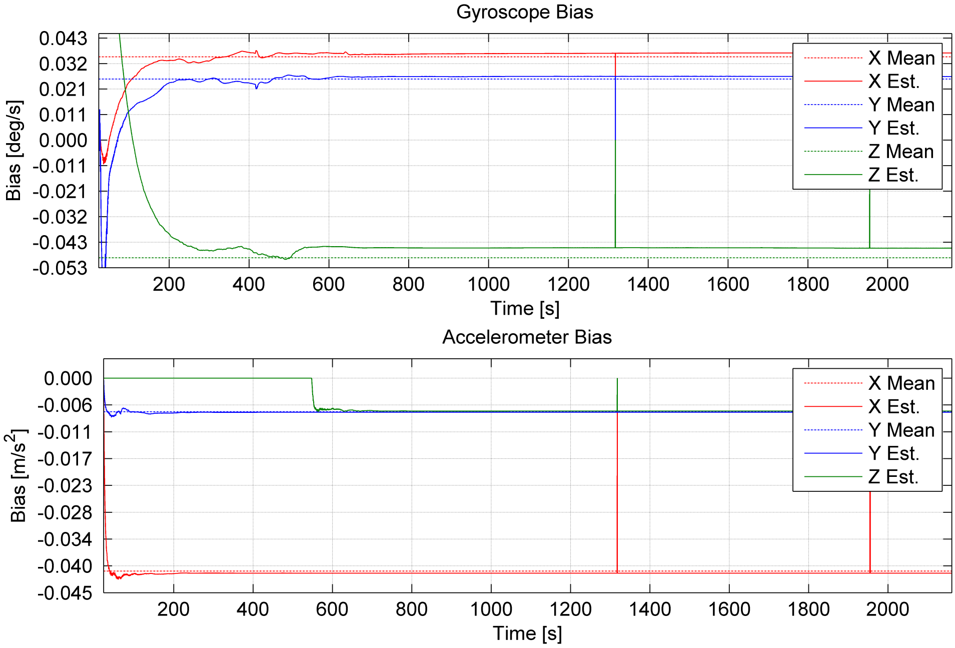


Figure 7.25: Case 2 (Online): Gyroscope Bias and Accelerometer Bias

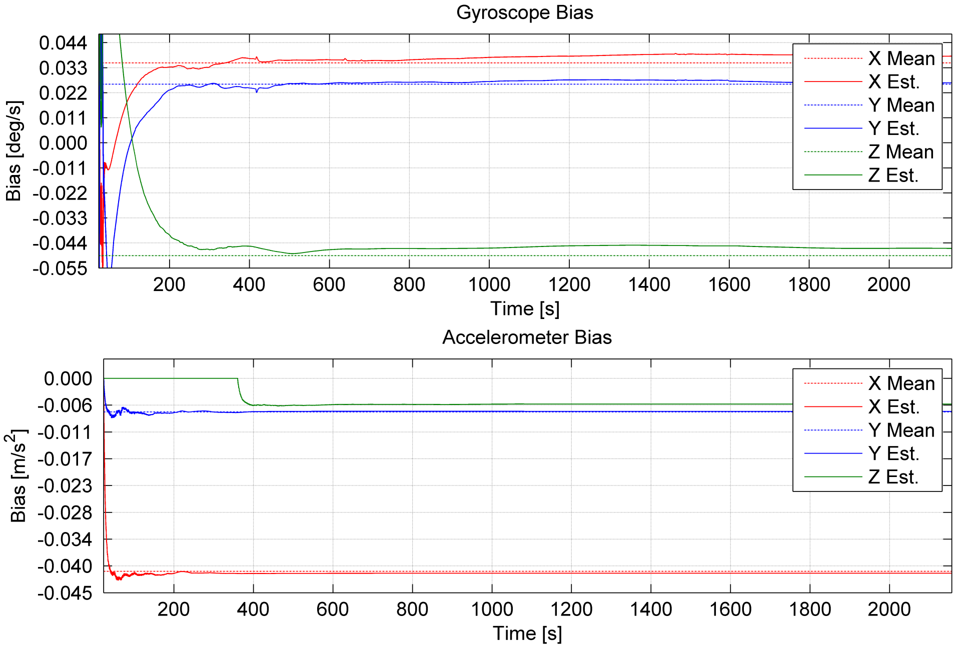


Figure 7.26: Case 2 (Offline): Gyroscope Bias and Accelerometer Bias

The position offset from the GNSS receiver at $t = 1593$ $[s]$ can be reflected in the main observer error $\boldsymbol{\xi}$, see Figure 7.27 and 7.28. The offline simulation acted more aggressively on the position measurement due to the higher position gain.

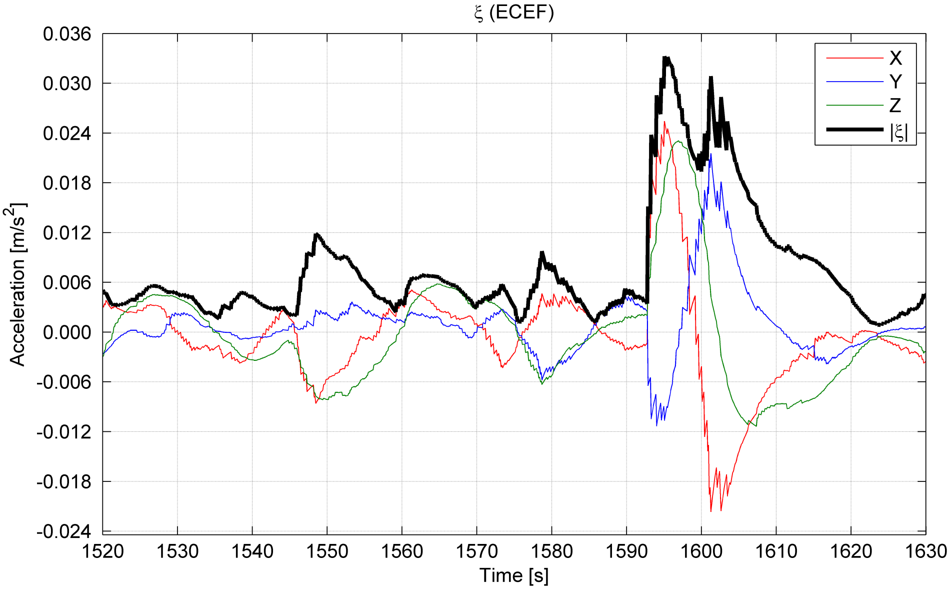


Figure 7.27: Case 2 (Online): Main Observer Error (ξ)

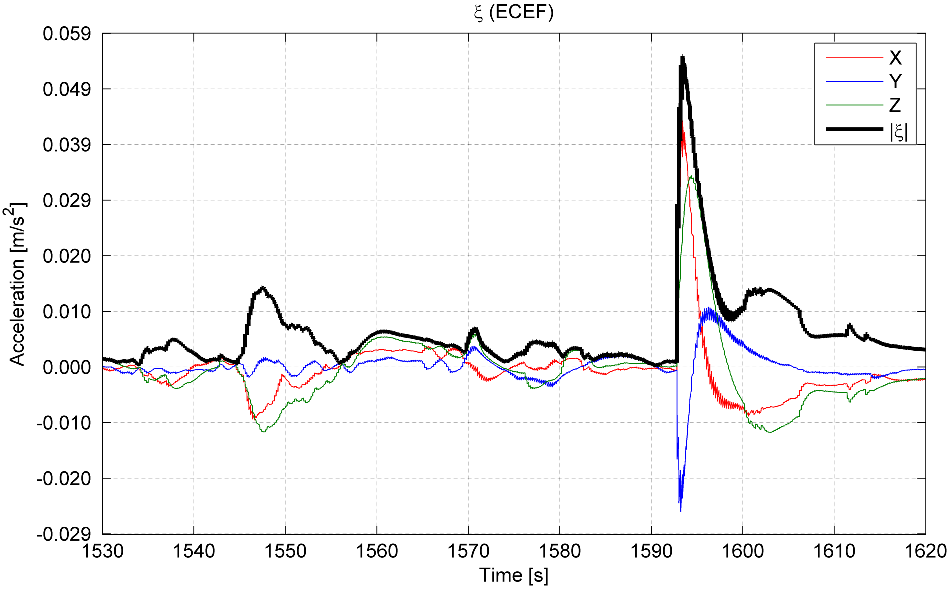


Figure 7.28: Case 2 (Offline): Main Observer Error (ξ)

7.3 Case 3: Loss of GNSS

This section studied the behaviour of the observer when the GNSS was absent under defined time periods, i.e. dead-reckoning. Three different tests were carried out:

- **Observer:** Under the first test only the nonlinear observer lost the GNSS aiding. The trolley was moved in a similar manner as the previous case studies.
- **Observer and Orion:** In the second test, both the nonlinear observer and the Orion lost the GNSS. The trolley was moved approximately 10 $[m]$ south, before returning back to the initial position.
- **Observer and Orion (stationary):** In the third and last test, the trolley was stationary and level during the whole period. This was carried out to see how much both systems drifted.

The green dots in the position plots denoted when the GNSS was absent, and when the aiding reappeared.

7.3.1 Results: Observer

Figure 7.29, 7.30, 7.31 and 7.32 illustrate the movement of the trolley during test test. The observer lost the GNSS aiding at the first green dot, while the Orion still was being aided. Furthermore, the GNSS measurements were observed to contain more noise than the previous tests, because of increased satellite interference. On the other hand, the IMU measurement noise wasn't as dominant as predicted in the simulation case studies in Chapter 6. This improved the dead-reckoning capabilities substantially.

The position estimates were observed to overshoot at some degree when the GNSS was absent. It seemed that the Orion also tried to overshoot its estimates, but was held back by the GNSS. For this reason it was hard to validate the quality of the estimates due to inconsistency in the GNSS measurements. When comparing the norm of the position errors for both the online and offline test, the offline results yielded an overall lower position error than the online test.

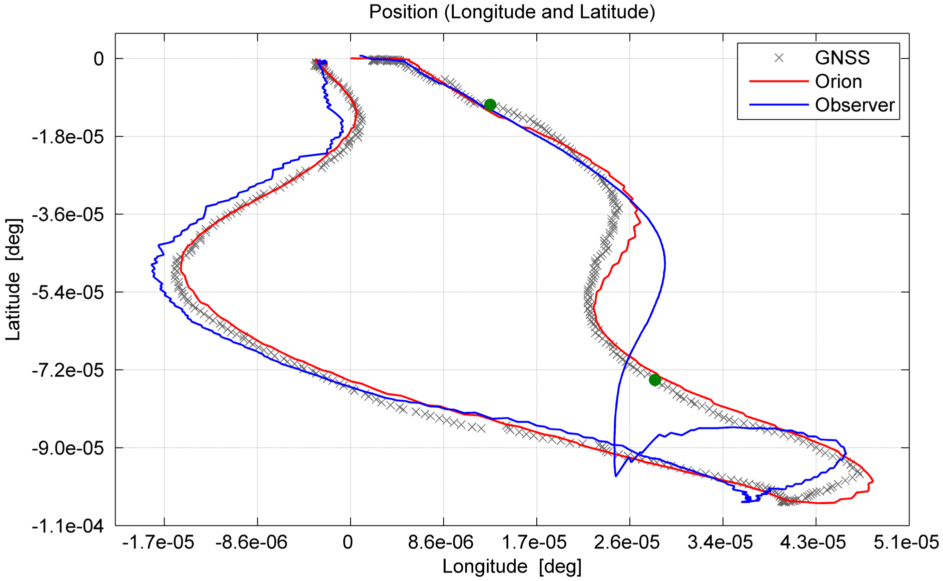


Figure 7.29: Case 3 (Online): Position (Latitude and Longitude)

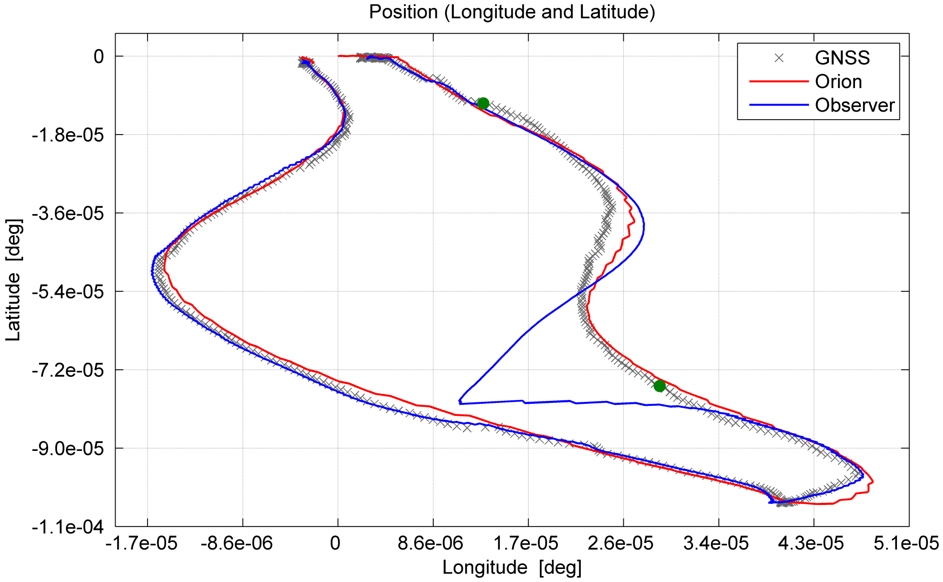


Figure 7.30: Case 3 (Offline): Position (Latitude and Longitude)

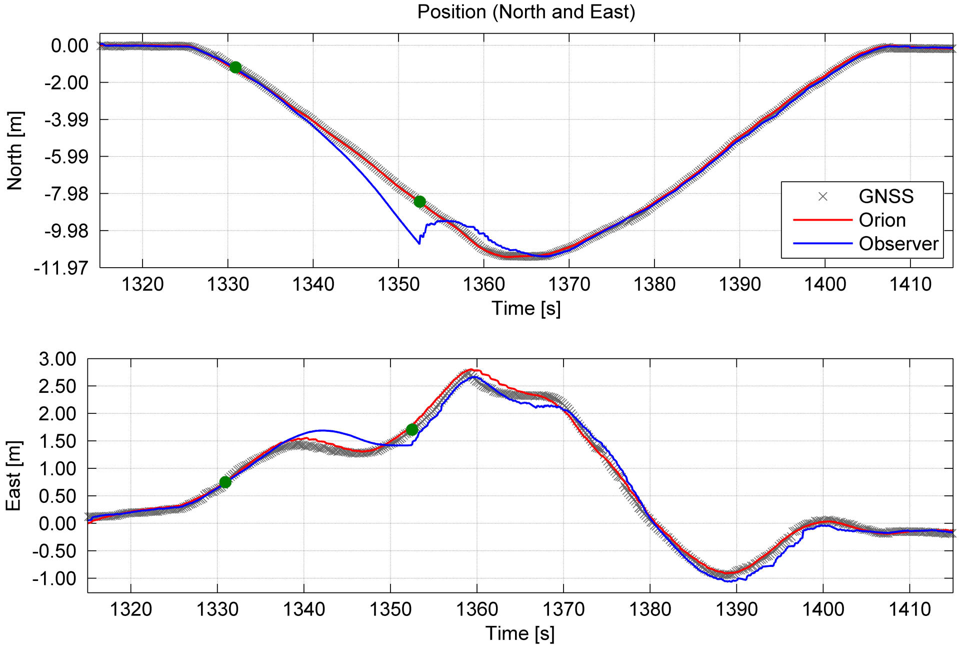


Figure 7.31: Case 3 (Online): Position (North and East)

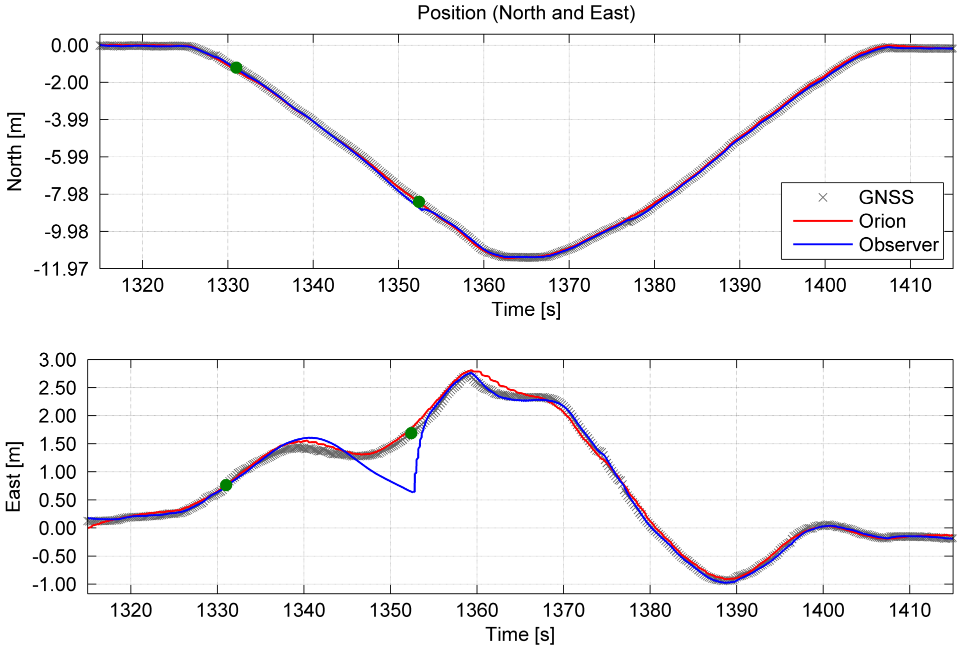


Figure 7.32: Case 3 (Offline): Position (North and East)

The position error can be seen in Figure 7.33 and 7.34, where the nonlinear observer in the online test drifted 2.21 $[m]$ north and 0.35 $[m]$ east, at the end of the dead-reckoning phase. In the offline simulation, the corresponding errors were 0.29 $[m]$ north and 1.15 [m] east. The cause of the degraded performance in the east direction, for the offline simulation, was due to the more aggressive GNSS gain. This caused the initial trajectory of the position estimate to be more sensitive to GNSS noise before the GNSS became absent.

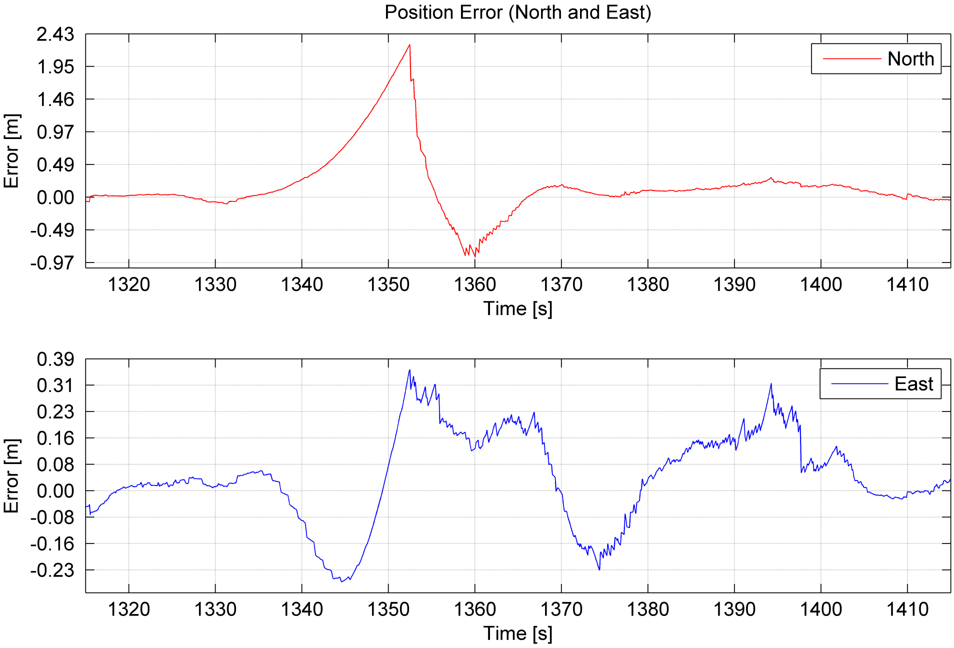


Figure 7.33: Case 3 (Online): Position Error (North and East)

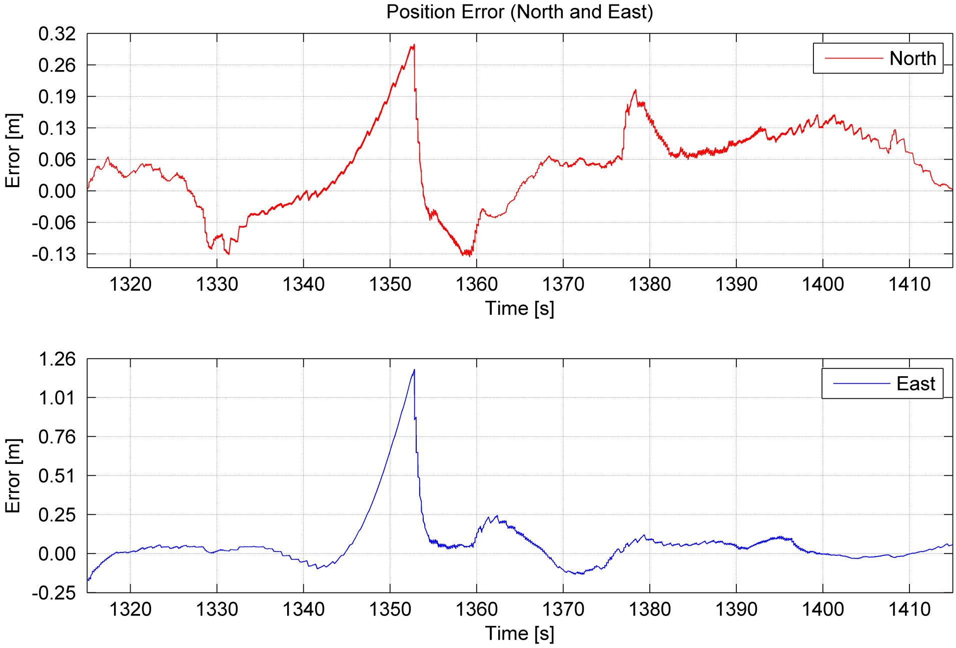


Figure 7.34: Case 3 (Offline): Position Error (North and East)

The absence of the GNSS caused the Euler angle estimate and the corresponding error to slightly drift, see Figure 7.35, 7.36, 7.37 and 7.38. The reasons for this deviation was explained in Section 7.1.

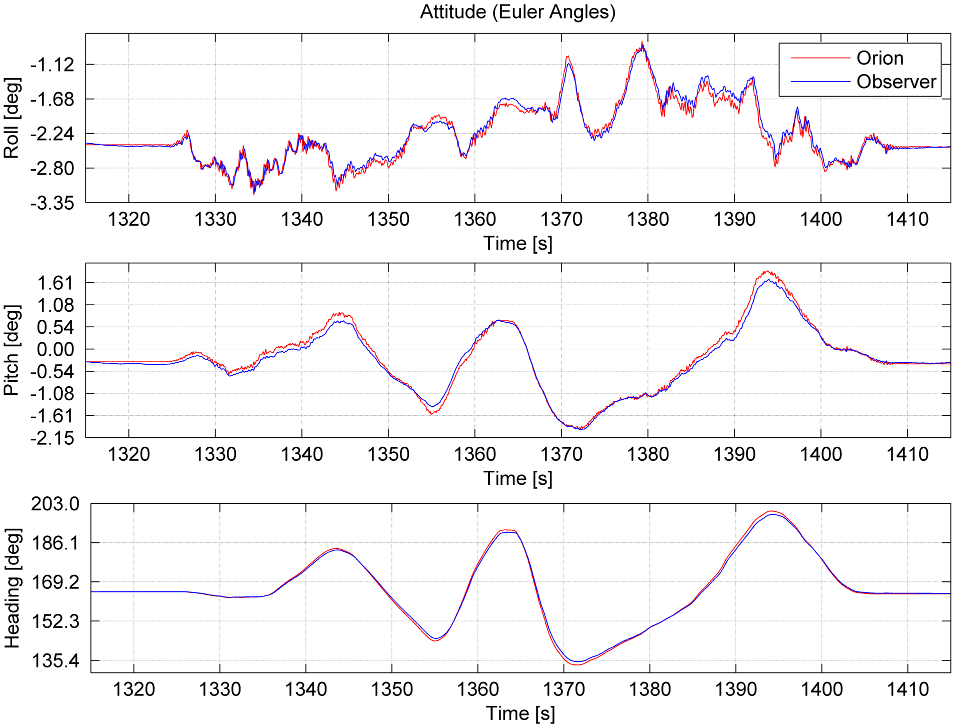


Figure 7.35: Case 3 (Online): Attitude (Euler Angles)

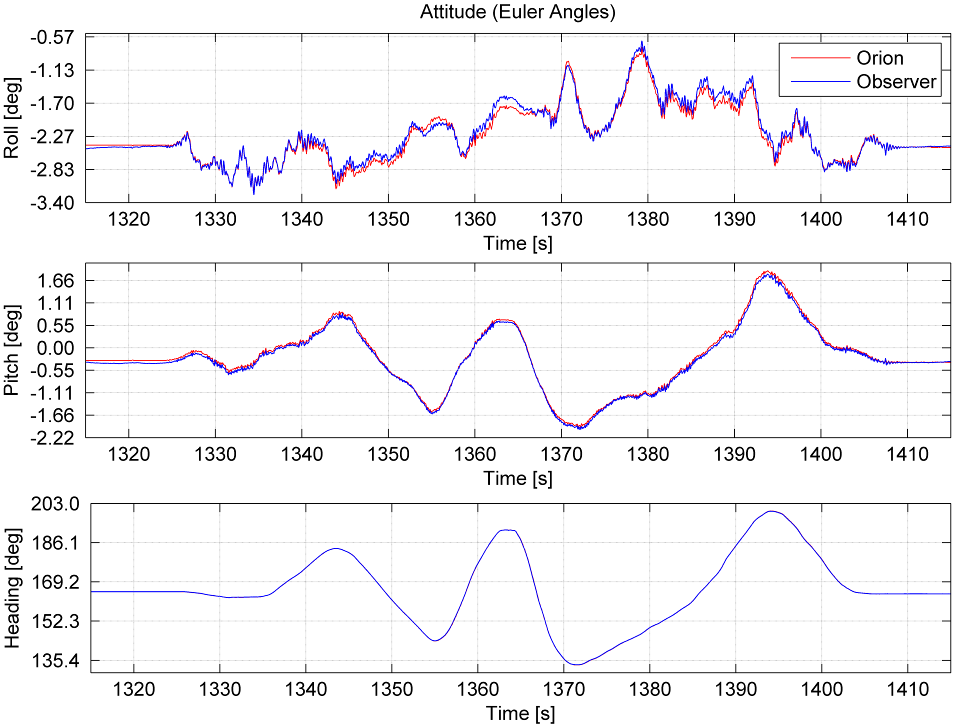


Figure 7.36: Case 3 (Offline): Attitude (Euler Angles)

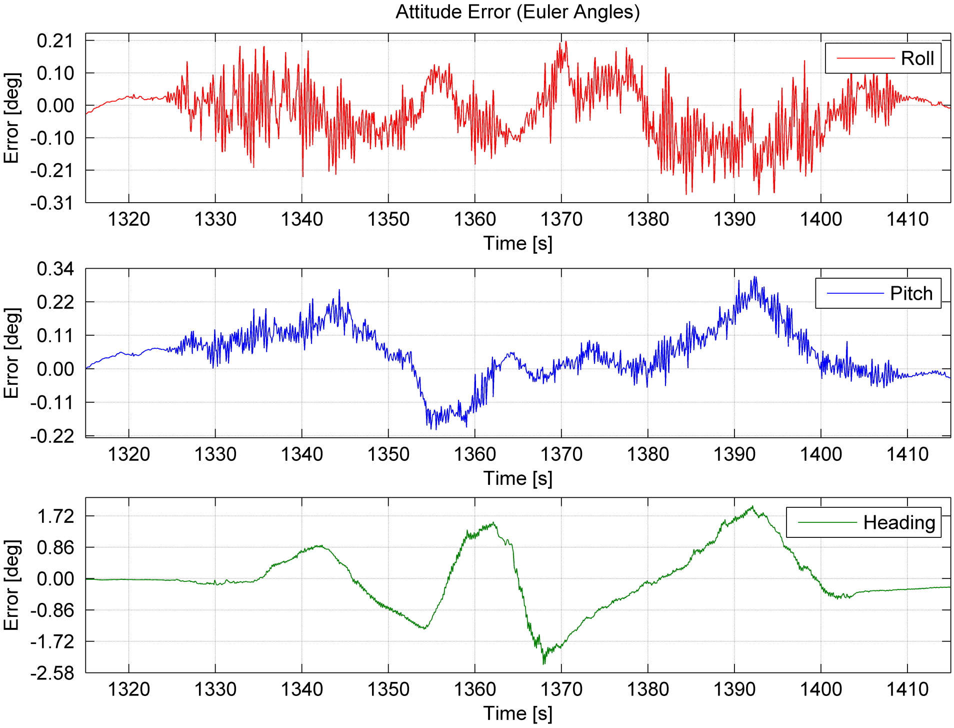


Figure 7.37: Case 3 (Online): Attitude Error (Euler Angles)

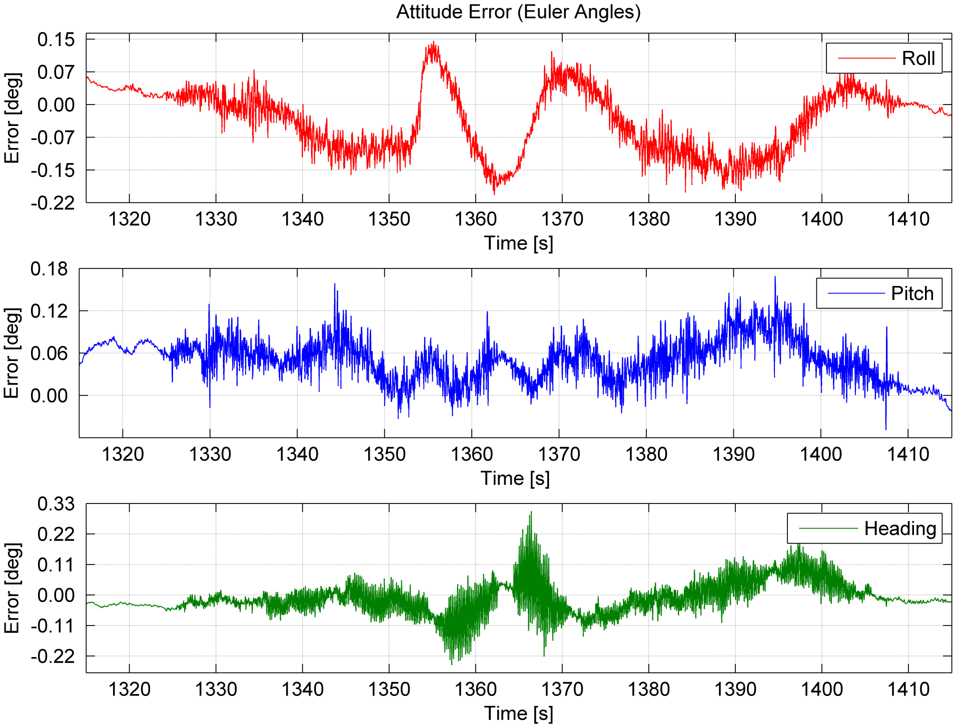


Figure 7.38: Case 3 (Offline): Attitude Error (Euler Angles)

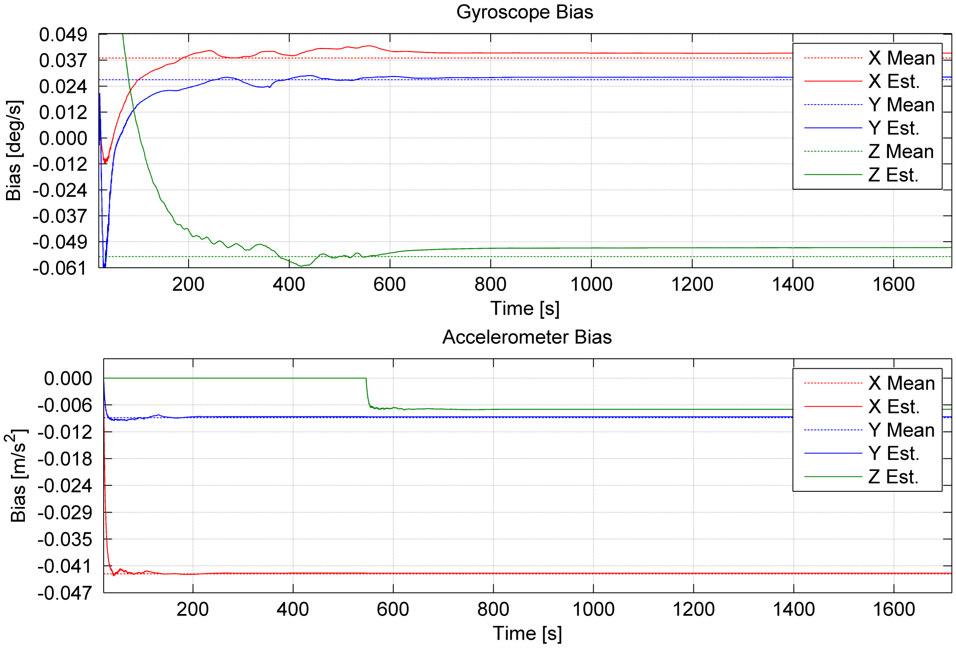


Figure 7.39: Case 3 (Online): Gyroscope Bias and Accelerometer Bias

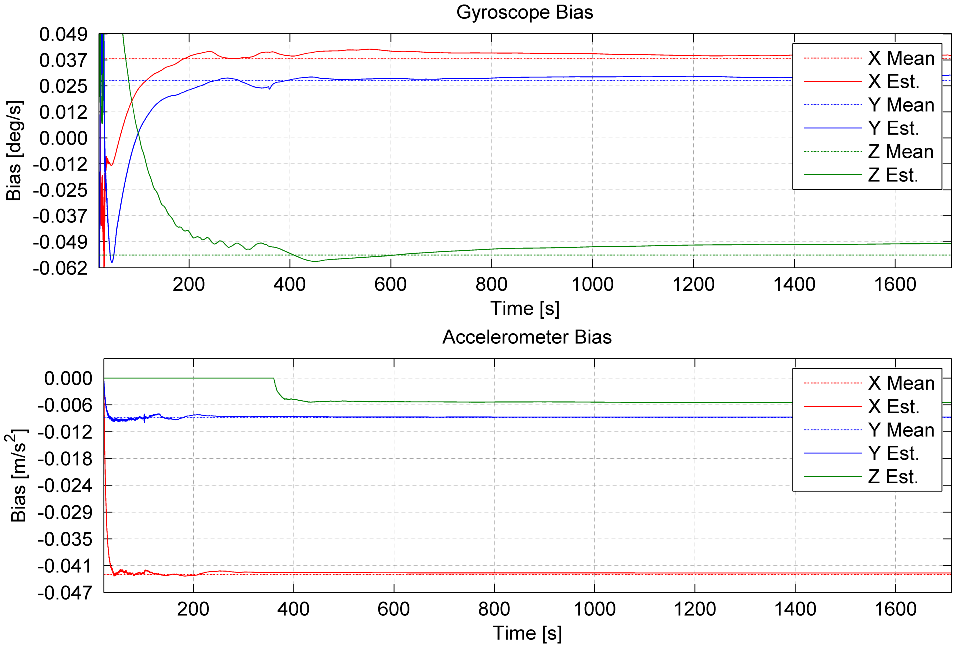


Figure 7.40: Case 3 (Offline): Gyroscope Bias and Accelerometer Bias

Right before the GNSS aiding was lost, the norm of the main observer error $||\boldsymbol{\xi}||$ in the offline simulation had a lower value than the online test. This can be seen in Figure 7.41

and 7.42. This was due to the more agressive GNSS gain in the offline simulation. It was also noticed that $||\boldsymbol{\xi}||$ was increasing right before the GNSS was lost. This was caused by an increase GNSS noise, and seemed to affect the estimates during the dead-reckoning phase.

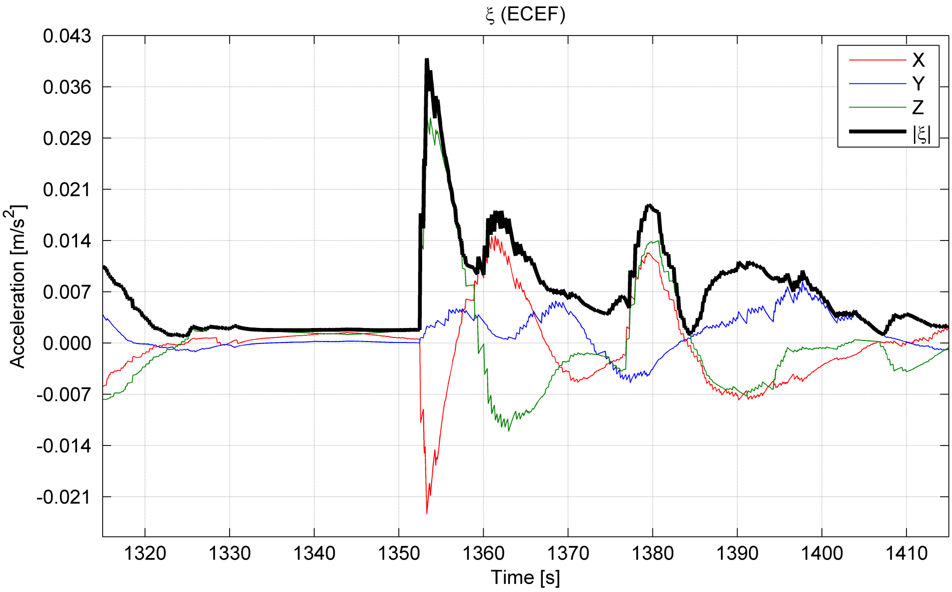


Figure 7.41: Case 3 (Online): Main Observer Error (ξ)

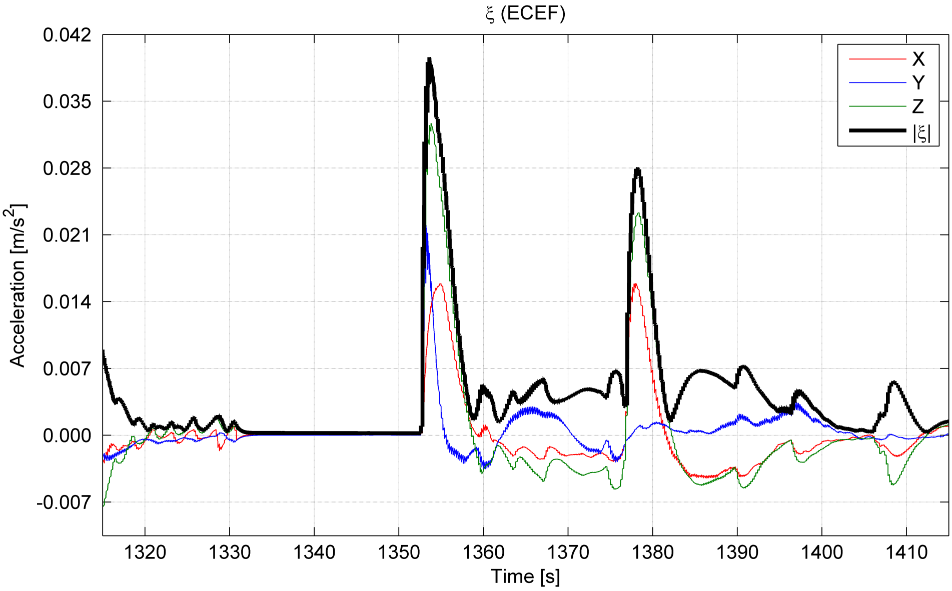


Figure 7.42: Case 3 (Offline): Main Observer Error (ξ)

Figure 7.43 shows the position estimates from another offline simulation, where the GNSS aiding was turned off 1 [s] earlier with a decreasing observer error $\boldsymbol{\xi}$, see Figure 7.44.

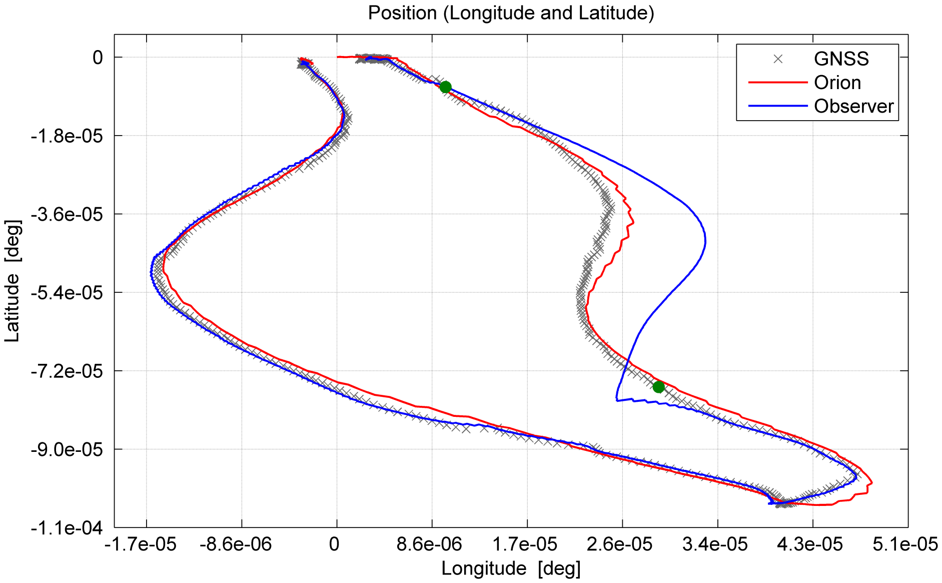


Figure 7.43: Case 3 (Offline): Position (Latitude and Longitude): Decreasing ξ

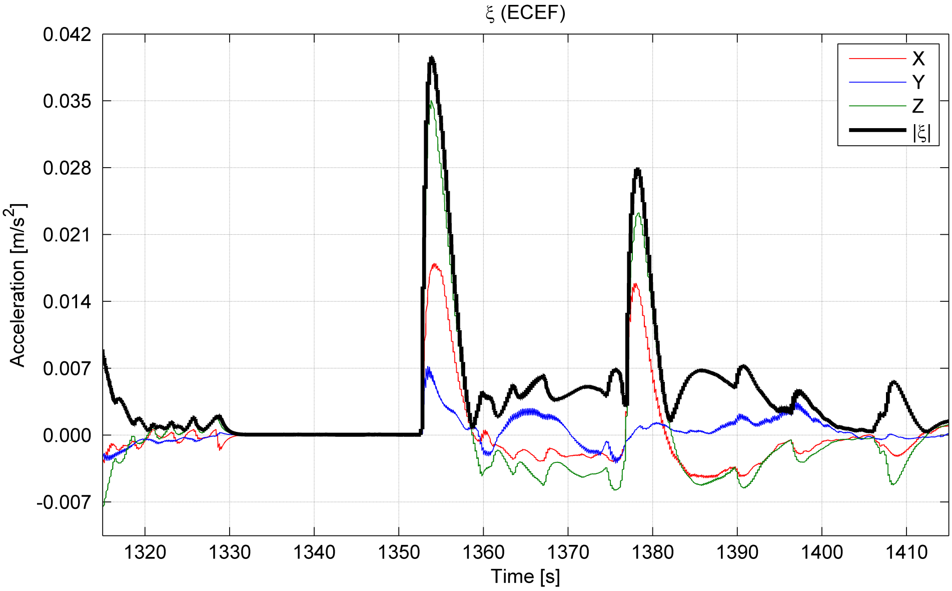


Figure 7.44: Case 3 (Offline): Main Observer Error (ξ): Decreasing ξ

7.3.2 Results: Observer and Orion

In this case, both navigation systems lost the GNSS aiding. The trolley was moved 10 [m] south, before it returned back to the initial position. This test was carried out to see the performance of both navigation systems during a straight line maneuver.

Figure 7.45, 7.46, 7.47 and 7.48 show the position estimates from both systems. Not surprisingly, the Orion performed better than the nonlinear observer with a drift of approximately 2 $[m]$ error in the east direction, and a negligible error to the north. The nonlinear observer provided the best overall position estimates in the offline simulation. The corresponding error was 5.5 $[m]$ in the north direction, compared to a -10 $[m]$ position error in the online test. The position drift to the east seemed to be the same for both the online test and the offline simulation, with approximately -25 $[m]$ drift. In this particular direction, the offline simulation performed slightly worse due to the more aggressive GNSS gain.

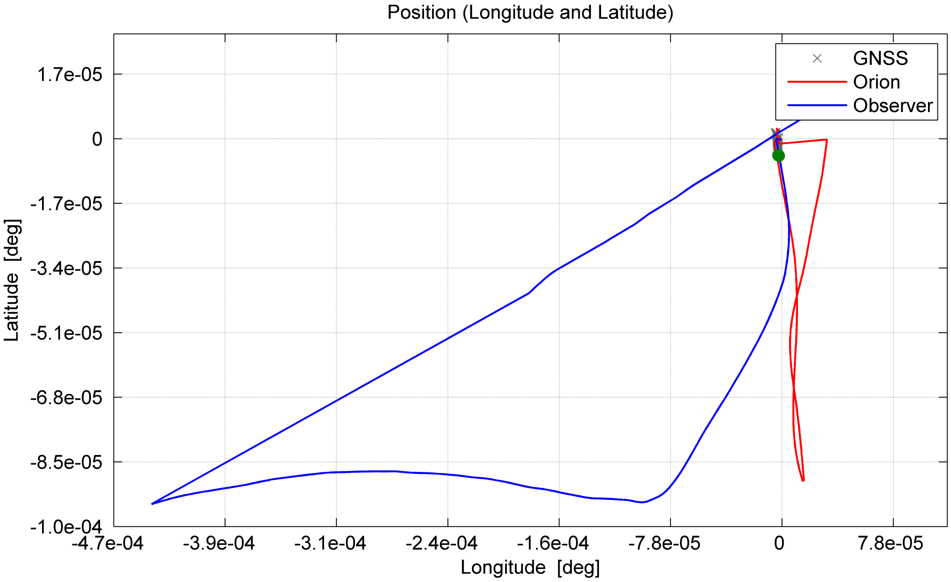


Figure 7.45: Case 3 (Online): Position (Latitude and Longitude)

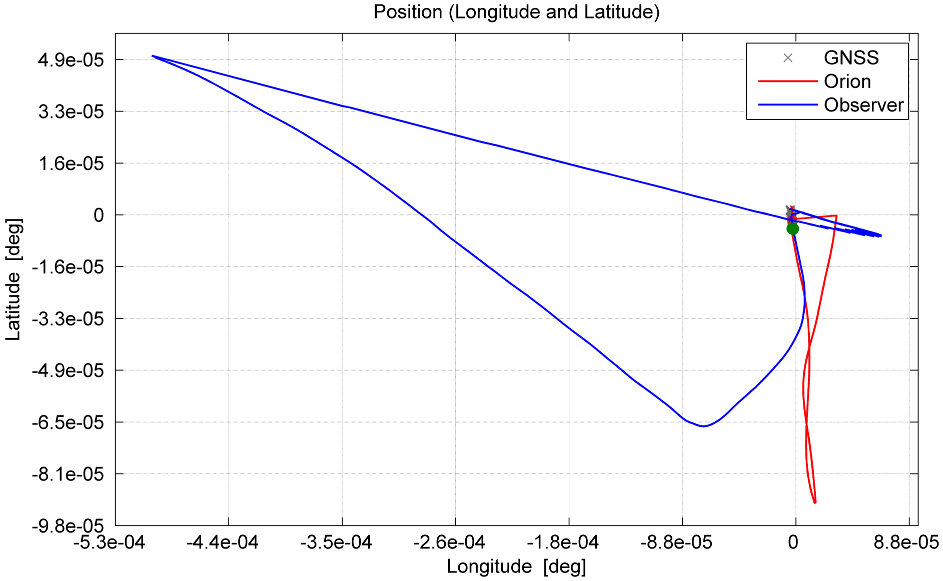


Figure 7.46: Case 3 (Offline): Position (Latitude and Longitude)

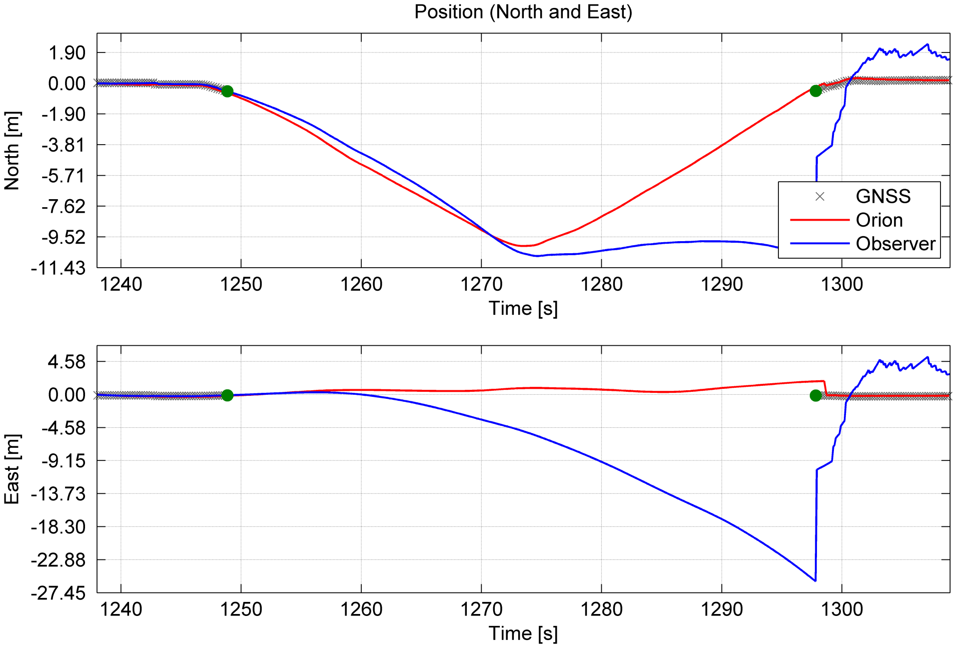


Figure 7.47: Case 3 (Online): Position (North and East)

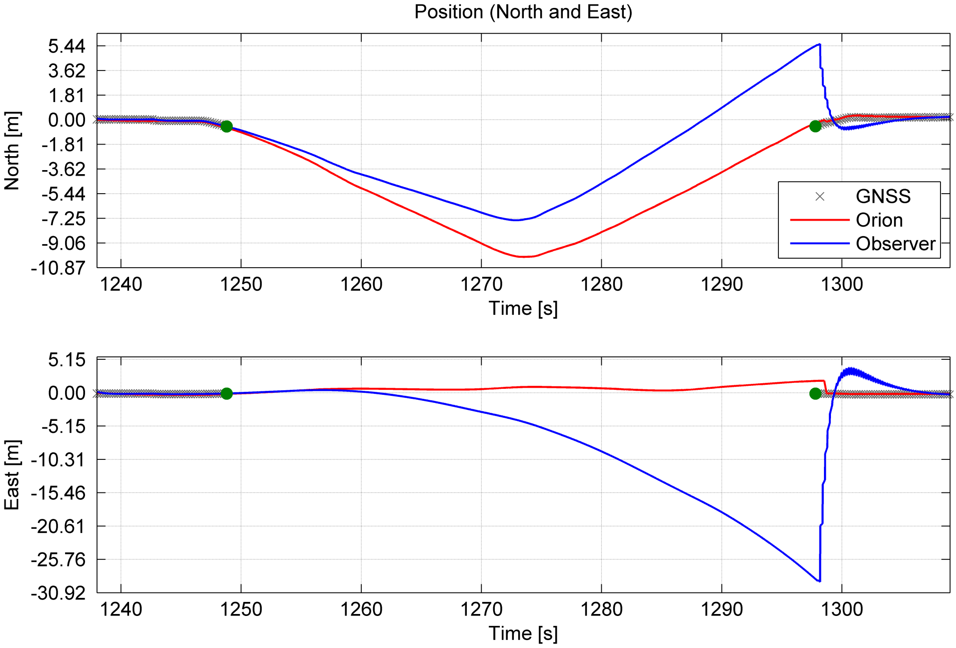


Figure 7.48: Case 3 (Offline): Position (North and East)

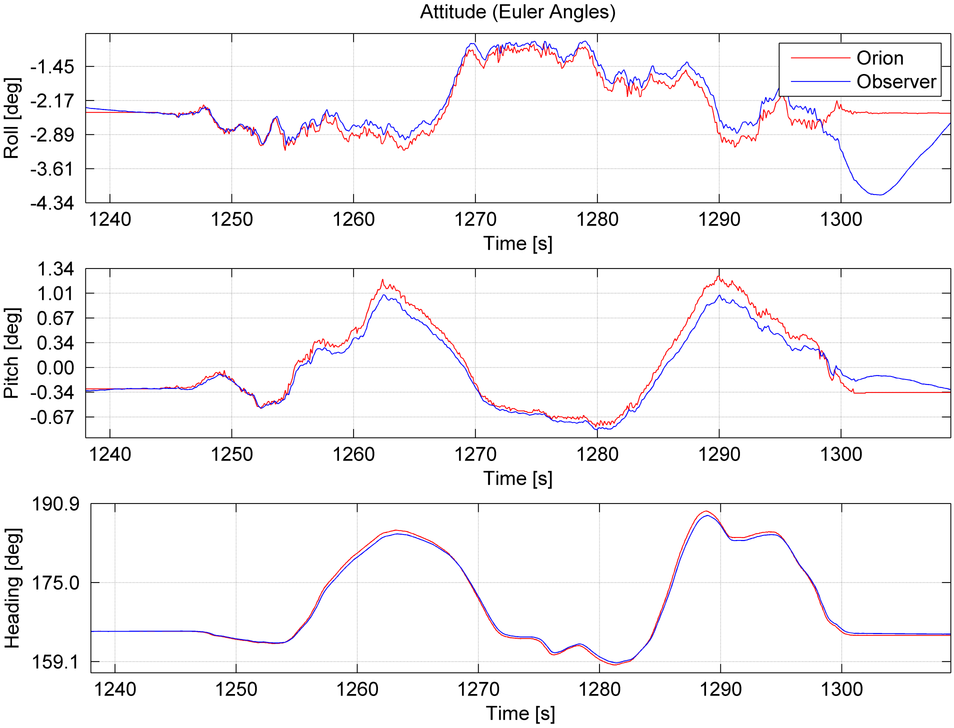


Figure 7.49: Case 3 (Online): Attitude (Euler Angles)

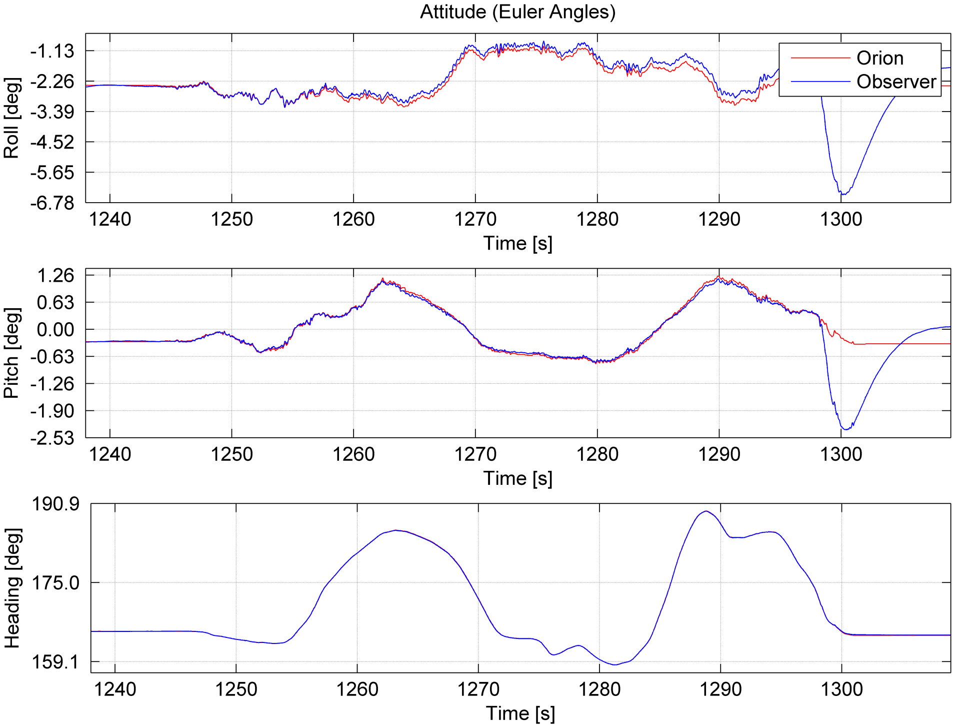


Figure 7.50: Case 3 (Offline): Attitude (Euler Angles)

The position estimate's impact on the Euler angle estimate, when the GNSS measurements returned, can be seen in Figure 7.51 and 7.52. This was because of the position error at the end of the dead-reckoning phase.

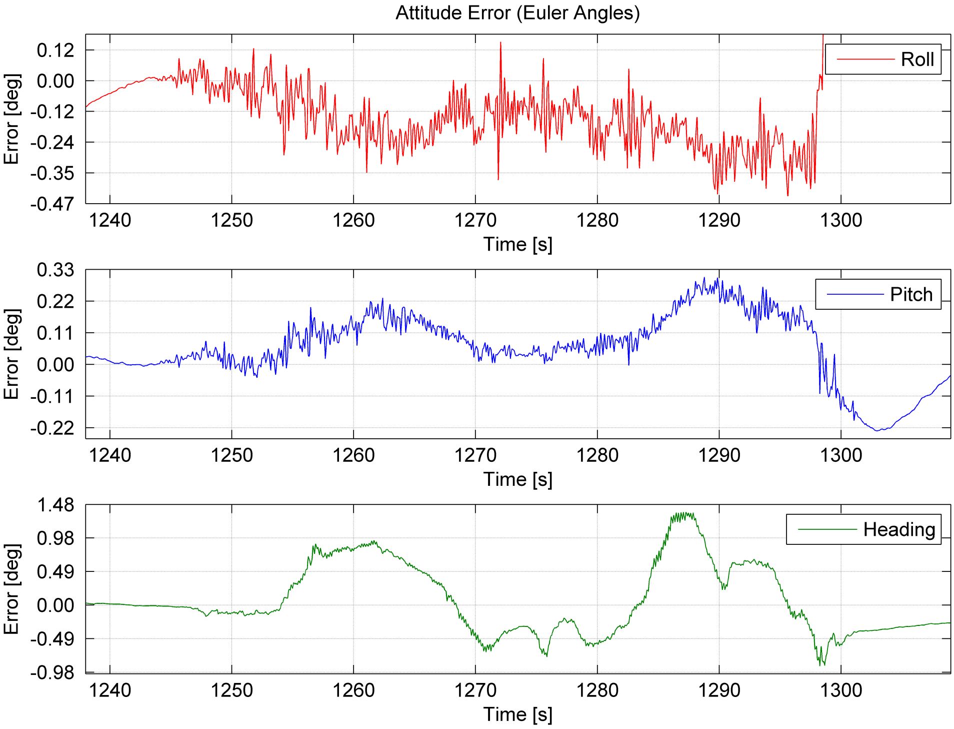


Figure 7.51: Case 3 (Online): Attitude Error (Euler Angles)

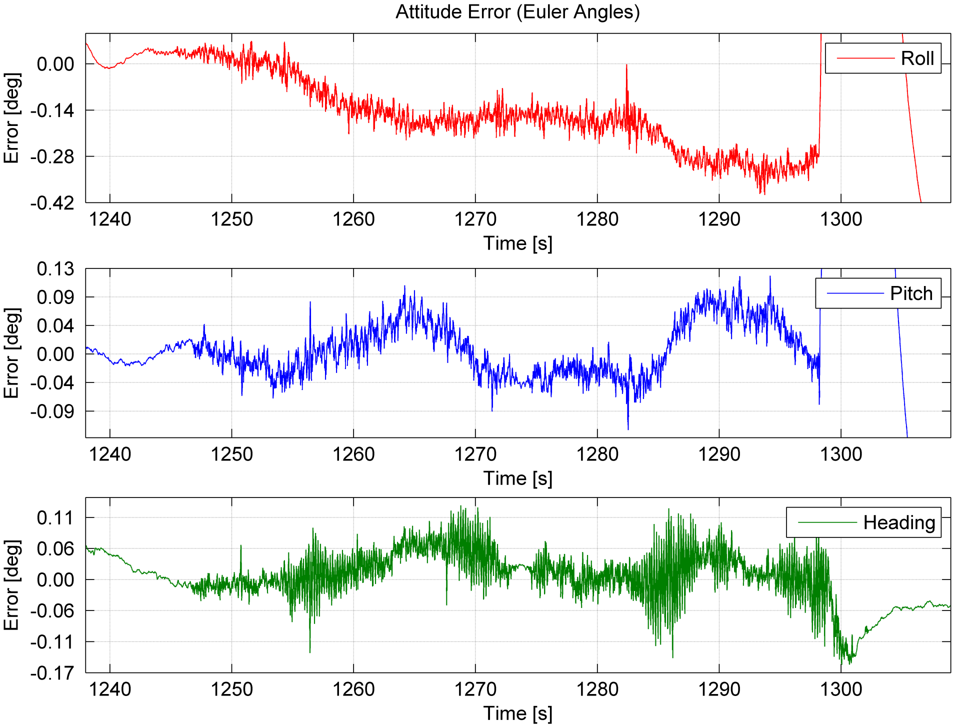


Figure 7.52: Case 3 (Offline): Attitude Error (Euler Angles)

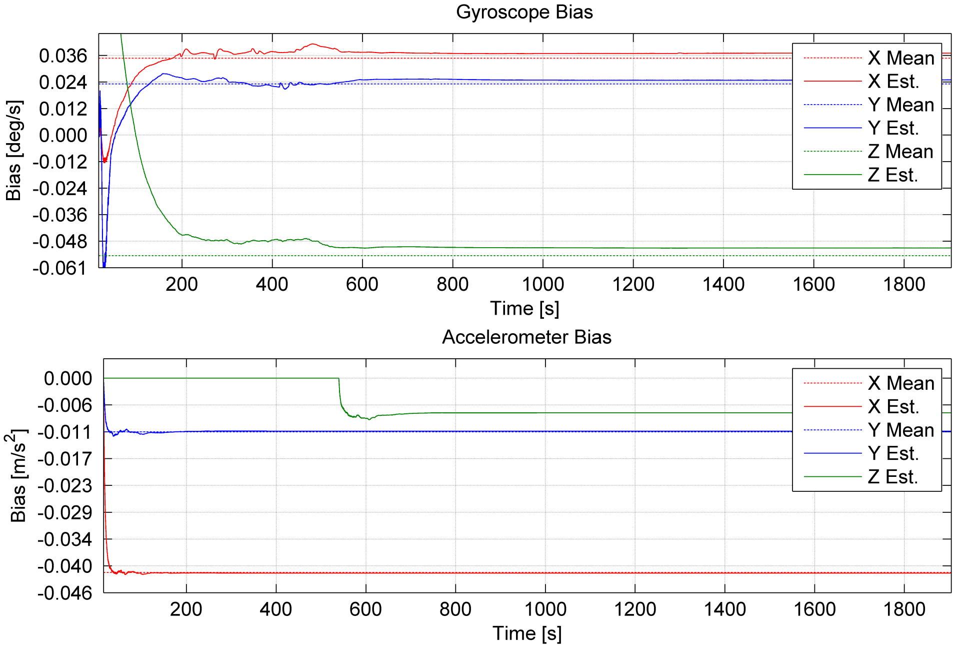


Figure 7.53: Case 3 (Online): Gyroscope Bias and Accelerometer Bias

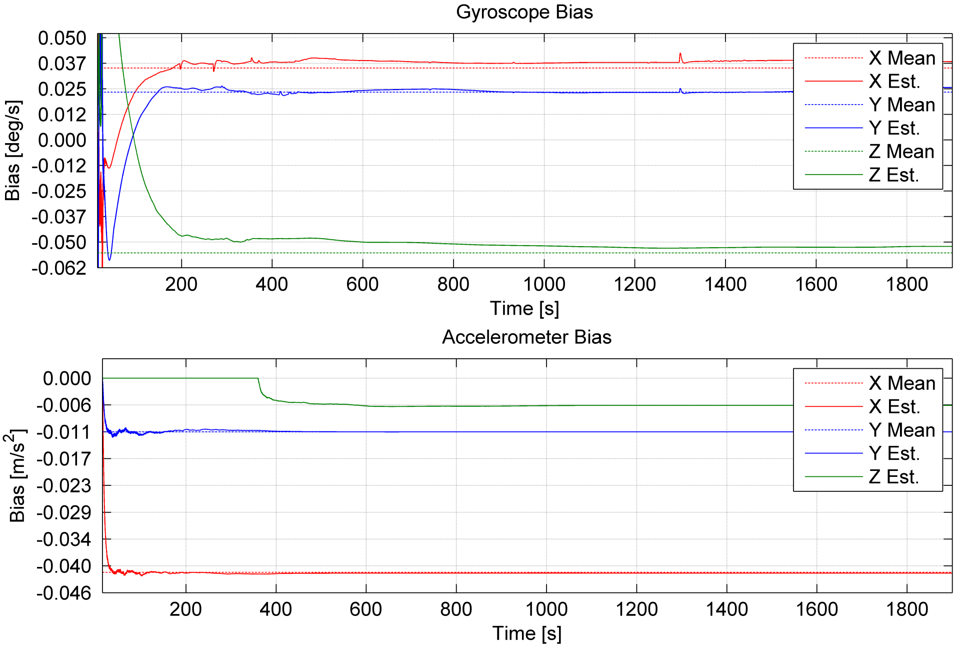


Figure 7.54: Case 3 (Offline): Gyroscope Bias and Accelerometer Bias

7.3.3 Results: Observer and Orion (Stationary)

The results in this section was based on the trolley being stationary, during the whole test. Since the trolley was stationary, it was possible to calculate a better estimate of the true gyroscope and accelerometer bias. This was done by taking the mean over several predefined time windows. The time window was defined to be 10 $[s]$ and resulted in a mean based on 5000 samples from each IMU sensor. This was sufficient to see the variation in the sensor biases, due to the random bias walk. See Figure 7.55 for an illustration of the estimated true biases, compared to the observer's estimates in the online test. The true sensor biases were denoted with the stippled lines. Between $t = 1210$ $[s]$ and $t = 1340$ $[s]$ the GNSS aiding was turned off for both navigation systems.

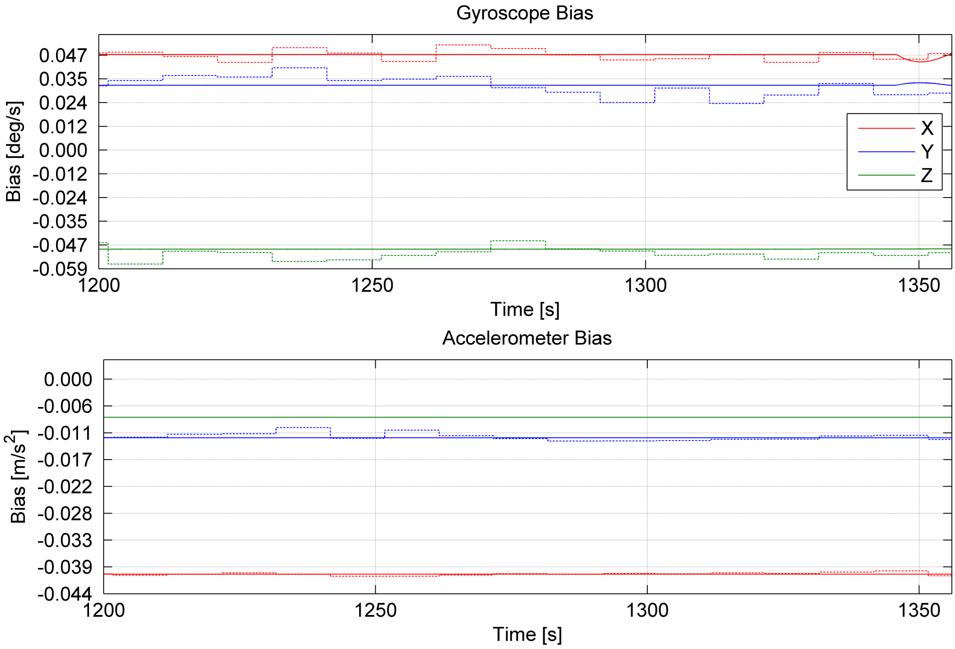


Figure 7.55: Case 3 (Online): Gyroscope Bias and Accelerometer Bias

From Figure 7.55 it was clear that the variation in the sensor bias were significant, especially for the gyroscope bias. It was discovered that when the GNSS aiding was gone, the first term in the attitude observer's injection term became zero, or very small, see (7.1). This was a result of not having feedback from the GNSS for correction of $\hat{\mathbf{f}}^e$.

$$\hat{\boldsymbol{\sigma}} = k_1 \underline{\mathbf{f}}^b \times \mathbf{R}_b^e(\hat{\mathbf{q}}_b^e)^\top \underline{\hat{\mathbf{f}}}^e + k_2 \mathbf{I}_{gyc} \underline{\mathbf{c}}^b \times \mathbf{R}_b^e(\hat{\mathbf{q}}_b^e)^\top \underline{\mathbf{c}}^e \tag{7.1}$$

This resulted in the gyroscope bias not being correctly estimated when the GNSS aiding was absent, for the x- and y-axis. This was also the case for the z-axis when the gyrocompass was turned off.

To solve this problem, the estimated specific force in the injection term got replaced with the gravity vector from the J2 gravity model, i.e $\hat{\mathbf{f}}^e = -\mathbf{g}^e(\hat{\mathbf{p}}^e)$. This could be realized since the trolley was stationary with zero non-gravitational accelerations. The gyroscope bias gain k_I was increased such that the estimated biases would follow the true gyroscope biases more aggressively. See 7.56 for the improved gyroscope bias estimates.

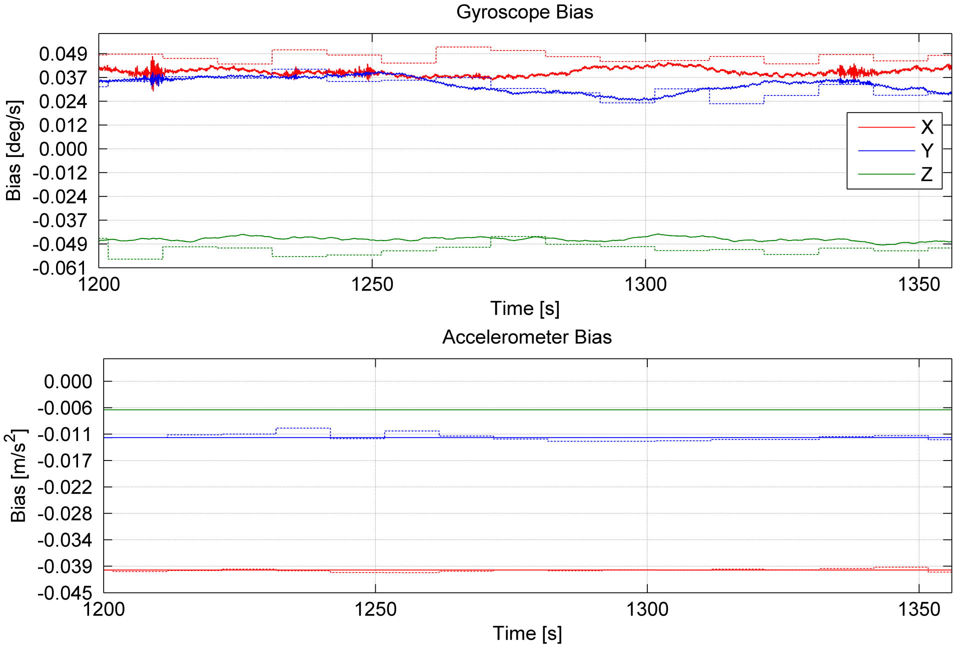


Figure 7.56: Case 3 (Offline): Gyroscope Bias and Accelerometer Bias

Figure 7.57 and 7.58 illustrate the position drift when the trolley was stationary for the online and offline tests, respectively. The online test utilized the estimated specific force in the attitude observer's injection term, whereas the offline simulation utilized the gravity vector.

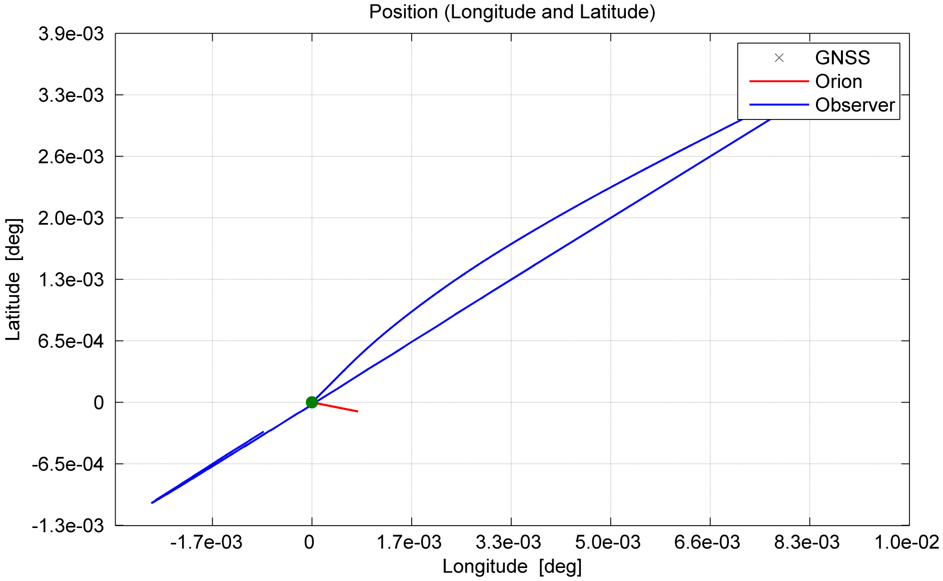


Figure 7.57: Case 3 (Online): Position (Latitude and Longitude)

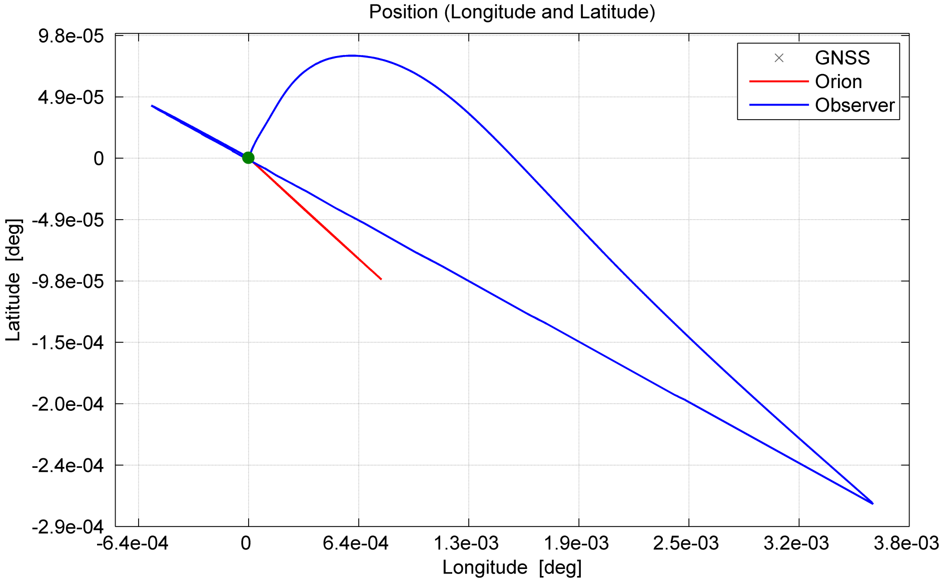


Figure 7.58: Case 3 (Offline): Position (Latitude and Longitude)

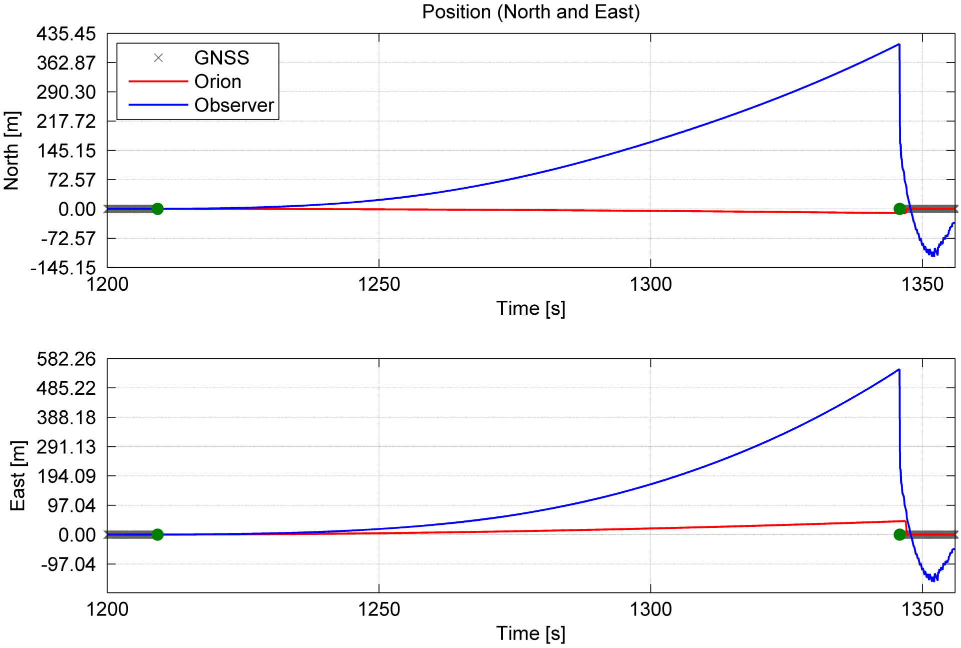


Figure 7.59: Case 3 (Online): Position (North and East)

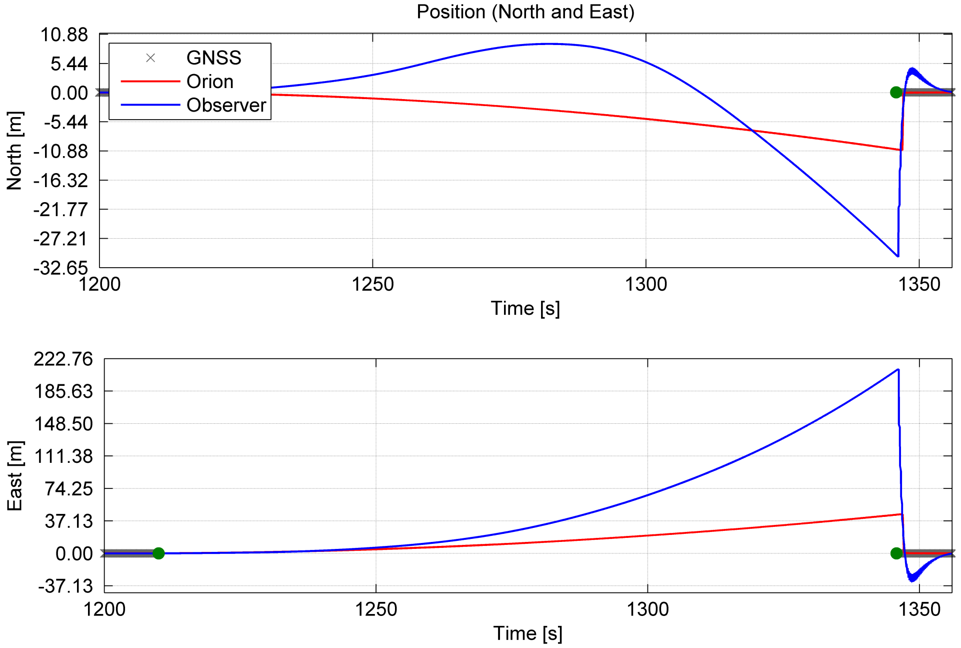


Figure 7.60: Case 3 (Offline): Position (North and East)

The position drift of the Orion reference system can be illustrated in Figure 7.61, where the drift of the nonlinear observer in the experimental test can be viewed in Figure 7.62. The drift in the improved offline simulation can be viewed in Figure 7.63.

The reference system drifted 10.3 $[m]$ north and -42.8 $[m]$ east, where the nonlinear observer drifted -395.9 $[m]$ north and -529.3 $[m]$ east in the online test. By replacing the estimated specific force with the gravity, the offline simulations yielded 29.7 $[m]$ drift to the north and -202.5 $[m]$ to the east. This improved the estimates 92.5 [%] in the north direction and 61.7 [%] in the east direction. Compared to the Orion, the nonlinear observer performed 2.9 and 4.7 times worse in the north and east axes, respectively.

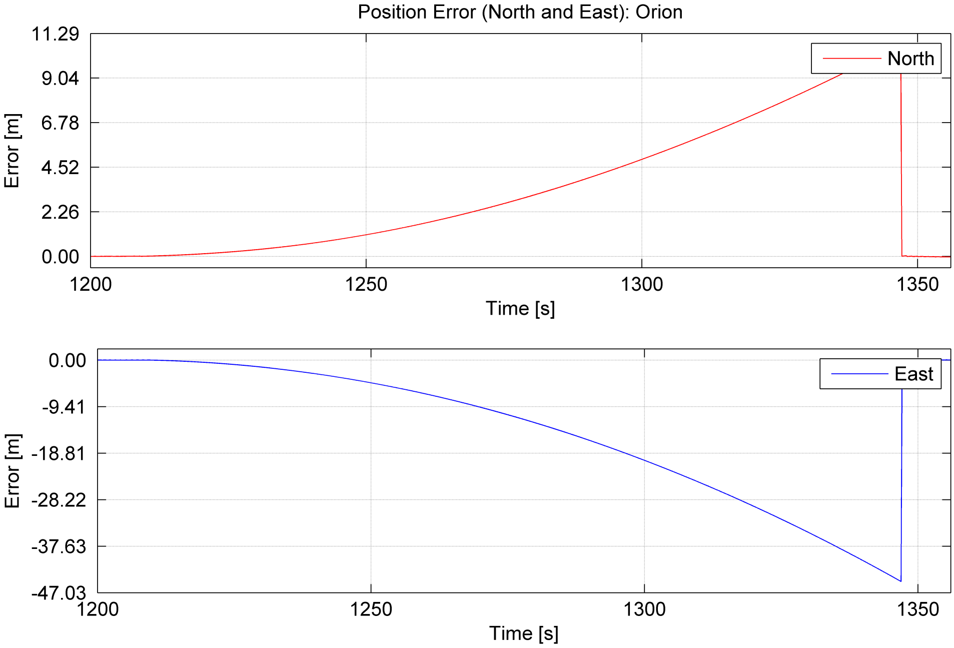


Figure 7.61: Case 3: Position Error (North and East): Orion

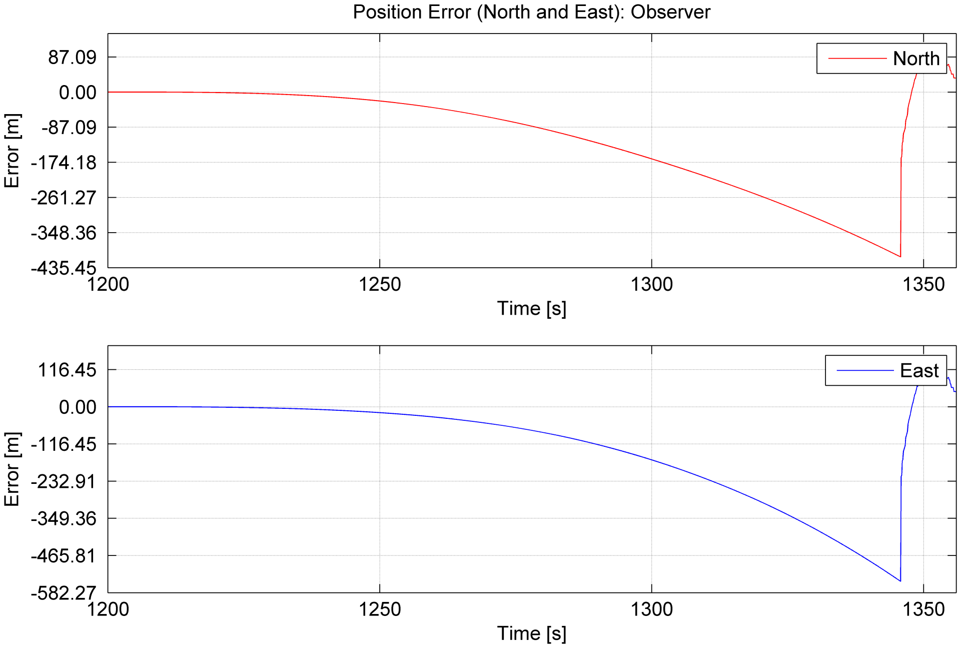


Figure 7.62: Case 3 (Online): Position Error (North and East): Observer

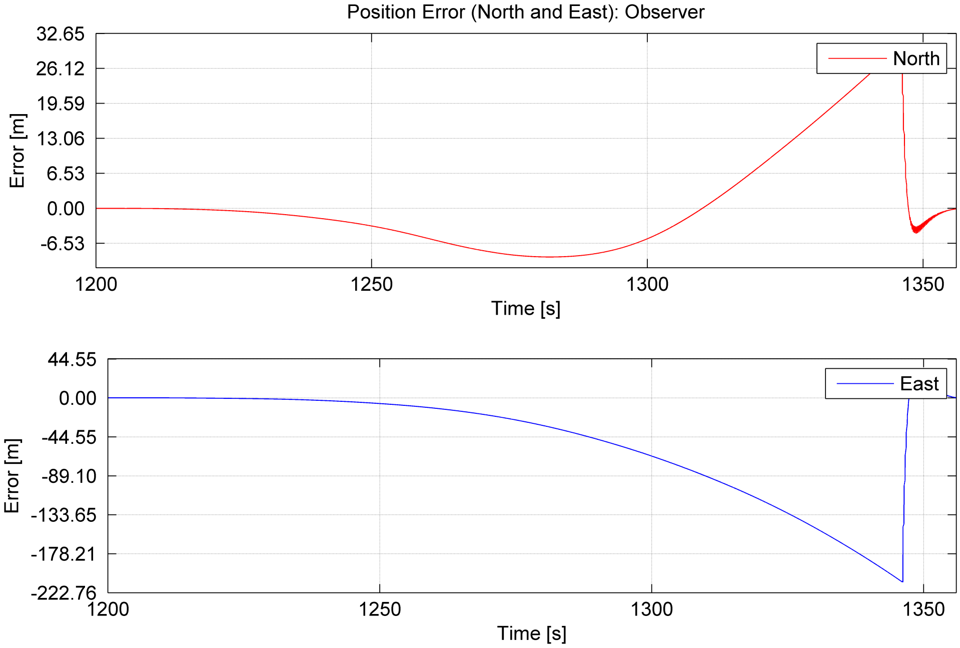


Figure 7.63: Case 3 (Offline): Position Error (North and East): Observer

The attitude, and its corresponding error, for the online and offline results can be viewed in Figure 7.64, 7.65, 7.66 and 7.67. The effect of the position estimate on the estimated

Euler angle estimate can clearly be illustrated in Figure 7.64. Since the attitude observer utlized the gravity for corrections in the injection term, and because of the higher k_I gain, the attitude estimates were more affected by the IMU sensor noise.

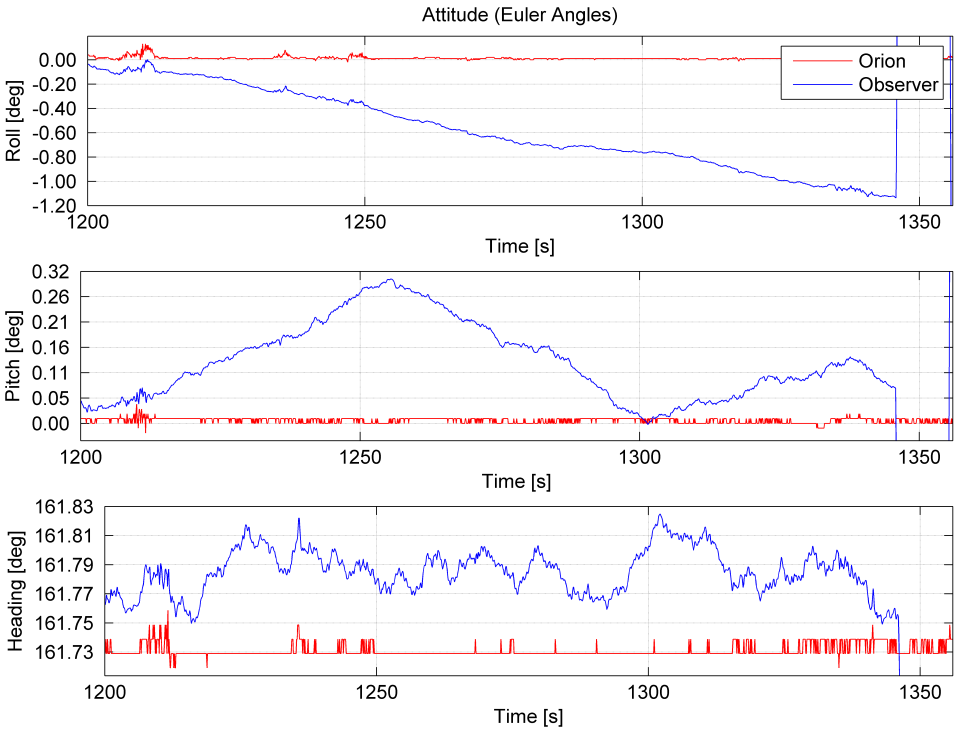


Figure 7.64: Case 3 (Online): Attitude (Euler Angles)

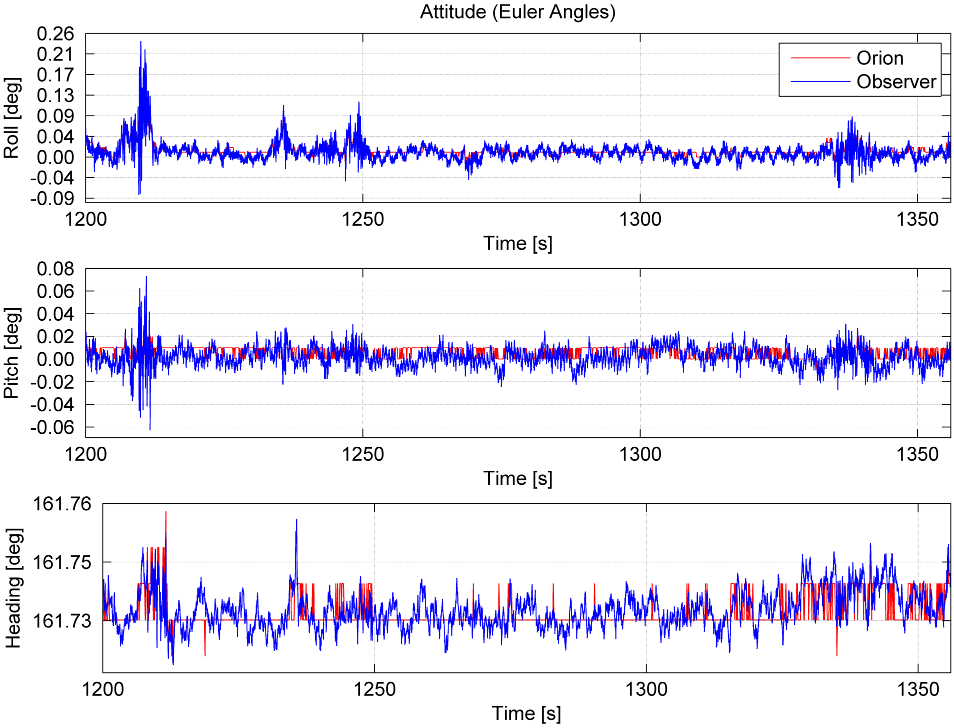


Figure 7.65: Case 3 (Offline): Attitude (Euler Angles)

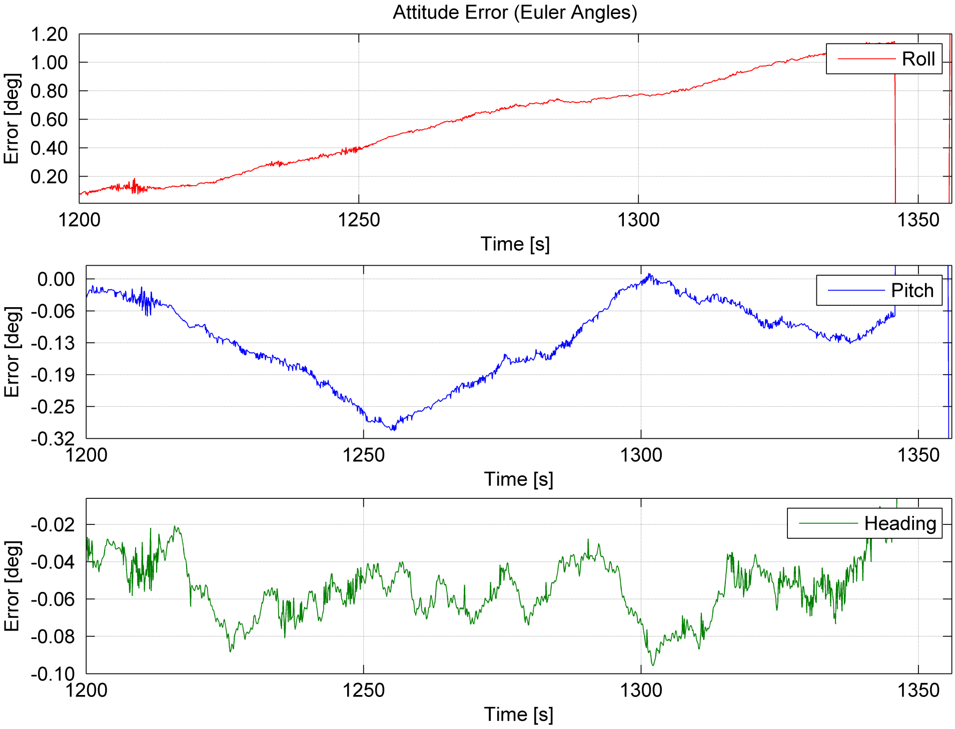


Figure 7.66: Case 3 (Online): Attitude Error (Euler Angles)

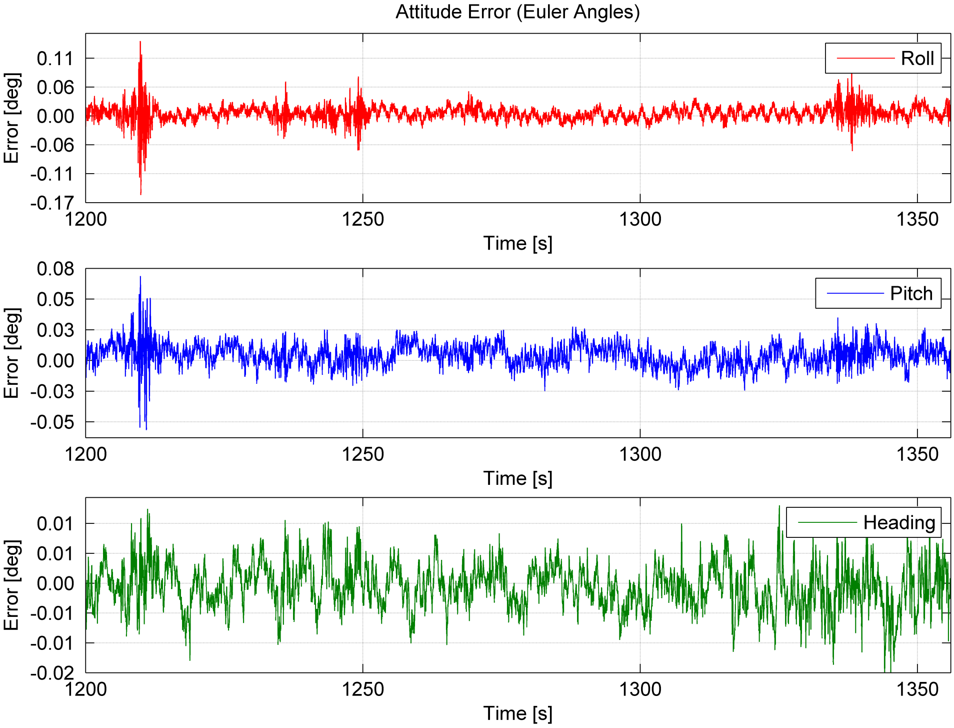


Figure 7.67: Case 3 (Offline): Attitude Error (Euler Angles)

7.4 Discussion

The results from the first test gave an almost satisfactional performance of the nonlinear observer, even though GNSS measurements were lost during the online tests. The corresponding offline simulation results removed this erroneous behaviour, resulting in much better estimates. The dynamic GNSS gain was maybe set to aggressive, where the observer postion estimates followed the GNSS measurements closely. By setting this gain lower the observer should be less affected by errors in the position measurements. The GNSS matching problem was solved in the time after the tests were carried out.

The results showed that the observer managed to estimate the heading with great precision when the gyrocompass was absent, using only the gyroscope measurements from the IMU. The resulting error, when the trolley was placed back at its initial position, was under 0.30 $[deg]$ for both the online and offline test. It was discovered that when the gyrocompass was absent, the attitude observer struggled to estimate the gyroscope bias in the z-axis, due to no heading references. This means that the z-axis estimate would become more errouneous over time, due to the random walk in the sensor bias.

It was additionally discovered that the observer didn't manage to properly estimate the gyroscope bias in the x- and y-axis when the GNSS was absent. This was due to not having feedback from the GNSS for correction of the estimated specific force, which was utilized in the attitude observer's injection term. Since the true gyroscope bias was shown to vary rapidly because of the bias walk in Section 7.3.3, this explained some of the errors in Section 7.3.1 and 7.3.2.

By replacing the estimated specific force with the gravity vector, the observer seemed to be able to continue it's estimation process. This modification is valid when the system experience small or zero non-gravitational accelerations, and was shown when the trolley was stationary in Section 7.3.3. Unfortunately, the same approach would induce errors in Section 7.3.1 and 7.3.2, since the trolley was accelerated under these particular tests.

Making the nonlinear observer perform during dead-reckoning was by far the most challenging task. When the GNSS aiding was gone, the nonlinear observer was strongly influenced by the IMU sensor errors such as noise, bias and axis misalignments. This made the position estimates grow unbounded at an exponential rate. This was also the case for the Orion INS, but a much lower rate.

The resulting error due to IMU sensor noise was observed to affect the estimates at a much smaller degree compared to the results in Cahpter 6. It seemed as the real noise was more correlated with the real signal, increasing the SNR substantially. In addition, the sensor biases on the ADIS16485 IMU were found to be within small regions, i.e. $M_{\hat{b}_g} = 1.6 \cdot 10^{-3}\ [\frac{rad}{s}]$ and $M_{\hat{b}_a} = 4.3 \cdot 10^{-2}\ [\frac{m}{s^2}]$ for the gyroscopes and accelerometers, respectively. These regions were found by taking the norms of the estimated biases. If the accelerometers were used uncalibrated in the nonlinear observer, the estimates can be utilized to calculate maximum attitude error:

$$\begin{aligned}\phi &\approx \arctan(\frac{g_y}{g_z}) = \pm 0.073\ [deg] \\ \theta &\approx -\arctan(\frac{g_x}{\sqrt{g_y^2 + g_z^2}}) = \mp 0.024\ [deg]\end{aligned}$$

Since the attitude errors were found to be $\phi < 0.22\ [deg]$ and $\theta < 0.27\ [deg]$ in Section 7.1, the biggest source of error in the current setup could be IMU misalignments.

The mounting process was found difficult to achieve with high accuracy, where the smallest adjustment could result in an axis misalignment up to several degrees between the IMU and the Orion. In addition, the IMU was listed with an axis-to-frame misalignment of ±1 $[deg]$ [1], where the frame represented the housing of the MEMS sensors. To reduce this error, the IMU had to be mounted with a better accuracy.

Vibrations could also be a substantial source of errors. By studying the freqeuncy spectrum of the IMU measurements it was possible to see some high frequent vibrations, see

Figure 7.68. The Orion INS was most likely the cause of some of these frequencies, due to its high frequent noise. Other sources of vibrations could be caused by nearby computer fans. Furthermore, it was clear that the placement of the IMU was not optimal due to these vibrations.

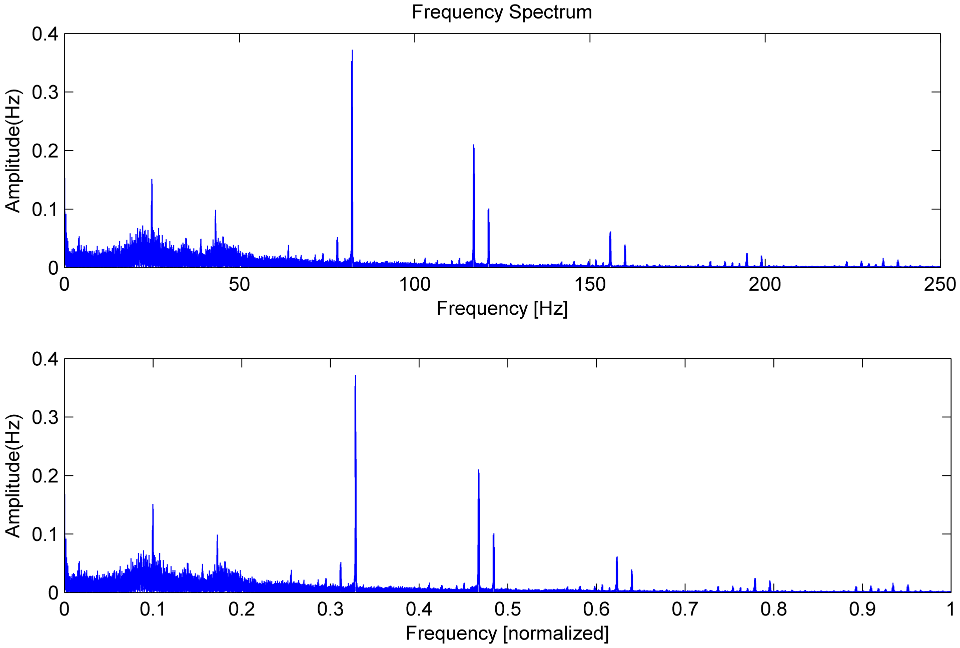


Figure 7.68: Frequency Spectrum of the Accelerometer Measurements (x-axis)

The main problem with vibrations occur when the IMU's sampling rate isn't fast enough to cover the high frequent vibrations. According to the Nyquist theorem, the sampling rate must be at least twice the highest frequency component in the measured signal, [26]. This meant that the IMU wasn't able to correctly sample vibrations higher than 250 $[Hz]$, due to its sampling rate of 500 $[Hz]$. Integrating the IMU's measurements under the presence of vibrations higher than 250 $[Hz]$ would lead to non-zero values, which again would lead to a fast position drift due to a second integration.

To solve the possible vibration problem, the IMU should be mounted away from known vibrations. In addition, the IMU sampling rate should be chosen to be as high as possible, such that any unknown vibrations are sampled. Last, a coning/sculling algorithm should be implemented such that any vibrations are accounted for in real-time, [39].

Noisy position measurements right before the GNSS aiding was lost affected the position estimates during the dead-reckoning phase. The noisy position measurements could project the observer in the wrong direction, and additionally induce larger errors in future estimates. To reduce the influence of noisy GNSS measurements, the GNSS weighting function $k_p(\chi, F_{gnss})$ should be combined with the main observer error $\boldsymbol{\xi}$. The main observer error gave an early indication of inconsistency in the GNSS measurements. In addition, the HDOP value from the GNSS receiver should be utilized for further improvement, see Section 3.2.3 for definition.

The overall performance of the nonlinear observer was experienced to be satisfactory when comparing the estimates with the reference system. When the GNSS aiding was present, the IMU errors was bounded and within a small region. By reducing the errors mentioned above, the future performance of the state estimator will be improved.

Chapter 8

Conclusion and Future Work

The main objective in this thesis was to develop a model-free state estimator for dynamic positioning using IMU, gyrocompass and GNSS. The estimator was compared to a reference system for validation and evaluation of the estimates using a logging system. Section 8.1 gives the conclusion of this thesis, whereas Section 8.2 lists the recommended future work.

8.1 Conclusion

Based on the results of this thesis, the following can be concluded:

- The nonlinear observer of Grip et al. [20] was successfully modified, implemented and tested using IMU, gyrocompass and GNSS, by utilizing simulations in MATLAB. The observer provided good results when it was consistently aided by the GNSS. This resulted in small errors in the presence of noise and bias on the IMU.
- The observer was implemented and tested on a real system using the ADIS16485 MEMS based IMU. Because of some minor implementation problems, the observer missed a significant number of GNSS measurements. This resulted in degraded performance under the experimental tests. Due to the implemented logging system, the results were improved using logged raw data from the sensors in offline MATLAB simulations. These simulations were compared and validated with the TSS Orion INS. When the observer was aided by the GNSS, these results were found promising.

- It was shown that the observer provided great heading estimates under gyrocompass outage, using only gyroscope measurements from the IMU. Under GNSS outage, the observer provided degraded estimates due to IMU sensor errors. It was discovered that the observer didn't manage to correctly estimate the gyroscope bias when gyrocompass and GNSS was absent. A partial solution was presented to solve this problem. This solution yielded good results when the system was stationary, i.e. dynamic positioning.
- A dynamic frequency dependent GNSS gain for the nonlinear observer was developed and implemented with promising results. This generalized the discrete observer to have consistent error dynamics under inconsistent GNSS rates.
- A comparison between the nonlinear observer and the linear Kalman filter were presented to illustrate the effect of unknown external forces on a model-free and a model-based estimation approach. The nonlinear observer was found to be unaffected by the external forces.
- The nonlinear observer implemented in the ECEF frame was found to negatively affect the heading estimate under GNSS outage. This was because of the position dependancy when calculating the Euler angles. The observer should in future applications be implemented in the NED frame to remove this dependancy.

8.2 Future Work

A list of future recommended work are listed below:

- Implement the nonlinear observer in the NED frame, such that the estimated heading is independent of the position estimate.
- Make the nonlinear observer fault-tolerant in the case of GNSS failures. Complement the dynamic GNSS gain such that it utilize HDOP values from the GNSS receiver in addition to the main observer error $\boldsymbol{\xi}$. The observer should also be made fault-tolerant in terms of gyrocompass and IMU failures. Consider the use of MRUs/VRUs to improve the gyroscope bias estimation during GNSS outages.
- Develop and implement logic to improve the accelerometer bias estimation method when the vessel has an angle offset due to different loading setups. The logic should include knowledge about the vessel and IMU sensor characteristics. Consider the inclusion of the vector bias estimation method and MRUs/VRUs for additional information.

- Consider having more than one IMU for redundancy and reduction of errors on the measurements. By taking the mean between measurements of two or more IMUs, the resulting measurement should be less affected by axis misalignments, noise, bias and scale factors.
- The final model-free state estimator should include wave-filtering. Consider how wave-filtering can be combined with the nonlinear observer.
- A complete stability analysis of the modified nonlinear observer should be carried out.

Bibliography

[1] Analog devices adis16485 datasheet (date: 28.02.2014). URL http://www.analog.com/static/imported-files/data_sheets/ADIS16485.pdf.

[2] Norwegian shipowners association: Maritime opportunities in the arctic (date: 23.02.2014). URL http://www.rederi.no/nrweb/mm.nsf/lupgraphics/Nordomrademelding.pdf/$file/Nordomrademelding.pdf.

[3] Matrix class for c++ (date: 09.11.2013). URL http://www.speqmath.com/tutorials/matrix/matrix.html.

[4] Microsemi smartfusion (date: 02.03.2014). URL http://www.microsemi.com/products/fpga-soc/design-resources/dev-kits/smartfusion/smartfusion-evaluation-kit.

[5] Fugro seastar 9200-g2 (date: 25.01.2014). URL http://www.seastar.co.uk/support/9200-g2/.

[6] Tss orion ins manual (date: 28.02.2014). URL http://www.teledyne-tss.com/pdf/T.TSS-ds.Orion%20INS.pdf.

[7] Unmanned systems technology (date: 02.03.2014). URL http://www.unmannedsystemstechnology.com/.

[8] Inertial navigation (date: 27.01.2014). URL http://www.vectornav.com/support/library?id=76.

[9] Glonass (date: 25.01.2014), . URL http://en.wikipedia.org/wiki/GLONASS.

[10] Global positioning system (date: 25.01.2014), . URL http://en.wikipedia.org/wiki/Global_Positioning_System.

[11] Galileo (date: 25.01.2014), . URL http://en.wikipedia.org/wiki/Galileo_(satellite_navigation).

[12] Brian D . 0. Anderson and Mahomed Mansour. New relations between the schur-cohn-fujiwara and nour eldin stability criteria. 1991.

[13] Matthew R. James Barbara F. La Scala, Robert R. Bitmead. Conditions for stability of the extended kalman filter and their application to the frequency tracking problem. 1995.

[14] Nathaniel Bowditch. The american practical navigator - an epitome of navigation. 2002.

[15] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control.* 2011.

[16] Leonard A. McGee Gerald L. Smith, Stanley F. Schmidt. Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. 1962.

[17] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi. Attitude estimation using biased gyro and vector measurements with time-varying reference vectors. May 2011.

[18] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi. Attitude estimation based on time-varying reference vectors with biased gyro and vector measurements. 2012.

[19] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi. Attitude estimation based on time-varying reference vectors with biased gyro and vector measurements. 2012.

[20] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi. Nonlinear observer for aided inertial navigation: Theory and experiments. February 2013.

[21] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi. Globally exponentially stable attitude and gyro bias estimation with application to gnss/ins integration. September 2013.

[22] Tarek Hamel and Robert Mahony. Attitude estimation on so(3) based on direct inertial measurements. IEEE, 2006.

[23] Hofmann-Wellenhof and Lichtenegger. *GPS, GLONASS, Galileo and more.* 2008.

[24] Minh-Duc Hua. Attitude estimation for accelerated vehicles using gps/ins measurements. 2013.

[25] Charles Jeffrey. An introduction to gnss. 2010.

[26] A.J Jerri. The shannon sampling theorem - its various extensions and applications: A tutorial review. 1977.

[27] R. E. Kalman. A new approach to linear filtering and prediction problems. 1960.

[28] T.R. Kane, P.W. Likins, and Levinson D.A. *Spacecraft Dynamics.* 1983.

[29] Hassan K. Khalil. Nonlinear systems (3rd edition). 2001.

[30] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. (5), Jun 2008.

[31] Microblox. Datum transformations and of gps and positions: Application note, 1999. URL `http://www.u-blox.ch/`.

[32] Comparison of Four Gravity Models. David y. hsu. 1996.

[33] Mohammed Ziaur Rahman. Beyond trilateration: Gps positioning geometry and analytical accuracy. 2012.

[34] Jochen Trumpf Robert Mahony, Tarek Hamel and Christian Lageman. Nonlinear attitude observers on so(3) for complementary and compatible measurements: A theoretical study. 2009.

[35] S. Salcudean. A globally convergent angular velocity observer for rigid body motion. IEEE, 1991.

[36] Elmer A. Sperry. Gyro-compass and gyro-pilot.

[37] E. Suli. Numerical solution of ordinary differential equations. 2013.

[38] Arland B. Thompson. Navigation equations in the earth centered earth fixed frame. 2000.

[39] D.H. Titterton and J.L. Weston. *Strapdown Inertial Navigation Technology.* 2004.

[40] Bjornar Vik and Thor I. Fossen. A nonlinear and observer for gps and ins integration. IEEE, 2001.

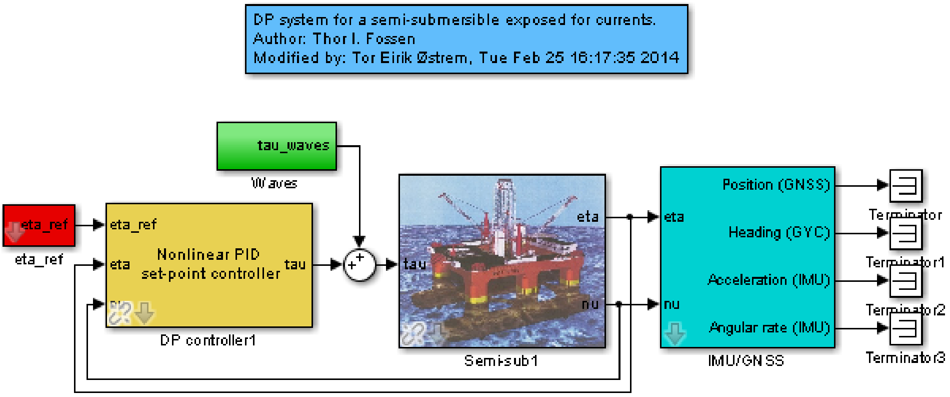
[41] Greg Welch and Gary Bishop. An introduction to the kalman filter. 1997.

Appendix A

Simulator Design

A marine vessel simulator was designed using Simulink to achieve data for testing of the nonlinear observer in MATLAB. The simulator was based on the 6-DOF model of a semi-submersible rig found in the gnc library. A nonlinear PID-controller was utilized to control the semi-sub. Both the model and the controller were implemented by Thor I. Fossen using the NED coordinate system as the inertial frame.

Since the observer was designed in the ECEF frame, some modifications/transformations was needed to achieve the corresponding measurements. The GNSS receiver was placed a defined distance over the center of origin (CO) and was set to give noise and bias free position measurements in LLA coordinates. Whereas the accelerometer and gyroscope measurements from the IMU were implemented using proper measurement models. As the purpose of the simulator was to test the functionality of the observer, the only environmental disturbance included was the wave-induced forces and moments $\boldsymbol{\tau}_{wave}$, realized by the JONSWAP spectrum. See Figure A.1 for the setup in Simulink.



DP system for a semi-submersible exposed for currents.
Author: Thor I. Fossen
Modified by: Tor Eirik Østrem, Tue Feb 25 16:17:35 2014

Figure A.1: Simulink Setup

A.1 Vessel

The 6-DOF semi-submersible model was:

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \mathbf{J}_{\Theta}(\boldsymbol{\Theta}_{nb})\boldsymbol{\nu} \\ \mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{D}\boldsymbol{\nu} + \mathbf{G}\boldsymbol{\eta} &= \boldsymbol{\tau}_{wave} + \boldsymbol{\tau} \end{aligned}$$

where

$$\boldsymbol{\eta} = \begin{bmatrix} (\mathbf{p}^n_{b/n})^\top & (\boldsymbol{\Theta}_{nb})^\top \end{bmatrix}^\top \tag{A.1}$$

$$\boldsymbol{\nu} = \begin{bmatrix} (\mathbf{v}^b_{b/n})^\top & (\boldsymbol{\omega}^b_{b/n})^\top \end{bmatrix}^\top \tag{A.2}$$

The system matrices $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$, $\mathbf{D}$ and $\mathbf{G}$ were found in the Simulink model with the following values:

$$\mathbf{M}_{RB} = 10^{10}\begin{bmatrix} 0.0027 & 0 & 0 & 0 & -0.0530 & 0 \\ 0 & 0.0027 & 0 & 0.0530 & 0 & -0.0014 \\ 0 & 0 & 0.0027 & 0 & 0.0014 & 0 \\ 0 & 0.0530 & 0 & 3.4775 & 0 & -0.0265 \\ -0.0530 & 0 & 0.0014 & 0 & 3.8150 & 0 \\ 0 & -0.0014 & 0 & 0 & -0.0265 & 3.7192 \end{bmatrix} \tag{A.3}$$

$$\mathbf{M}_{A} = 10^{10}\begin{bmatrix} 0.0017 & 0 & 0 & 0 & -0.0255 & 0 \\ 0 & 0.0042 & 0 & 0.0365 & 0 & 0 \\ 0 & 0 & 0.0021 & 0 & 0 & 0 \\ 0 & 0.0365 & 0 & 1.3416 & 0 & 0 \\ -0.0255 & 0 & 0 & 0 & 2.2267 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.2049 \end{bmatrix} \tag{A.4}$$

$$\mathbf{D} = 10^{9}\begin{bmatrix} 0.0004 & 0 & 0 & 0 & -0.0085 & 0 \\ 0 & 0.0003 & 0 & 0.0067 & 0 & -0.0002 \\ 0 & 0 & 0.0034 & 0 & 0.0017 & 0 \\ 0 & 0.0067 & 0 & 4.8841 & 0 & -0.0034 \\ -0.0085 & 0 & 0.0017 & 0 & 7.1383 & 0 \\ 0 & -0.0002 & 0 & -0.0034 & 0 & 0.8656 \end{bmatrix} \tag{A.5}$$

$$\mathbf{G} = 10^{10}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0006 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.4296 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.6212 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.6}$$

The position $\mathbf{p}^n_{b/n}$ was transformed to LLA coordinates, before being converted the the ECEF poisition $\mathbf{p}^e_{b/n}$. The position was then derived once and twice to get the velocity $\mathbf{v}^e_{b/n}$ and acceleration $\mathbf{a}^e_{b/n}$. A small local area was assumed when transforming from NED to ECEF.

A.2 Controller

The controller was implemented as a 3-DOF nonlinear PID controller:

$$\boldsymbol{\tau}_{pid} = \mathbf{K}_p\mathbf{e} + \mathbf{K}_i\int\mathbf{e} - \mathbf{K}_d\dot{\mathbf{e}}$$

with the error $\mathbf{e} = \mathbf{R}_n^b(\psi)(\boldsymbol{\eta}_{ref} - \boldsymbol{\eta}_{pid})$ and the error derivative $\dot{\mathbf{e}} = \boldsymbol{\nu}_{pid}$.

The set point was defined as $\boldsymbol{\eta}_{ref} = \begin{bmatrix} x_{ref}^n & y_{ref}^n & \psi_{ref} \end{bmatrix}^\top$, the position/attitude input was defined as $\boldsymbol{\eta}_{pid} = \begin{bmatrix} x^n & y^n & \psi \end{bmatrix}^\top$, and the velocity input was defined as $\boldsymbol{\nu}_{pid} = \begin{bmatrix} u & v & r \end{bmatrix}^\top$.

The controller design matrices $\mathbf{K}_p$, $\mathbf{K}_i$ and $\mathbf{K}_d$ were already defined in the controller in Simulink, with the following values:

$$\mathbf{K}_p = 0.1 \cdot 0.1 \cdot 10^{10} \begin{bmatrix} 0.0440 & 0 & 0 \\ 0 & 0.0069 & 0 \\ 0 & 0 & 6.9241 \end{bmatrix} \tag{A.7}$$

$$\mathbf{K}_i = 2 \cdot 0.1 \cdot 10^{10} \begin{bmatrix} 0.0440 & 0 & 0 \\ 0 & 0.0069 & 0 \\ 0 & 0 & 6.9241 \end{bmatrix} \tag{A.8}$$

$$\mathbf{K}_d = \frac{0.1^3}{10} \cdot 10^{10} \begin{bmatrix} 0.0440 & 0 & 0 \\ 0 & 0.0069 & 0 \\ 0 & 0 & 6.9241 \end{bmatrix} \tag{A.9}$$

A.3 Measurements

A.3.1 GNSS Model

The lever arm $\mathbf{r}_{gnss}^b = \begin{bmatrix} x_{gnss}^b & y_{gnss}^b & z_{gnss}^b \end{bmatrix}^\top$ denoted the position of the receiver with respect to $\{b\}$, where the position $\mathbf{p}_{b/n}^n$ was transformed to $\mathbf{p}_{gnss/n}^n$ using (2.32) from Section 2.1.2:

$$\mathbf{p}_{gnss/n}^n = \mathbf{p}_{b/n}^n + \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})\mathbf{r}_{gnss}^b \tag{A.10}$$

The NED position $\mathbf{p}_{gnss/n}^n$ was then transformed to LLA coordinates $\mathbf{p}_{gnss/n}^l$, without adding noise or bias.

A.3.2 Gyrocompass Model

The Gyrocompass heading ψ_{GYC} $[deg]$ was calculated from the Euler angle ψ $[rad]$ in the following way:

$$\begin{aligned} \psi_{\pi,3\pi} &= \psi + 2\pi \\ \psi_{GYC} &= \frac{180}{\pi}(\psi_{\pi,3\pi} - 2\pi\text{round}(\frac{\psi_{\pi,3\pi}}{4\pi})) \end{aligned}$$

The heading output ψ_{GYC} yielded an output between 0 [*deg*] and 360 [*deg*]. No noise or bias was added to the heading measurement.

A.3.3 IMU Model

The position of the IMU, denoted with the body-fixed coordinate system $\{m\}$, was assumed to be in CO. This means that the lever arm $\mathbf{r}^b_m = \begin{bmatrix} x^b_m & y^b_m & z^b_m \end{bmatrix}^\top = \mathbf{0}_{3\times1}$, where $\mathbf{r}^b_m$ denoted the position of $\{m\}$ with respect to $\{b\}$. The position, velocity and acceleration could therefore be set as:

$$\mathbf{p}^e_{m/n} = \mathbf{p}^e_{b/n},\ \mathbf{v}^e_{m/n} = \mathbf{v}^e_{b/n},\ \mathbf{a}^e_{m/n} = \mathbf{a}^e_{b/n}.$$

Then the accelerometer and gyroscope models were defined from [15] to be:

$$\begin{aligned} \mathbf{f}^b_{imu} &= \mathbf{R}^b_e(\mathbf{q}^e_b)[\mathbf{a}^e_{m/n} + 2\mathbf{S}(\boldsymbol{\omega}^e_{in})\mathbf{v}^e_{m/n} - \mathbf{g}^e(\mathbf{p}^e_{m/n})] + \mathbf{b}^b_{acc} + \mathbf{w}^b_{acc} \\ \boldsymbol{\omega}^b_{imu} &= \boldsymbol{\omega}^b_{b/n} + \mathbf{R}^b_e(\mathbf{q}^e_b)^\top\boldsymbol{\omega}^e_{ne} + \mathbf{b}^b_{gyr} + \mathbf{w}^b_{gyr} \end{aligned}$$

where $\mathbf{g}^e(\mathbf{p}^e_{m/n})$ was modeled with the J2 gravity model from [32], and $\mathbf{R}^b_e(\mathbf{q}^e_b)$ was found using appropriate function boxes in the Simulink library.

The biases $\mathbf{b}^b_{acc}$ and $\mathbf{b}^b_{gyr}$ were for simplicity defined to be constant during the simulations, i.e. $\dot{\mathbf{b}}^b_{acc} = \mathbf{0}_{3\times1}$ and $\dot{\mathbf{b}}^b_{gyr} = \mathbf{0}_{3\times1}$. The accelerometer and gyroscope noise, $\mathbf{w}^b_{acc}$ and $\mathbf{w}^b_{gyr}$, will be defined in the next section.

Noise

For practical reasons bias and noise were added during offline simulations in MATLAB after collecting the perfect data from the Simulink simulation. The measurement noise $\mathbf{w}^b$ was implemented as a normal distributed white gaussian noise with zero mean and

Noise density	Value
Accelerometers	0.055 $[mg/\sqrt{Hz}]$
Gyroscopes	0.0066 $[deg/s/\sqrt{Hz}]$

Table A.1: Simulation: Noise RMS amplitudes, [1]

a variance $\sigma_n^2 = P_n = A_n^2$, where P_n was the mean noise power and A_n was the root mean square (RMS) amplitude . This can be realized since white noise has a flat power spectrum with $P_n = \sigma_n^2$. The RMS amplitudes for the sensors were found using the ADIS datasheet, see Table A.1.

The RMS amplitudes $A_{acc,n}$ and $A_{gyr,n}$, for a sampling rate of $F_{imu} = 500$ $[Hz]$ and 1 $[mg] = 9.81 \times 10^{-3}$ $[m/s^2]$, can be approximated to:

$$A_{acc,n} = 9.81 \times 10^{-3} \times 0.055 \times \sqrt{500} = 0.0121 \; [m/s^2] \qquad (A.11)$$

$$A_{gyr,n} = \frac{\pi}{180} \times 0.0066 \times \sqrt{500} = 0.0026 \; [rad/s] \qquad (A.12)$$

The noises $\mathbf{w}_{acc}^b$ and $\mathbf{w}_{gyr}^b$ were then implemented with $P_{acc,n} = A_{acc,n}^2 = 0.0121^2$ and $P_{gyr,n} = A_{gyr,n}^2 = 0.0026^2$.

A.4 Environment

The only environmental force being modeled was the wave-induced forces $\boldsymbol{\tau}_w$, which was carried out with the use of the JONSWAP spectrum. This spectrum describes the non-fully developed wave spectra of the North Sea, where fully developed means that the sea is a result of winds blowing over large distances for a long time period.

A.4.1 JONSWAP

The spectral density function is described as, [15]:

$$S(\omega) = 155\frac{H_s^2}{T_1^4}\omega^{-5}\exp(\frac{-944}{T_1^4}\omega^{-4})\gamma^Y \qquad (A.13)$$

where $\gamma = 3.3$ is a normal choice with:

$$Y = \exp(-(\frac{0.191\omega T_1 - 1}{\sqrt{2}\sigma})^2)$$

and

$$\sigma = \begin{cases} 0.07 & \text{for } \omega \leq \frac{5.24}{T_1} \\ 0.09 & \text{for } \omega > \frac{5.24}{T_1} \end{cases} \tag{A.14}$$

T_1 is the average wave period and H_s is the significant wave height.

In the next section the wave spectrum will be approximated using a nonlinear least-squares method.

A.4.2 Approximated Wave Model

From [15] it's possible to approximate first-order wave induced forces using a second-order transfer function $\mathbf{H}_w(s) = \text{diag}\{h^{\{1\}}(s), ..., h^{\{6\}}(s)\}$ for a system with 6-DOF, where:

$$h^{\{i\}}(s) = \frac{K_\omega^{\{i\}} s}{s^2 + 2\lambda^{\{i\}}\omega_0^{\{i\}} s + (\omega_0^{\{i\}})^2}, \text{ for } i = 1, ..., 6 \tag{A.15}$$

where $K_\omega^{\{i\}}$ is defined as:

$$K_\omega^{\{i\}} = 2\lambda^{\{i\}}\omega_0^{\{i\}}\sigma^{\{i\}}$$

where $\lambda^{\{i\}}$ is the damping coefficient, $\omega_0^{\{i\}}$ is the peak frequency and $\sigma^{\{i\}}$ is the wave intensity chosen as:

$$\begin{array}{rcl} P_h^{\{i\}}(\omega_0^{\{i\}}) & = & S(\omega_0^{\{i\}}) \\ & \Updownarrow & \\ (\sigma^{\{i\}})^2 & = & \max\limits_{0<\omega<\infty} S(\omega) \end{array}$$

where $S(\omega)$ is the spectral density function of the JONSWAP wave spectra, defined by (A.13), and $P_{hh}^{\{i\}}(\omega_0^{\{i\}})$ is the spectral density function of the corresponding $h^{\{i\}}(s)$ defined from [15] as:

$$P_{hh}^{\{i\}}(\omega) = |h^{\{i\}}(j\omega = s)|^2 = \frac{4(\lambda^{\{i\}}\omega_0^{\{i\}}\sigma^{\{i\}})^2\omega^2}{((\omega_0^{\{i\}})^2 - \omega^2)^2 + 4(\lambda^{\{i\}}\omega_0^{\{i\}}\omega)^2} \tag{A.16}$$

The damping coefficient $\lambda^{\{i\}}$ can be chosen such that $P_{hh}^{\{i\}}(\omega_0^{\{i\}})$ matches $S(\omega^{\{i\}})$ using a nonlinear least-squares method. For simplicity, the approximation was chosen to be the same for each DOF, i.e. $h(s) = h^{\{i\}}(s)$.

Further, by selcting the peak frequency $\omega_0 = 1.2$ $[rad/s]$ and the significant wave height $H_s = 10$ $[m]$ the approximation is illustrated in Figure A.2, for $\sigma = 0.3218$ and $\lambda = 0.1017$.

According to [15] $\boldsymbol{\tau}_w$ can be approximated linearly as:

$$\boldsymbol{\tau}_w \approx \mathbf{K}_w\mathbf{H}_w(s)\mathbf{w}_w(s) \tag{A.17}$$

where $\mathbf{K}_w$ is designed such that the position and attitude vector η is affected desirely. The elements of $\mathbf{H}_w(s)$ are found from (A.15), and $\mathbf{w}_w(s) = \begin{bmatrix} w_w^{\{1\}} & w_w^{\{2\}} & w_w^{\{3\}} & w_w^{\{4\}} & w_w^{\{5\}} & w_w^{\{6\}} \end{bmatrix}^\top$ where $w_w^{\{1\}}$ is a Gaussian white noise process with zero mean and the spectral density $P_{ww}^{\{i\}}(\omega) = 1$.

A.4.3 The Simulated Seastate

By choosing the wave peak frequency as $\omega_0 = 1.2$ $[rad/s]$ and the significant wave height $H_s = 1$ $[m]$, the JONSWAP wave spectrum can be linearly approximated using `wavespec` and `lsqcurvefit in jonswap.m,` found in the gnc library in from the mss toolbox in `MATLAB`. This resulted in $\sigma = 0.3218$ and $\lambda = 0.1017$. See Figure A.2 for plot of linearized spectrum.

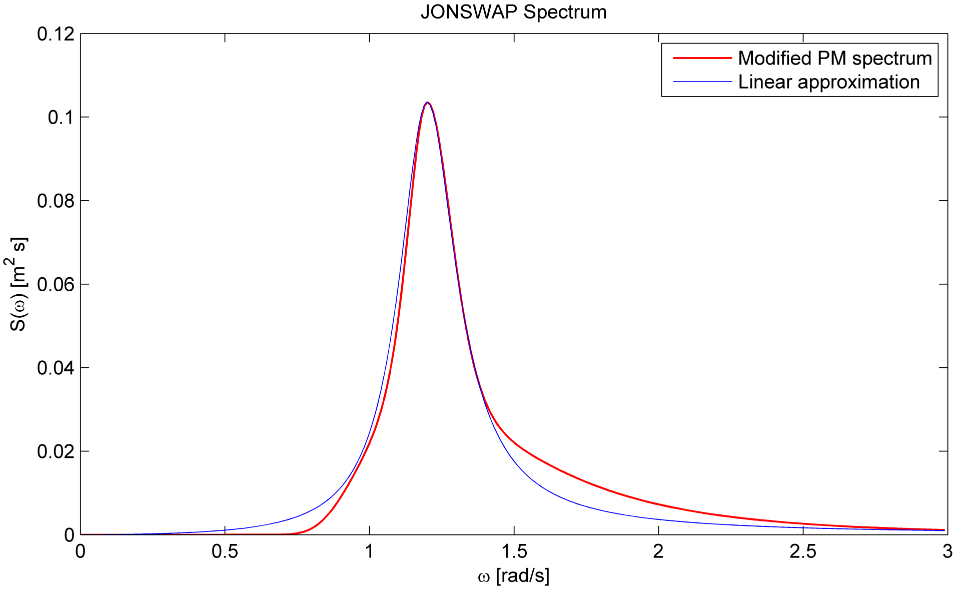


Figure A.2: Linearized JONSWAP Spectrum

Appendix B

Kalman Filter and Nonlinear Observer (NED)

This appendix gives an overview of the nonlinear observer and the Kalman filter (KF) for DP utilized in the comparison in Section 6.5, with corresponding initial conditions and design matrices for the KF. Both state estimators were implemented in the NED frame. Figure B.1 illustrates the Simulink setup.

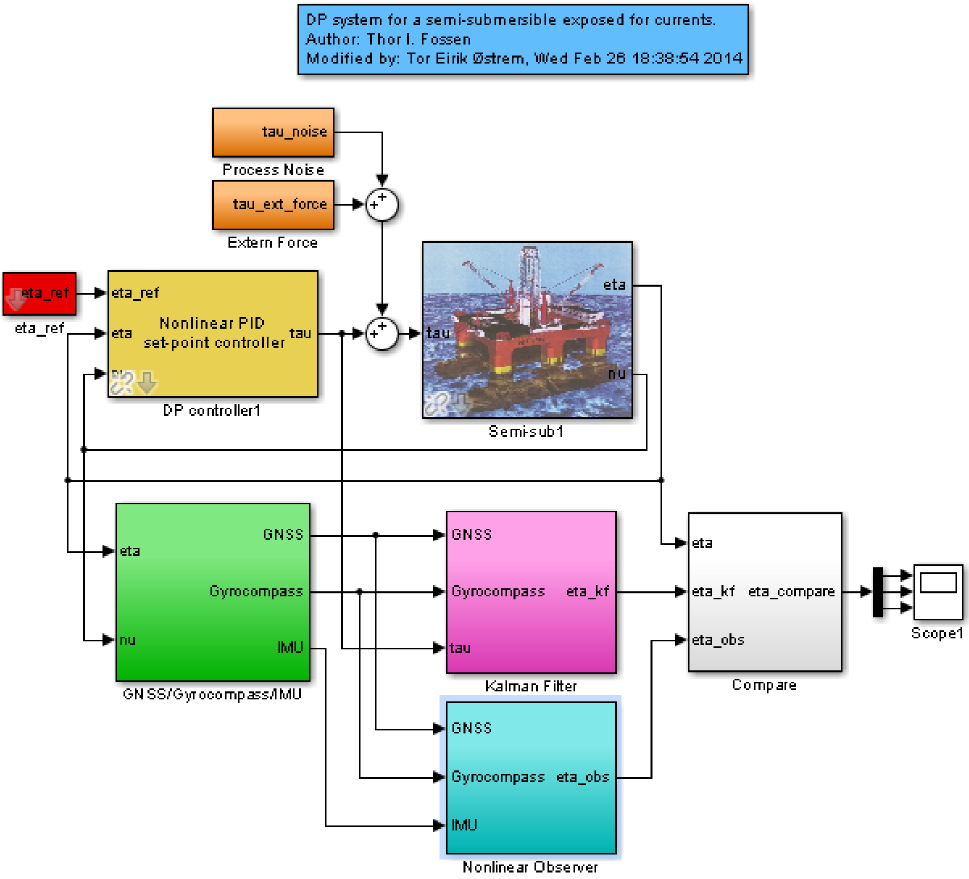


Figure B.1: Simulink Setup

B.1 Nonlinear Observer

When deriving the nonlinear observer in the NED frame, the Earth's rotational rate can be neglegted since the vessel is assumed to navigate in a small local area. The gravitational model can be replaced with $\mathbf{g}^n = \begin{bmatrix} 0 & 0 & 9.81 \end{bmatrix}^\top$ and the north seeking vector in the gyrocompass substitute is $\mathbf{c}^n = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top$. With these simplifactions, the nonlinear observer in the NED frame can be given by:

$$\begin{aligned}
\dot{\hat{\mathbf{p}}}^n &= \hat{\mathbf{v}}^n + \theta_{GNSS}\mathbf{K}_{pp}(\mathbf{p}^n_{GNSS} - \hat{\mathbf{p}}^n) && \text{(B.1)}\\
\dot{\hat{\mathbf{v}}}^n &= \hat{\mathbf{f}}^n + \mathbf{g}^n \\
&\quad +\theta^2_{GNSS}\mathbf{K}_{vp}(\mathbf{p}^n_{GNSS} - \hat{\mathbf{p}}^n) && \text{(B.2)}\\
\dot{\boldsymbol{\xi}} &= -\mathbf{R}^n_b(\hat{\mathbf{q}}^n_b)\mathbf{S}(\hat{\boldsymbol{\sigma}})\mathbf{f}^b_{IMU} \\
&\quad +\theta^3_{GNSS}\mathbf{K}_{\xi p}(\mathbf{p}^n_{GNSS} - \hat{\mathbf{p}}^n) && \text{(B.3)}\\
\hat{\mathbf{f}}^n &= \mathbf{R}^n_b(\hat{\mathbf{q}}^n_b)\mathbf{f}^b_{IMU} + \boldsymbol{\xi} && \text{(B.4)}\\
\dot{\hat{\mathbf{q}}}^n_b &= \frac{1}{2}\hat{\mathbf{q}}^n_b \otimes (\bar{\boldsymbol{\omega}}^b_{ib,IMU} - \bar{\hat{\mathbf{b}}}^b_g + \bar{\hat{\boldsymbol{\sigma}}}) && \text{(B.5)}\\
\dot{\hat{\mathbf{b}}}^b_g &= \mathbf{Proj}(\hat{\mathbf{b}}^b_g, -k_I\hat{\boldsymbol{\sigma}}) && \text{(B.6)}
\end{aligned}$$

where

$$\hat{\boldsymbol{\sigma}} = k_1\underline{\mathbf{f}^b} \times \mathbf{R}^n_b(\hat{\mathbf{q}}^n_b)^\top\underline{\hat{\mathbf{f}}^n} + k_2\underline{\mathbf{c}^b} \times \mathbf{R}^n_b(\hat{\mathbf{q}}^n_b)^\top\underline{\mathbf{c}^n} \tag{B.7}$$

and

$$\underline{\mathbf{c}^b} = \frac{\mathbf{c}^b}{\|\mathbf{c}^b\|},\ \underline{\mathbf{c}^e} = \frac{\mathbf{c}^n}{\|\mathbf{c}^n\|},\ \underline{\mathbf{f}^b} = \frac{\mathbf{f}^b_{IMU} - \hat{\mathbf{b}}^b_a}{\|\mathbf{f}^b_{IMU}\|},\ \underline{\hat{\mathbf{f}}^n} = \frac{\hat{\mathbf{f}}^n}{max\{\|\hat{\mathbf{f}}^n\|, \delta\}} \tag{B.8}$$

Since the accelerometer bias was absent, the corresponding terms were removed.

B.2 Kalman Filter

The linear discrete KF has the well-known structure, [15]:

$$\begin{aligned}
\mathbf{K}(k) &= \bar{\mathbf{P}}(k)\mathbf{H}^\top[\mathbf{H}(k)\bar{\mathbf{P}}(k)\mathbf{H}^\top + \mathbf{R}]^{-1} && \text{(B.9)}\\
\hat{\mathbf{x}}(k) &= \bar{\mathbf{x}}(k) + \mathbf{K}(k)[\mathbf{y}(k) - \mathbf{H}\bar{\mathbf{x}}(k)] && \text{(B.10)}\\
\hat{\mathbf{P}}(k) &= [\mathbf{I} - \mathbf{K}(k)\mathbf{H}]\bar{\mathbf{P}}(k)[\mathbf{I} - \mathbf{K}(k)\mathbf{H}]^\top \\
&\quad +\mathbf{K}(k)\mathbf{R}\mathbf{K}^\top(k) && \text{(B.11)}\\
\bar{\mathbf{x}}(k+1) &= \boldsymbol{\Phi}\hat{\mathbf{x}}(k) + \boldsymbol{\Delta}\mathbf{u}(k) && \text{(B.12)}\\
\bar{\mathbf{P}}(k+1) &= \boldsymbol{\Phi}\hat{\mathbf{P}}(k)\boldsymbol{\Phi}^\top + \boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^\top(k) && \text{(B.13)}
\end{aligned}$$

where (B.9) represents the kalman gain matrix, (B.10) the state estimate update, (B.11) the error covariance update, (B.12) the state estimate propagation and (B.13) the error covariance propagation.

The design matrices $\mathbf{Q}$ and $\mathbf{R}$ represent the process noise covariance matrix and the measurement covariance matrix, respectively, defined as $\mathbf{Q} = \mathbf{Q}^\top > 0$ and $\mathbf{R} = \mathbf{R}^\top > 0$. Further, the initial conditions are defined as $\bar{\mathbf{x}}(0) = \mathbf{x}_0$ and $\bar{\mathbf{P}}(0) = E[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^\top] = \mathbf{P}_0$, where $\hat{\mathbf{P}}(k) = \hat{\mathbf{P}}^\top(k) > 0$ is computed online.

The discrete Kalman Filter (B.9-B.13) is based on the following discretized system model, [15]:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Delta}\mathbf{u}(k) + \mathbf{\Gamma}\mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k)\end{aligned}$$

where

$$\mathbf{\Phi} = \exp(\mathbf{A}h) \approx \mathbf{I} + \mathbf{A}h, \ \mathbf{\Delta} = \mathbf{A}^{-1}(\mathbf{\Phi} - \mathbf{I})\mathbf{B}, \ \mathbf{\Gamma} = \mathbf{A}^{-1}(\mathbf{\Phi} - \mathbf{I})\mathbf{E} \tag{B.14}$$

$\mathbf{\Phi}$, $\mathbf{\Delta}$ and $\mathbf{\Gamma}$ represent the discretized system matrices based on the linear state space model:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w} \tag{B.15}$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \tag{B.16}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the input vector $\mathbf{w}$ is the process noise vector, $\mathbf{y}$ is the output vector, $\mathbf{v}$ is the measurement noise vector, $\mathbf{A}$ is the system matrix, $\mathbf{B}$ is the input matrix, $\mathbf{E}$ is the process noise matrix and $\mathbf{H}$ is the output matrix.

B.2.1 Kalman Filter for Dynamic Positioning

The Kalman Filter utilized in 6.5 is based on the model for the semi-submersible rig used to achieve the simulation data for comparison. Since the Kalman filter was supposed to estimate $\boldsymbol{\eta} = \begin{bmatrix} N & E & \psi \end{bmatrix}^\top$ and $\boldsymbol{\nu} = \begin{bmatrix} u & v & r \end{bmatrix}^\top$, the system matrices for the rig was reduced to:

$$\mathbf{M}_{RB} = 10^{10}\begin{bmatrix} 0.0027 & 0 & 0 \\ 0 & 0.0027 & -0.0014 \\ 0 & -0.0014 & 3.7192 \end{bmatrix} \tag{B.17}$$

$$\mathbf{M}_{A} = 10^{10}\begin{bmatrix} 0.0017 & 0 & 0 \\ 0 & 0.0042 & 0 \\ 0 & 0 & 3.2049 \end{bmatrix} \tag{B.18}$$

$$\mathbf{D} = 10^{9}\begin{bmatrix} 0.0004 & 0 & 0 \\ 0 & 0.0003 & -0.0002 \\ 0 & -0.0002 & 0.8656 \end{bmatrix} \tag{B.19}$$

$$\mathbf{G} = 10^{10}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{B.20}$$

The linearized system (2.60-2.62), with $\mathbf{b} \in \mathbb{R}^{3\times 1}$ representing the unmodeled nonlinear dynamics and the gaussian noise vectors $\mathbf{w}_1 \in \mathbb{R}^{3\times 1}$ and $\mathbf{w}_2 \in \mathbb{R}^{3\times 1}$ representing uncertainty of the linearized model, can be given as, [15]:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{\nu} \tag{B.21}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{D}\boldsymbol{\nu} + \mathbf{G}\boldsymbol{\eta} = \boldsymbol{\tau} + \mathbf{b} + \mathbf{w}_2 \tag{B.22}$$

The unmodeled nonlinear dynamics$\mathbf{b}$ is normally modeled as:

$$\dot{\mathbf{b}} = \mathbf{w}_1$$

By defining the state vector $\mathbf{x} = \begin{bmatrix} \boldsymbol{\eta} & \mathbf{b} & \boldsymbol{\nu} \end{bmatrix}^\top$, (B.21-B.22) can be represented by the linear state space model:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w} \tag{B.23}$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} \tag{B.24}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\mathbf{M}^{-1}\mathbf{G} & \mathbf{M}^{-1} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix} \tag{B.25}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{M}^{-1} \end{bmatrix} \tag{B.26}$$

$$\mathbf{E} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{M}^{-1} \end{bmatrix} \tag{B.27}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{B.28}$$

Then the discretized system matrices can be achieved by (B.14), such that the Kalman filter can be calculated using (B.9-B.13).

The initial conditions for the KF were defined as:

$$\bar{\mathbf{P}}(0) = 10^{-6}\mathbf{I}_{9\times9} \tag{B.29}$$

$$\bar{\mathbf{x}}(0) = \mathbf{0}_{9\times1} \tag{B.30}$$

After some trial, the design matrices were calculated as:

$$\mathbf{Q} = \begin{bmatrix} 0.001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^7 \end{bmatrix} \tag{B.31}$$

$$\mathbf{R} = 10^{-10}\mathbf{I}_{3\times3} \tag{B.32}$$

Appendix C

Observer Code

C.1 MATLAB

The MATLAB code listed under illustrate the nonlinear observer represented in ECEF, with the mean filtering accelerometer bias estimation method, and additional estimates such as NED velocity and the Euler angles. The complete source code for both C++ and MATLAB can be found in the digital appendix, in addition to the processing code for the logged data.

```
function ...
    [p_e,v_e,xi_e,q_be,b_g,w,...                % Obs. est.
    k_GYC,k_GPS,n_GYC,n_GPS,...                 % Msr. counters and flags
    eul_nb, v_n]...                             % Extra estimates
    = observer(...
    p_e_L,v_e_L,xi_e_L,q_be_L,b_g_L,w_L,...     % Obs. last est.
    IMU_buff,GYC_buff,GPS_buff,...              % Msr. buffers
    k_GYC_L,k_GPS_L,n_GYC_L,n_GPS_L,...         % Msr. counters and flags
    t,r_GNSS,r_IMU)                             % Time and lever arms

%OBS Summary of this function goes here
%   Detailed explanation goes here
%   Observer:       50 Hz
%   IMU:            100 Hz
%   Gyrocompass:    10 Hz
%   GPS:            1 Hz

% Tuning parameters
theta = 2;
X = 0.5;
k_0_r = 20;
k_1 = 1.5;
k_2 = 5;
k_0_I = 20;
```

```
k_I = 0.001;

% System and sensor parameters
M_f = 9.80665;
M_bg = .35*pi/180; M_bg_h = .36*pi/180;
m_f = 0.98*norm(M_f); delta = 0.98*m_f;
Mw = 0.6; a_w = 0;
mg = 9.80665/1000;
% For the gyrocompass injection
% term at small local areas
const_lat = 40;
const_lon = 10;

% Increment estimates
p_e = p_e_L; v_e = v_e_L; xi_e = xi_e_L; q_be = q_be_L;
b_g = b_g_L; w = w_L;
k_GPS = k_GPS_L; k_GYC = k_GYC_L;
n_GPS = n_GPS_L; n_GYC = n_GYC_L;
Gamma = Gamma_L;

% Initial gain boost
t_init = 360;
if t < t_init,
    k_1 = k_1*k_0_r;
    k_2 = k_2*k_0_r;
    k_I = k_I*k_0_I;
else
    t_gyr = t-t_init;
    beta0 = 1.1; T_gyr1 = 2*t_init; k_I_h = k_I*k_0_I; k_I_l = 5*k_I;
    tau0 = T_gyr1/(log(k_I_h-k_I_l)-log((beta0-1)*k_I_l));
    k_I = k_I_l+(k_I_h-k_I_l)*exp(-t_gyr/tau0);
end

% WGS84
a = 6378137; f = 1/298.257223563; b = a*(1-f);
e = sqrt((a^2-b^2)/a^2); e_ = sqrt((a^2-b^2)/b^2);

% Measurements in different frequencies
F_OBS = 50; F_IMU = 500;
% Euler discrete integration time
T_OBS = 1/F_OBS; T_IMU = T_OBS/size(IMU_buff,2);

% Arm from CG {c} to IMU {m} (BODY)
r_cm = r_IMU';
% Arm from CG {c} to GNSS {g} (BODY)
r_cg = r_GNSS';
% Arm from IMU {m} to GNSS {g} (BODY)
r_mg = r_cg-r_cm;

% Integration loop
for i=1:size(IMU_buff,2),
    % Extract msr. from buffer
    f_IMU_b = mg*[f_IMU_b(1,1);f_IMU_b(2,1);f_IMU_b(3,1)];
    o_IMU_b = (pi/180)*[o_IMU_b(1,1);o_IMU_b(2,1);o_IMU_b(3,1)];

    % Rotation matrix (BODY to ECEF)
    R_be = eye(3)+2*q_be(1)*Smtrx(q_be(2:4))+2*Smtrx(q_be(2:4))^2;

    % Calculate gravity in ECEF
    GM = 3986005e8; J2 = 108263e-8; a = 6378137; omega_ie = [0;0;7.292115e-5];
    p_norm = norm(p_e); gti2 = J2*a^2/p_norm^2;
    gti3 = J2*a^2*p_e(3)^2/p_norm^4;
    g_p1 = -p_e(1)*GM/p_norm^3*(1+3/2*gti2-15/2*gti3)+p_e(1)*omega_ie(3)^2;
    g_p2 = -p_e(2)*GM/p_norm^3*(1+3/2*gti2-15/2*gti3)+p_e(2)*omega_ie(3)^2;
    g_p3 = -p_e(3)*GM/p_norm^3*(1+9/2*gti2-15/2*gti3);
    g_p = [g_p1; g_p2; g_p3];

    % Estimate of ECEF specific force
    b_a = w(2:4,1);
    f_IMU = f_IMU_b-b_a;
    f_e = R_be*f_IMU+xi_e;

    % GNSS MEASUREMENT
    k_GPS = k_GPS + 1;
    F_GPS = 1/(k_GPS*T_IMU);
    K_GPS = F_IMU/F_GPS;
    if n_GPS == 1 && i == size(IMU_buff,2) && open_loop == 0,
        % Compute ECEF from LLA (closed method)
        % GPS [deg]->[rad]
        GPS_buff = [(pi/180)*GPS_buff(1,1);(pi/180)*GPS_buff(2,1);...
            GPS_buff(3,1)];
        p_GPS_l_g = GPS_buff;
        N = a/sqrt(1-e^2*sin(p_GPS_l_g(1))^2);
        p_GPS_e_g(1) = (N+p_GPS_l_g(3))*cos(p_GPS_l_g(1))*cos(p_GPS_l_g(2));
        p_GPS_e_g(2) = (N+p_GPS_l_g(3))*cos(p_GPS_l_g(1))*sin(p_GPS_l_g(2));
        p_GPS_e_g(3) = (b^2/a^2*N+p_GPS_l_g(3))*sin(p_GPS_l_g(1));
        p_GPS_e_g = p_GPS_e_g';
        % Position of GNSS with respect to {m}
        p_GPS_e = p_GPS_e_g - R_be*r_mg;
        % Reset counter and new measurement flag
        k_GPS = 0;
        n_GPS = 0;
        % Set discrete gain
        D_GPS = K_GPS;
        % Dynamic pos. gain
        k_p = X*F_GPS;
        K_pp = k_p*eye(3)*.6;
        K_vp = k_p*eye(3)*.11;
        K_xip = k_p*eye(3)*.006;
    else
        K_pp = zeros(3);
        K_vp = zeros(3);
        K_xip = zeros(3);
        D_GPS = 0;
    end

    % GYROCOMPASS MEASUREMENT
    k_GYC = k_GYC + 1;
    F_GYC = 10;
    K_GYC = F_IMU/F_GYC;
    if n_GYC == 1 && i == size(IMU_buff,2),
        phi = 0;
        the = 0;
        % Heading to yaw
        GYC_buff = (pi/180)*(GYC_buff-round(GYC_buff/(360))*360);
        psi = GYC_buff;
        % BODY vector
        s_ph = sin(phi); c_ph = cos(phi);
        s_th = sin(the); c_th = cos(the);
        s_ps = sin(psi); c_ps = cos(psi);
        gc_b = [c_ps*c_th;-s_ps*c_ph+c_ps*s_th*s_ph;...
            s_ps*s_ph+c_ps*c_ph*s_th];
        % ECEF vector
        lat_gc = degtorad(const_lat); lon_gc = degtorad(const_lon);
        s_la = sin(lat_gc); c_la = cos(lat_gc);
        s_lo = sin(lon_gc); c_lo = cos(lon_gc);
        gc_e = [-c_lo*s_la;-s_la*s_lo;c_la];
        % Reset counter and new measurement flag
        k_GYC = 0;
        n_GYC = 0;
        % Set discrete gain
        D_GYC = K_GYC;
    else
        gc_b = ones(3,1);
        gc_e = ones(3,1);
        D_GYC = 0;
    end

    % Determine observer gains
    if D_GPS > 0 && D_GYC > 0,      % GPS && GYC
        K_1 = k_1;
        K_2 = k_2;
        K_I = k_I;
    elseif D_GPS > 0 && D_GYC == 0, % GPS && !GYC
        K_1 = k_1;
        K_2 = 0;
        K_I = k_I;
    elseif D_GPS == 0 && D_GYC > 0, % !GPS && GYC
        K_1 = k_1;
        K_2 = k_2;
        K_I = k_I;
    else                            % !GPS && !GYC
        K_1 = k_1;
        K_2 = 0;
        K_I = k_I;
    end

    % Body-fixed vectors and ECEF reference vectors
    v1b = f_IMU/max(norm(f_IMU),delta);
    v1e = f_e/max([norm(f_e),delta]);
    v2b = gc_b/norm(gc_b);
    v2e = gc_e/norm(gc_e);

    % Prediction
    v1b_p = R_be'*v1e;
    v2b_p = R_be'*v2e;

    % Injection term
    sig = K_1*cross(v1b,v1b_p)+K_2*cross(v2b,v2b_p);
    if t > t_init,
       sig = K_1*cross(v1b,v1b_p)+K_2*[0;0;1].*cross(v2b,v2b_p);
    end

    % Accelerometer bias: Mean Filtering Method
    beta1 = 1.1; T_acc1 = 1000; fch1 = 0.05; fcl1 = 0.000005;
    tau1 = T_acc1/(log(fch1-fcl1)-log((beta1-1)*fcl1));
    fc1 = fcl1+(fch1-fcl1)*exp(-t/tau1);
    wc1 = 2*pi*fc1; Tf1 = 1/wc1;
    db_a = 1/(Tf1+T_IMU)*[f_IMU_b(1:2)-b_a(1:2);0];
    if t >= t_init,
        t_acc = t-t_init;
        beta2 = 1.1; T_acc2 = 500; fch2 = 0.02; fcl2 = 0.0002;
        tau2 = T_acc2/(log(fch2-fcl2)-log((beta2-1)*fcl2));
        fc2 = fcl2+(fch2-fcl2)*exp(-t_acc/tau2);
        wc2 = 2*pi*fc2; Tf2 = 1/wc2;
        f_g = R_be'*g_p;
        db_a_z = 1/(Tf2+T_IMU)*(f_IMU_b(3)+f_g(3)-b_a(3));
        db_a(3) = db_a_z;
    end
    %db_a = zeros(3,1); % NO ACC BIAS
    b_a = b_a+T_IMU*db_a;
    w(1,1) = norm(b_a)^2;
    w(2:4,1) = b_a;


    % Corrector
    if D_GPS > 0,
        p_e_i = p_e;
        p_e = p_e + T_IMU*D_GPS*theta*K_pp*(p_GPS_e-p_e_i);
        v_e = v_e + T_IMU*D_GPS*theta^2*K_vp*(p_GPS_e-p_e_i);
        xi_e = xi_e + T_IMU*D_GPS*theta^3*K_xip*(p_GPS_e-p_e_i);
    end
    % Predictor
    v_e = v_e + T_IMU*(-2*Smtrx(omega_ie)*v_e+f_e+g_p);
    p_e = p_e + T_IMU*v_e;
    xi_e = xi_e + T_IMU*(-R_be*Smtrx(sig)*f_IMU+R_be*db_a);

    % Attitude estimation
    omegah = o_IMU_b-b_g+sig;
    Theta = [   0, -omegah';
                omegah, -Smtrx(omegah)];
    Thetae = [  0, -omega_ie';
                omega_ie, Smtrx(omega_ie)];
    dq_be = .5*(Theta-Thetae)*q_be+(1-q_be'*q_be)*q_be;
    q_be = q_be+T_IMU*dq_be;
    q_be = q_be/norm(q_be);

    % Gyroscope bias estimation
    db_g = -K_I*sig;
    if b_g'*b_g > M_bg^2 && b_g'*db_g > 0
        c = min([1, (b_g'*b_g-M_bg^2)/(M_bg_h^2-M_bg^2)]);
        db_g = (eye(3)-c*(b_g*b_g')/(b_g'*b_g))*db_g;
    end
    b_g = b_g+T_IMU*db_g;
end


% Compute LLA in {c} from ECEF {c} (closed method)
lon = atan(p_e_c(2)/p_e_c(1));
p = sqrt(p_e_c(1)^2+p_e_c(2)^2);
theta_ = atan((p_e_c(3)*a)/(p*b));
lat = atan((p_e_c(3)+e_^2*b*sin(theta_)^3)/(p-e^2*a*cos(theta_)^3));
N = a/sqrt(1-e^2*sin(lat)^2);
h = p/cos(lat)-N;
p_l_c = [lat;lon;h];

% Quaternion from ECEF to NED
q_lon = [cos(lon/2);0;0;- sin(lon/2)];
q_lat = [cos((lat+pi/2)/2);0;sin((lat+pi/2)/2);0];
q_en = [q_lat(1)*q_lon(1)-q_lat(2:4)'*q_lon(2:4);
    q_lat(1)*q_lon(2:4)+q_lon(1)*q_lat(2:4)+cross(q_lat(2:4),q_lon(2:4))];
R_en = Rquat(q_en);

% Compute attitude
```

```
q_nb = [ q_en(1)*q_be(1)-q_en(2:4)'*q_be(2:4);
        q_en(1)*q_be(2:4)+q_be(1)*q_en(2:4)+cross(q_en(2:4),q_be(2:4))];
q_nb = q_nb/norm(q_nb);
R_nb = Rquat(q_nb);
eul_nb = [atan2(R_nb(3,2),R_nb(3,3));
    -asin(R_nb(3,1));
     atan2(R_nb(2,1),R_nb(1,1))];

% Convert to degrees (and yaw to heading)
eul_nb(1:2,1) = (180/pi)*eul_nb(1:2,1);
psi_3pi = eul_nb(3,1)+2*pi;
eul_nb(3,1) = (180/pi)*(psi_3pi-2*pi*round(psi_3pi/(4*pi)));

% Rotate velcoity from ECEF to NED
v_n = R_en*v_e;

end
```

Appendix D

Additional Data

D.1 Raw Data

This appendix lists the sensor raw data for Case 1 in 7.1 from Section 7, which was used to generate the frequency spectrum in Figure 7.68, and run the observer offline. The plotted data contain measurements obtained from the logging software, which was interpreted in MATLAB to a viewable format.

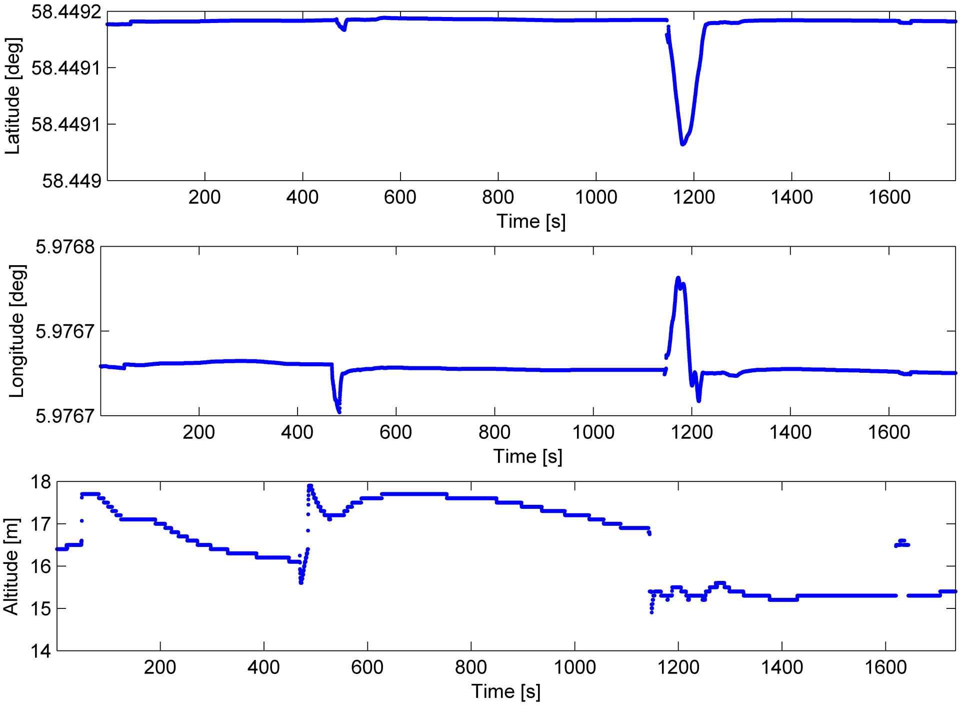


Figure D.1: Raw Data: GNSS

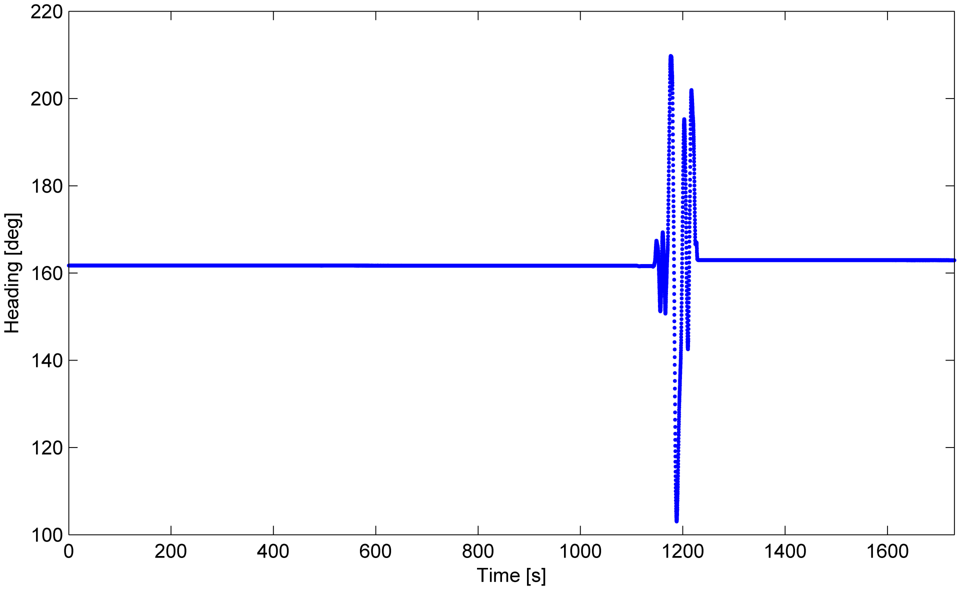


Figure D.2: Raw Data: Gyrocompass (Orion)

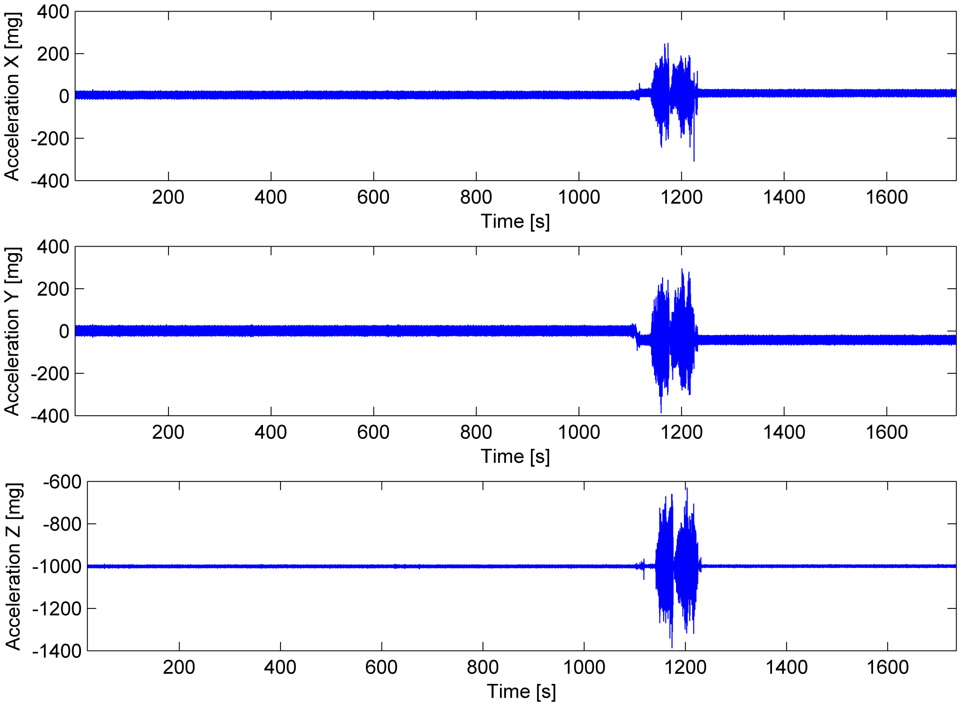


Figure D.3: Raw Data: IMU (Accelerometers)

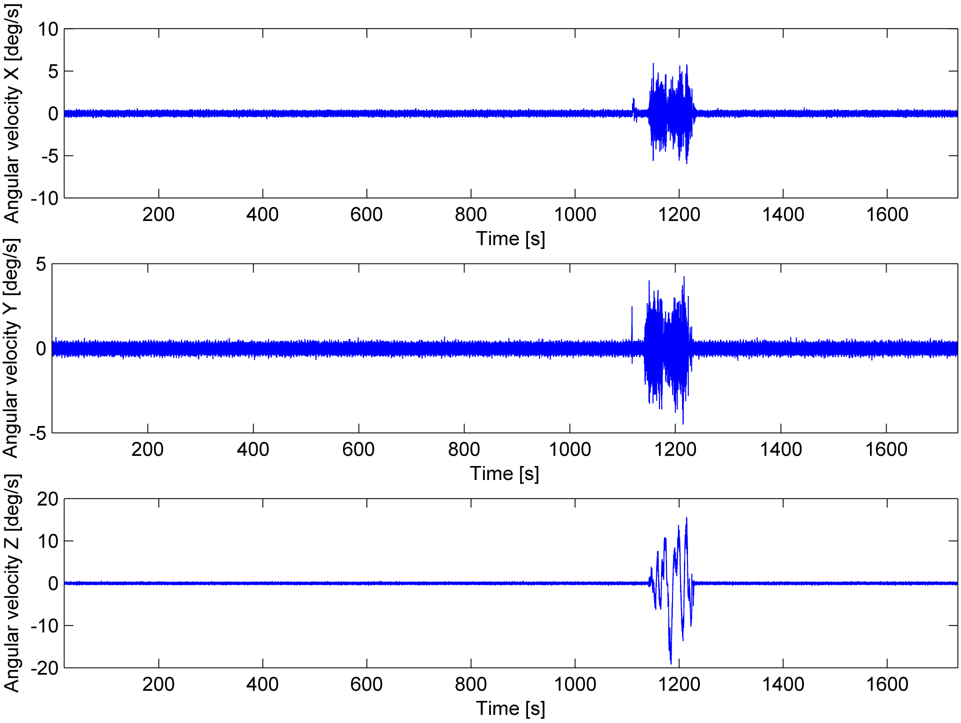


Figure D.4: Raw Data: IMU (Gyroscopes)

Appendix E

Digital Appendix

The following list represents the contents of the digital appendix:

- **Dynamic GNSS Gain:** The MATLAB code for plotting the eigenvalues and infinity norms of the error dynamic matrix.
- **GUI:** The C# code for the graphical user interface.
- **Kalman Filter versus Nonlinear Observer:** The MATLAB and Simulink code for running the state estimator comparison.
- **Logging Data with Processing Tools:** The logging data and MATLAB code for processing the corresponding data.
- **Logging System:** The C++ code for the logging system.
- **Nonlinear Observer C++:** The C++ code for the nonlinear observer implemented in the experimental setup.
- **Simulator Design:** The MATLAB and Simulink code for running the state estimator in the simulator.

Due to the large amount of logging data, only Case 1 in Section 7.1 was included.