Tonja Joseph

State Estimation for a Self-Driving Racing Car

Master's thesis in Cybernetics and Robotics

Supervisor: Torleiv Håland Bryne

June 2020

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Tonja Joseph

State Estimation for a Self-Driving Racing Car

Master's thesis in Cybernetics and Robotics
Supervisor: Torleiv Håland Bryne
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

MSC THESIS DESCRIPTION SHEET

Name: Tonja Joseph
Department: Engineering Cybernetics
Thesis title (Norwegian): Tilstandsestimering for en selvkjørende racerbil
Thesis title (English): State estimation for a self-driving racing car

Thesis Description: The purpose of the thesis is to develop a fault-tolerant state estimator for a self-driving racing car based on GNSS/INS sensor fusion. States include position, velocity and attitude (PVA) and signals derivable from PVA.

The following tasks should be considered:

1. Perform a literature review on PVA determination based on INS and miscellaneous aiding sensors for self-driving cars.
2. Derive and implement an extended Kalman filter (EKF) applicable of estimating position, velocity and attitude (PVA) for the Revolve NTNU racing car based on IMU and dual GNSS/RTK GNSS measurements. GNSS outliers should be handled. Add additional sensors if relevant.
3. If time permits, perform a short literature review on road tire friction estimation and derive and implement a road tire friction estimation algorithm.
4. Test the algorithm(s) in simulations.
5. If time permits, implement the estimators on the Revolve car's on-board computer, perform drive tests and compare the results with the output of the on-board navigation system.
6. Present and discuss the results and error sources.
7. Conclude on your results and suggest future work.

Start date: 2020-01-06
Due date: 2020-06-25

Thesis performed at: Department of Engineering Cybernetics, NTNU
Supervisor: Adjunct associate professor Torleiv H. Bryne, Dept. of Eng. Cybernetics, NTNU

Abstract

An autonomous Formula Student race car requires accurate information about the vehicle's pose and motion in order to complete the challenging dynamic events of the competition as fast as possible. To reach all the way to the top of the ranking list, the race car should display robust performance despite complex tracks and partial system failure. This includes reliable and efficient fusion of the available sensor data, where all sources of uncertainty are optimally accounted for.

In this thesis, a state estimation solution is derived, implemented and validated with simulations. The proposed design is a multiplicative extended Kalman filter. The filter is formulated indirectly and driven by high rate inertial measurements. Low rate position and baseline measurements from dual global navigation satellite system (GNSS) receivers and radar-based ground speed measurements are used for corrections. The proposed design uses a unit quaternion as the nominal attitude parametrization and the three parameter angular error state. This removes any risks of singularities in the system. The Kalman filter is based on a purely kinematic model. Hence, it is independent of the vehicle's parameters and robust to varying track and tire conditions.

The validation of the developed state estimation system shows promising potential. The system shows rapid convergence towards the true vehicle states and is capable of detecting and rejecting outliers in the sensor measurements. The validation is carried out offline with simulated data. This is representative, but not sufficient for verification of on-track performance. Further testing and development still remains before the system can be considered viable. To facilitate further development, the current design is thoroughly examined. The main contributing factors to the desirable performance achieved in the simulator are discussed. In the very end of this thesis, suggestions for future work are proposed based on the experience gained throughout this process.

Keywords: State Estimation, Sensor Fusion, Autonomous Vehicles, Kalman Filters, Aided Inertial Navigation Systems

Sammendrag

En selvkjørende Formula Student-racerbil er avhengig av å ha nøyaktig kunnskap om sin nåverende posisjon, fart og attitude, ofte referert til som bilens *tilstander*, til enhver tid. Dette er nødvendig for å fullføre de dynamiske konkurranseeventene så fort som mulig. For å nå hele veien opp til toppen av resultatlistene, må racerbilen være istand til å vise robust ytelse i alle sine systemer til tross for utfordrende kjørebaner og delvise systemfeil. Her inngår også pålitelig og effektiv fusjon av den tilgjengelige sensordataen, hvor alle kilder til usikkerhet er tatt hensyn til.

Denne masteroppgaven redegjør, implementerer og validerer et tilstandsestimeringssystem for en Formula Student-racerbil som møter de nevnte kravene. Det foreslåtte designet er et multiplikativt utvidet Kalmanfilter. Filteret er indirekte formulert, og drives av høyfrekvente målinger fra treghetssensorer. Posisjonsmålinger og basevektormålinger fra to GNSS-mottakere, samt radarbaserte bakkehastighetsmålinger, er brukt til å korrigere tilstandsestimatene. I et slikt design brukes kvarternioner til å parametrisere den nominelle attituden til fartøyet, derav betegnelsen "mulitplikativt". Dette eliminerer risikoen for singulariteter i systemet. Attitudefeilen er et såpass lite signal at den uansett ikke er utsatt for singulariteter og er representert med den tredimensjonale vinkelfeilen. Kalmanfilteret er også basert på en kinematisk modell, og er dermed uavhengig av parametrene til bilen. Dette gjør at tilstandsestimeringssystemet også er robust mot varierende dekk- og bakkeforhold.

Valideringen av det utviklede tilstandsestimeringssystemet viser et lovende potensiale. De estimerte tilstandende konvergerer raskt mot deres sanne verdier, gitt av en simulator. Det utviklede systemet er også istand til å detektere og forkaste ekstremverdier, ansett som upålitelige, basert på statistisk hypotesetesting. Valideringen er gjort i simulator. Dette er representativt, men ikke tilstrekkelig for å kunne si noe sikkert om ytelsen under et faktisk løp. Det gjenstår derfor forsatt validering og videreutvikling før det utviklede tilstandsestimeringssytemet kan anvendes på bil. For å legge til rette for denne videreutviklingen, er det nåværende designet nøye drøftet i denne rapporten. De største bidragsytende faktorene til de tilstrekkelig gode resultatene oppnådd i simulator er identifisert og diskutert. Avslutningsvis inneholder oppgaven konkrete forslag til hensiktsmessige fremtidige fokusområder i videreutviklingen av produktet.

Preface

This master thesis concludes the final evaluation of the five year long Master of Science program provided by the Department of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU). The work amounts to a full semester, corresponding to 30 credits. This thesis is not a continuation of a specialization project.

The thesis is carried out on behalf of Revolve NTNU, with the task of developing a state estimation system for a driverless Formula Student race car. With a deep interest in applied control theory and autonomous vehicle technology, it has been a privilege to take part in the fast-paced and educated environment of Revolve NTNU.

For the reader, knowledge of basic control theory and Kalman filtering is convenient for understanding the content of this thesis. It is also useful to be familiar with classical kinematics and the representation of dynamic systems in terms of ordinary differential equations.

I would like to show a special gratitude to my supervisor, Torleiv H. Bryne, for assistance and feedback throughout the semester, facilitating a major learning outcome. The general sharing of knowledge and experience, both with regards to the system development and the writing of this thesis, has been highly appreciated. I would also like to acknowledge Henning Ward for sharing useful advise in the initial steps of this project. Moreover, I am grateful towards the entire organization of Revolve NTNU. A special thanks goes to the driverless team and the members of the Vehicle Dynamics and Control Systems (VDCS) group for academic discussions and encouragement towards the project. Lastly, I would like to extend my sincere thanks to my family for always being there and showing unparalleled support.

Tonja Joseph
Trondheim, June 2020

Contents

List of Figures

List of Tables

Abbreviations

ABS anti-lock braking system.

AUV autonomous underwater vehicle.

BPP Business Plan Presentation.

CG center of gravity.

DoD U.S Department of Defense.

EKF extended Kalman filter.

ESC electronic stability control system.

ESKF error state Kalman filter.

FFI Norwegian Defence Research Establishment.

FS Formula Student.

FSG Formula Student Germany.

GM General Motors.

GNC guidance, navigation and control.

GNSS global navigation satellite system.

GPS Global Positioning System.

GSS ground speed sensor.

IEKF iterative extended Kalman filter.

IMU inertial measurement unit.

INS inertial navigation system.

KF Kalman filter.

LiDAR Light Detection and Ranging.

MEKF multiplicative extended Kalman filter.

MEMS micro-electro-mechanical systems.

MPC model predictive control.

MSS Marine Systems Simulator.

NED North East Down.

NTNU the Norwegian University of Science and Technology.

PSD power spectral density.

PVA position, velocity and attitude.

RMS root mean square.

SAE the Society of Automotive Engineers.

SLAM simultaneous localization and mapping.

SSKF steady state approximation of the extended Kalman filter.

SWPS Steering Wheel Position Sensor.

TUM the Technical University of Munich.

VDCS Vehicle Dynamics and Control Systems.

Nomenclature

T_{acc} Time constant of the accelerometer bias model.

T_{ars} Time constant of the gyrsocope bias model.

$\Delta\boldsymbol{\omega}_{nb}^{b}$ Gyroscope error model.

$\mathbf{0}_3$ 3x3 zero matrix.

$\boldsymbol{A}_d$ Discrete Jacobian matrix of the error state dynamics wrt. the error state.

$\boldsymbol{A}$ Continuous Jacobian matrix of the error state dynamics wrt. the error state.

$\boldsymbol{E}_d$ Discrete Jacobian matrix of the error state dynamics wrt. the noise.

$\boldsymbol{E}$ Continuous Jacobian matrix of the error state dynamics wrt. the noise.

$\boldsymbol{H}$ Measurement Jacobian matrix.

$\boldsymbol{I}_3$ 3x3 identity matrix.

$\boldsymbol{K}_k$ Kalman gain matrix at timestep k.

$\boldsymbol{Q}_{acc}$ Process noise covariance matrix of the accelerometer.

$\boldsymbol{Q}_{ars}$ Process noise covariance matrix of the gyroscope.

$\boldsymbol{R}_{acc}$ Measurement noise covariance matrix of the accelerometer.

$\boldsymbol{R}_{ars}$ Measurement noise covariance matrix of the gyroscope.

$\boldsymbol{\delta}(\tau)$ Continuous Dirac delta function.

$\boldsymbol{\omega}_{imu}^{b}$ Angular rate measurement from the gyroscope.

$\boldsymbol{\omega}_{nb}^{b}$ True angular rate.

$\boldsymbol{b}_{acc}^{b}$ True accelerometer bias.

$\boldsymbol{b}_{ars}^{b}$ True gyroscope bias.

$\boldsymbol{f}_{imu}^{b}$ Specific force measurement from the accelerometer.

$\boldsymbol{f}_{nb}^{b}$ True specific force.

$\boldsymbol{g}_{nb}^{n}$ True gravity.

$\boldsymbol{p}_{nb}^{n}$ True position.

$\boldsymbol{q}_{nb}^{n}$ True attitude quaternion.

$\boldsymbol{v}_{nb}^{n}$ True velocity.

$\boldsymbol{w}_{acc}^{b}$ Accelerometer measurement noise.

$\boldsymbol{w}_{ars}^{b}$ Gyroscope measurement noise.

$\boldsymbol{x}$ True state vector.

$\delta\boldsymbol{\theta}_{nb}^{b}$ Three parameter angular error state.

$\delta\boldsymbol{b}_{acc}^{b}$ Accelerometer bias error state.

$\delta\boldsymbol{b}_{ars}^{b}$ Gyroscope bias error state.

$\delta\boldsymbol{g}_{nb}^{n}$ Gravity error state.

$\delta\boldsymbol{p}_{nb}^{n}$ Position error state.

$\delta\boldsymbol{q}_{nb}^{n}$ Quaternion representation of the attitude error state.

$\delta\boldsymbol{v}_{nb}^{n}$ Velocity error state.

$\delta\boldsymbol{x}_{k}^{+}$ A posteriori error state estimate at timestep k.

$\delta\boldsymbol{x}_{k}^{-}$ A priori error state estimate at timestep k.

$\delta\boldsymbol{x}$ Error state vector estimated by the ESKF.

$\delta\boldsymbol{y}$ Error between the low rate measurements and the corresponding quantities estimated within the nominal state space.

$\hat{\boldsymbol{P}}_{k}^{+}$ A posteriori estimate of the error state covariance matrix at timestep k.

$\hat{\boldsymbol{P}}_{k}^{-}$ A priori estimate of the error state covariance matrix at timestep k.

$\hat{\boldsymbol{b}}_{acc,ins}^{b}$ Nominal accelerometer bias.

$\hat{\boldsymbol{b}}_{ars,ins}^{b}$ Nominal gyroscope bias.

$\hat{\boldsymbol{g}}_{ins}^{n}$ Nominal gravity.

$\hat{\boldsymbol{p}}_{ins}^{n}$ Nominal position.

$\hat{\boldsymbol{q}}_{ins}^{n}$ Nominal attitude quaternion.

$\hat{\boldsymbol{v}}_{ins}^{n}$ Nominal velocity.

$\hat{\boldsymbol{x}}_{ins}$ Nominal state vector estimated by the INS.

h Time step in seconds.

$\{b\}$ BODY coordinate frame.

$\{n\}$ North East Down coordinate frame.

1 Introduction

A central task in the field of control engineering is the manipulation of physical systems to achieve a desired behaviour. To achieve a certain behaviour, the control engineer must design a control law that specifies the correct set points for actuator inputs at any point in time. Such a control law typically depends on a set of states and a strategy known as feedback control, where knowledge about the states are used to determine actuator inputs which again act on the states.

A major obstacle in control applications is to accurately obtain information about the current state of a system. To enable observation of the states of interest, the system is equipped with various measurement units. However, a general challenge is restrictions in the insight into all desired states of a system. To alleviate this problem a common approach is to construct *estimates* of the desired states based on the available information.

When a control law is dependent on estimates of a system's states, a crucial question is how these estimates should be constructed optimally. An available measurement is often described by a measurement function. The map between the measurement and the states, represented by this function, is in general not injective. Thus, the estimates cannot be a static function of the measurements. Instead, they are designed as the output of a dynamic system. This yields an error term between the estimates and the true states, where concepts of control theory can be utilized to achieve the desirable behaviour of convergence towards zero.

1.1 Autonomous vehicle technology

Vehicles are a widely studied control application and represent a platform for application of the latest within autonomous technology. An autonomous vehicle is defined as a vehicle capable of sensing its environment and operating without human involvement (Synopsys, 2020). The Society of Automotive Engineers (SAE) currently defines six levels of driving automation ranging from level 0, where the vehicle is fully manual, to level 5, where the vehicle is fully autonomous. The levels in between capture vehicles with some degree of automation, collectively known as semi-autonomous vehicles.

At present, the majority of vehicles on the road can be described as semi-autonomous due to advanced safety features like assisted braking and parking system. Modern cars are for instance often equipped with an anti-lock braking system (ABS), preventing the wheels from locking during hard braking, and an electronic stability control system (ESC), stabilizing the lateral motion in order to prevent skidding. Other advanced safety features include collision warning and avoidance, rollover

Figure 1.1: Devbot 2.0 competing in Season Alpha

prevention and crosswind compensation (Consumer Reports, 2020).

The concept of fully autonomous vehicles dates all the way back to 1939, when General Motors (GM) created an exhibit displaying their vision of an automated highway system guiding self-driven cars (Gringer, 2020). From here, continuous work towards this vision has been carried out, both in academia and in industry. Speedy and surprising progress has been made, yielding a number of great achievements. A memorable milestone is Google's self-driven car, tested on public roads since 2010 (Woollaston, 2020).

1.2 Autonomous racing

Autonomous *racing* has relatively recently gained a lot of attention and is a rapidly increasing technology. It refers to the sport of racing ground-based vehicles, controlled by computers. In early 2016, Roborace started working on the world's first driverless race car, named "Devbot". In August the same year, it successfully drove twelve laps of a Formula E circuit. Roborace is also the company starting Season Alpha, the world's first motor-sports series for self-driven cars. Autonomous racing pushes the limits of autonomous vehicle technology, making it an exciting engineering challenge.

1.3 Problem formulation

This thesis is carried out in collaboration with Revolve NTNU, a student engineering team developing an autonomous race car. With this race car, the team com-

petes against other university teams in the highly established Formula Student (FS) competition. The competition is arranged and supported by major players in the automotive industry (Revolve, 2020). The task at hand is to develop an accurate and robust state estimation system, utilizing the available sensor configuration and applying concepts of control theory.

All autonomous vehicle technology rely on information about the current state of the vehicle and its surroundings. To retrieve this information modern vehicles are equipped with advanced sensing systems. Fusion of these sensors, taking parameter uncertainties and other error sources into account, is currently a popular topic of research and development. As mentioned, autonomous race cars are intended to push the limits of speed and rapid maneuvers. This increases the demands on the autonomous systems. Currently, autonomous racing is a topic poorly covered in literature, and most work in the field is carried out by students. It is therefore rewarding to take part in this field of research.

1.4 Main contributions

The main contributions of this thesis are:

- Derivation of models for the system dynamics and measurements for an aided inertial navigation system on an autonomous Formula Student race car.
- Implementation of a multiplicative extended Kalman filter (MEKF) with an indirect formulation, combining high rate inertial sensor data with low rate dual GNSS position measurements along with ground speed measurements for accurate state estimation.
- Modifications of the state estimation system design at stand-still with regards to the target application for increased performance.
- Detection and rejection of outliers in the sensor data using a probabilistic χ_2-test.
- Validation of the developed navigation system with simulations, and discussion of its strengths and weaknesses with regards to the target application.

1.5 Thesis outline

This thesis aims to systematically present the work carried out in this project. The report is organized in seven chapters with distinct purpose. In the following, a brief overview of the thesis organization is given.

Chapter 2: Background and Motivation

Chapter 2 contains information about the background and motivational factors for this project. More information about Revolve NTNU and the FS competitions is provided, disclosing the bigger picture and substantiating focus areas during the project. This chapter also mentions some challenges due to the COVID-19 outbreak.

Chapter 3: Preliminaries

In Chapter 3, the preliminary knowledge necessary for understanding the developed navigation solution is put forward. The chapter contains a short introduction to navigation before it moves to a more thorough explanation of the concepts relevant for the system developed here. The various design choices made in this project are also substantiated in this chapter.

Chapter 4: The State Estimation System

Chapter 4 systematically presents the final implementation of the state estimation system. All equations for the various operations carried out in the developed system are stated and explained. In other words, this chapter contains the realization of the concepts presented in Chapter 3.

Chapter 5: Results and Discussion

This chapter aims to validate the final performance of the developed state estimation system. The achieved results are also discussed. After this, small modifications are applied to the final design in order to examine essential design choices. In the last part of this chapter, individual sensors are removed from the sensor configuration to investigate their contribution to the final performance.

Chapter 6: Conclusion

In Chapter 6, the work and findings of this thesis are summarized.

Chapter 7: Future Work

The final chapter proposes suggestions for future work, with the aim of facilitating further development of this state estimation system.

2 *Background and Motivation*

This chapter provides information about the background and motivational factors behind this project. The aim of this chapter is to illustrate the bigger picture and to present the most important contextual information.

2.1 Revolve NTNU

Revolve NTNU is an independent student organization at NTNU. This year's team consists of approximately 80 members from 24 different fields of study working voluntarily to design and build a self-developed race car in only eight months. This is carried out in parallel with full time studies (Revolve, 2020).

2.2 Formula Student

Every summer Revolve NTNU competes in Europe's most established educational engineering competition, Formula Student (FS), against university teams from all over the world (IMechE, 2020). Multiple FS competitions are organized, in several different countries, where Formula Student Germany (FSG) is recognized as the most prestigious.

Formula Student provides students the opportunity to push their engineering design and project management skills through application of classroom theory in the demanding real world. With heavy industry backing, it has become a well respected initiative, developing students into desirable engineers (Revolve, 2020). All the FS competitions are organized similarly, consisting of both static and dynamic events. The static events focus on cost, business plan and engineering design, while the dynamic events focus on the on-track performance of the race car.

2.3 Formula Student Driverless

The teams can choose to compete in one out of three classes: the combustion class, the electric class or the rather new driverless class, introduced in 2017. Revolve NTNU has since 2018 competed both in the electric and in the driverless class. The members are divided into two dedicated divisions, focusing on one competition class each. This thesis is carried out in collaboration with the driverless team of Revolve NTNU.

The driverless class sticks out from the other two in multiple ways. The competition task in this class is no longer to build a race car from scratch, but rather to adopt

a previously built FS car and transform it into an autonomous race car by adding appropriate sensors and software. To illustrate how the quality and performance of the race car will be measured, some essential aspects of the driverless competition is provided below.

Among the static events, the event named "Engineering Design" is especially important in the driverless class. The task to be solved in this competition require advanced sensing and software systems utilizing recently developed concepts, where many of these concepts are still subject of research. In addition, safety is a crucial topic. All design choices should therefore be carefully considered when developing the autonomous systems, and properly defended at the Engineering design event.

In the dynamic events the car has to drive tracks of varying length and complexity as fast as possible, and fully autonomously, without hitting any delimiters. The delimiters are cones placed on both sides of the track with even spacing and color coding to signal left or right side. To perform well in the dynamic events the autonomous race car is dependant on robust, reliable and precise state estimation, detection, localization and mapping - among other things.

2.4 The car: Atmos

For the 2020 season, the driverless team of Revolve NTNU has chosen to compete with the electrical race car named Atmos. Atmos is the third generation four-wheel driven electrical race car of Revolve, developed and produced by the team of 2018 (Revolve, 2020). The efforts of this team is rewarded with a second place in FSG 2018, proving great capabilities. Atmos was also used by the driverless team of 2019. According to the competition rules, it is allowed to compete in the driverless class with the same vehicle up to two times.

Atmos is configured with a large number of sensors ranging from simple voltage meters on every other battery cell to complex high-technology sensors such as the two LiDAR sensors. The most relevant sensors for the autonomous system on Atmos are presented more in detail in Appendix A.

Figure 2.1: The driverless vehicle of team 2019 and 2020, named Atmos.

2.5 The autonomous software system

The autonomous software system of Atmos is referred to as "the autonomous pipeline" within the team. Key components of this pipeline are state estimation, redundant cone detection systems, simultaneous localization and mapping (SLAM), trajectory planning and model predictive control (MPC). The overall task solved by the autonomous pipeline is to detect and localize the various types of cones in its environment based on sensor data, place the cones in a map and place itself in the same map. The autonomous software of the car should then, based on this knowledge, generate the optimal path and determine the optimal actuator set points to complete the racetrack as fast as possible.

The competition rules provide information about what the colours and sizes of the cones signalize. This facilitates understanding of the racetrack layout based on the detected cones. The autonomous pipeline should handle obstacles such as noise on sensor data, measurement delays, false positives and negatives, limitations in sight of the various sensors, environmental disturbances and general mechanical and electrical constraints.

2.6 Performance goals of Revolve NTNU Driverless

Goals are important as they outline the expected outcome of the work carried out by the team. They set clear performance standards for the team members and guide their efforts. A common set of goals also has a unifying effect and facilitates team spirit and motivation. Due to this, Revolve NTNU Driverless has set some

specific and measurable goals for the 2020 season. These are listed below.

Overall goal

- Rank Top 3 overall at every competition Revolve NTNU attends

Sub-goals

- Test ready car by the 13th of April
- Top 3 in all dynamic events
- Top 5 in Engineering Design

Dynamic goals

- 3.75 sec on Acceleration, driving 75 meters straight
- 5.5 sec on SkidPad, driving an eight-shaped track
- 10 m/s avg. on AutoX, driving a more complex track without prior knowledge about the layout
- 17 m/s avg. on TrackDrive, driving the AutoX track 10 times

Static goals

- 70/100 points on Cost
- 70/75 points on Business Plan Presentation (BPP)
- 230/300 points on Engineering Design

2.7 Aim of this project

The project subject to this thesis aims to develop a robust and reliable state estimation system for the driverless vehicle of Revolve NTNU that contributes to the achievement of the performance goals presented above. The state estimation system should provide reliable information about the current states of the vehicle at any point in time. By doing so, it should facilitate high performance of the other interconnected subsystems of the vehicle's autonomous software. The state estimation system should be designed with the specific application in mind, and show decent capabilities in precision, convergence speed and fault handling despite the expected high speeds and rapid maneuvers.

2.8 Remarks regarding the COVID-19 situation

This project has been affected by the unexpected and unfortunate COVID-19 outbreak. From the initial awareness of the virus in January 2020, a large spread has taken place resulting in a global state of emergency. Due to this pandemic, a series of measures has been introduced by governments in numerous countries with the aim of slowing the transmission of the virus.

In Norway, comprehensive measures to facilitate social distancing is implemented. All at risk of carrying the virus is required to stay in home quarantine or home isolation depending on the risk factors. Educational institutions, meeting venues and a number of businesses are closed, and it is highly encouraged to avoid public spaces and public transport (NIPH, 2020).

For Revolve NTNU the COVID-19 situation caused loss of access to offices, workshops, and production and testing spaces lend by various sponsors. The project was hence put on hold. The COVID-19 pandemic has reached a level of criticality beyond what was expected. 22nd of March 2020, it was announced that FSG 2020 is cancelled. Shortly after this disappointing yet unsurprising announcement, the other FS competitions followed. Consequently, the decision was taken by Revolve NTNU to cancel this year's project.

The original purpose and contribution of this thesis is highly affected by the unfortunate situation. However, it has been possible to validate the system to a certain degree through simulations. The findings of the work carried out here is also still relevant for future teams of Revolve and is deemed purposeful.

3 Preliminaries

In this chapter, the preliminaries of the developed state estimation system is presented. The chapter captures the essential knowledge gained through literature studies in the initial phase of the project. All concepts used in the final state estimation solution are explained and substantiated, providing the reader with the necessary prerequisites for understanding the final implementation.

3.1 Introduction to navigation

Navigation capability is fundamental for an autonomous vehicle. In literature regarding autonomous vehicles, the term navigation may refer to two different tasks (Farrell, 2008).

1. Accurate determination of vehicle states, where common states of interest are position, velocity and attitude.
2. Planning and execution of the maneuvers necessary to move between desired locations.

The first capability is necessary to achieve the second. In this report, the term navigation refers to the first of the tasks listed above. The second task is rather referred to as a combination of planning, guidance and control. The navigation task, as it is defined here, is also commonly described as *state estimation* when the problem is resolved based on a defined state space for the vehicle system. Navigation systems can in other words be regarded as a subcategory of state estimation systems, restricted to the specific application of vehicles.

The acknowledged state-space model format shows that a system may have many internal variables or *states*, $\mathbf{x}$, and fewer outputs, $\mathbf{y}$. This is due to a variety of reasons, for instance lack of appropriate sensors. From a more theoretical perspective, state estimation refers to the problem of achieving knowledge about all states in the state vector, when only a subset of the states are directly measured (Farrell, 2008).

State estimation is useful in control applications where knowledge of the entire system state would allow higher performance of the total control system acting on a physical system. Figure 3.1 shows the structure of a general guidance, navigation and control (GNC) system of a plant, for instance a vehicle. The grey blocks represent various subsystems of the GNC system. Together, these blocks are responsible for both of the tasks listed above. The blue block represents the vehicle, equipped with sensors and actuators.

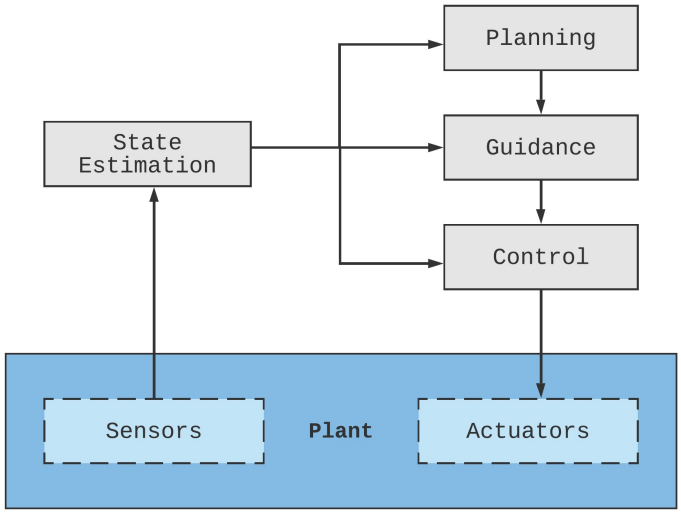


Figure 3.1: General structure of a GNC system

As Figure 3.1 shows, the state estimation system represents the bridge between the sensors, monitoring the vehicle, and the rest of the GNC system. It is thus a highly interconnected component of the total autonomous software system.

3.2 Aided inertial navigation systems

An inertial navigation system (INS) estimates the position, velocity and attitude (PVA) of a vehicle based on inertial sensor measurements. An inertial measurement unit (IMU) commonly consists of two types of measuring instruments. These are gyroscopes and accelerometers. Gyroscopes measure the angular rates of the vehicle, including the Earth's rotation, and the accelerometers measure the specific forces, which represent the sum of the accelerations and the sensed gravity.

The integrative nature of an INS is advantageous in the sense that it smooths out high-frequency errors such as sensor noise. On the negative side, integration of low frequency errors due to biases or misalignment will cause the estimates to drift further and further away from the true state. It is therefore very common to correct estimates arriving from an INS with external aiding sensors (Farrell, 2008). Navigation systems combining high rate inertial sensors with low rate aiding sensors to perform state estimation are referred to as *aided* navigation systems. By selecting suitable low rate aiding sensors the expected drifting in the INS estimates can be eliminated.

Table 3.1: Comparison of the properties of GNSS and INS

GNSS	INS
- Low bandwidth	+ High bandwidth
+ Bounded errors	- Unbounded errors
+ Good long-term accuracy	- Poor long-term accuracy
- Relies on external information [1]	+ Self-contained
- Susceptible for obstruction, jamming and spoofing	+ Non-jammable and non-spoofable
- No attitude estimation[2]	+ Attitude estimation

A common way to design aided navigation systems is to fuse an INS with a navigation solution based on global navigation satellite systems (GNSS). Table 3.1 show some of the main advantages and disadvantages of the navigation solutions provided by each measurement source. As seen in Table 3.1, the properties of the INS and the GNSS are complementary. It is therefore reasonable to combine them, utilizing the advantages of each solution to obtain a reliable estimate of the vehicle states (Bryne and Fossen, 2019). It is also common to use more than one aiding sensor to increase robustness. On Atmos, an aided inertial navigation system will be developed utilizing dual GNSS position measurements, the baseline vector between the two GNSS antennas and velocity measurements from a radar-based ground speed sensor (GSS) as the low rate aids.

3.3 Coordinate frames

In the design process of a navigation system there are several coordinate or reference frames in which the available measurements and computed quantities can be expressed. It is important to have knowledge about the reference frame of the various quantities in order to correctly relate them to each other. The reference frames relevant for this navigation system are described in the following.

3.3.1 Nomenclature

Before proceeding, the notational convention used in this thesis will be specified. The reference frame in which the vector is represented is indicated by a superscript. In addition, the subscript tells something about the way the current quantity describes the behaviour of one frame with respect to another frame. To exemplify, $\boldsymbol{v}_{nb}^{n}$ denotes the velocity of the BODY frame with respect to the NED frame, due to the subscript, given in NED coordinates, due to the superscript. The

[1] Satellite position and velocity
[2] Heading can be estimated with a dual receiver solution

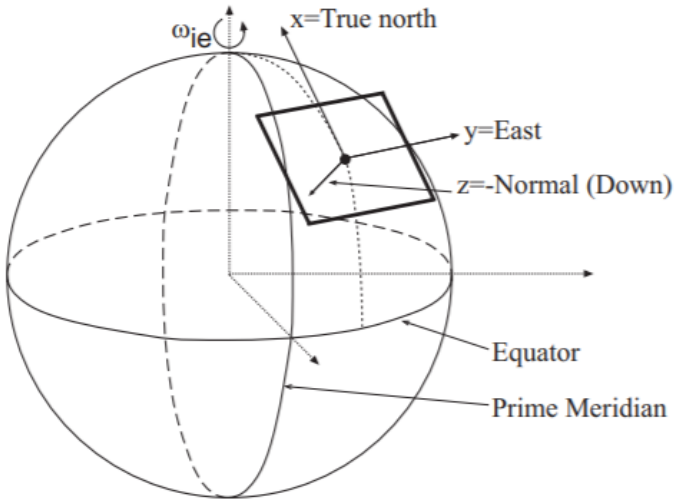


Figure 3.2: Illustration of how the NED reference frame, denoted $\{n\}$,is defined.

subscript convention for the reference frame is sometimes omitted. This is to avoid unnecessarily long subscripts, when the information is considered evident.

The rotation matrix between two frames are denoted $\boldsymbol{R}$ with a subscript indicating the original reference frame and a superscript indicating the destination reference frame. For example, $\boldsymbol{R}_b^n$ represent the rotational transformation from BODY to NED such that (3.1) holds.

$$\boldsymbol{v}_{nb}^n = \boldsymbol{R}_b^n \boldsymbol{v}_{nb}^b \tag{3.1}$$

3.3.2 North east down frame

The NED frame, denoted $\{n\}$, is the north, east, down rectangular coordinate system often referred to in the everyday life. It is determined by fitting a tangent plane to a fixed point on Earth's surface, as shown in Figure 3.2. The fixed point is the origin of the local frame chosen such that it is convenient for local measurements. The x-axis of the NED frame points to true north along the tangent plane, the z-axis point towards the interior of the Earth, perpendicular to the reference ellipsoid[3] and finally the y-axis completes the right-handed coordinate system pointing east (Farrell, 2008).

[3]The reference ellipsoid is a mathematically defined surface approximating the geoid which is the truer shape of the Earth

The NED frame is applicable to local navigation applications constrained to a smaller geographic area, where the assumption of a flat Earth is deemed reasonable (Bryne and Fossen, 2019). In the aircraft and marine craft control system literature, the Earth's rotation is often neglected, such that the NED frame has a local definition where it is non-rotating and assumed inertial (Perez and Fossen, 2011). It is reasonable to make this assumption when studying motion with very high velocities compared to the Earth's rotation. This is true for the target vehicle of the state estimation system developed here, and the NED frame is therefore assumed inertial here as well.

3.3.3 BODY frame

The BODY frame, denoted $\{b\}$, is a rigid frame attached to the navigating vehicle. Its origin is commonly at the center of gravity (CG) of the vehicle, with the x-axis pointing forward, the z-axis pointing downwards and the y-axis pointing to the right side such that the right-handed coordinate system is complete. Inertial sensors such as accelerometers and gyroscopes commonly provide measurements in this frame (Bryne and Fossen, 2019).

3.4 Sensor fusion using Kalman filtering

Sensor fusion is the task of optimally combining the information provided by various sensors, for instance on a vehicle. One of the tools widely used for this purpose is the Kalman filter (KF) (Kalman, 1960). The linear KF is recognized as the state-of-the-art, optimal state estimator for linear dynamic systems and has been so for several decades.

3.4.1 The principle of linear Kalman filtering

The equations of the linear Kalman filter will not be derived here. Instead, it is attempted to break down the concept and provide an intuitive explanation of how this method works, similar to what is done by Plett (2004) in his report about Kalman filtering for battery management systems.

Very generally, any causal dynamic system generates its outputs as some function of past and present inputs, where the states of this system summarizes the effect of all past inputs. The present system output can hence be computed from the present input and the present state, and there is no need for storing past input values.

The dynamics of a linear system can be described by the discrete-time state space

model:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k + \boldsymbol{w}_k \tag{3.2a}$$

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{3.2b}$$

With the following nomenclature

$\boldsymbol{x}_k$	State vector at time instance k
$\boldsymbol{u}_k$	The known input to the system at time instance k
$\boldsymbol{y}_k$	The output or measurement vector at time instance k.
$\boldsymbol{w}_k, \boldsymbol{v}_k$	Process noise and measurement noise respectively, assumed mutually uncorrelated white Gaussian random processes
$\boldsymbol{A}_k, \boldsymbol{B}_k, \boldsymbol{H}_k$	Matrices describing the linear dynamics of the system

Equation (3.2a) is called the process equation. This equation captures the evolving system dynamics, and the system stability, controllability and sensitivity to disturbance may all be derived from this equation. Equation (3.2b) is referred to as the output or measurement equation. Given a system on this form, it may be desired to estimate the unmeasured dynamic state, $\boldsymbol{x}_k$, in real-time, using the knowledge about the system's measured input and output signals. The KF is the optimal way to achieve this estimate under certain assumptions. Here, the assumption of uncorrelated Gaussian white noise processes is deemed the most essential.

The Kalman filter problem can be formulated as follows: Using the entire observed data $\{\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_k\}$ and $\{\boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_k\}$, find the estimate, $\hat{\boldsymbol{x}}_k$, of the true state, $\boldsymbol{x}_k$, that minimizes the mean squared error. Mathematically the same problem can be

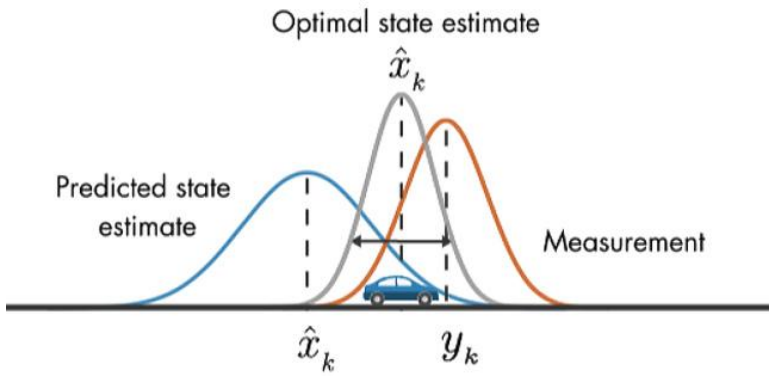


Figure 3.3: The Kalman filter combines knowledge about the dynamics of an internal state and external measurements

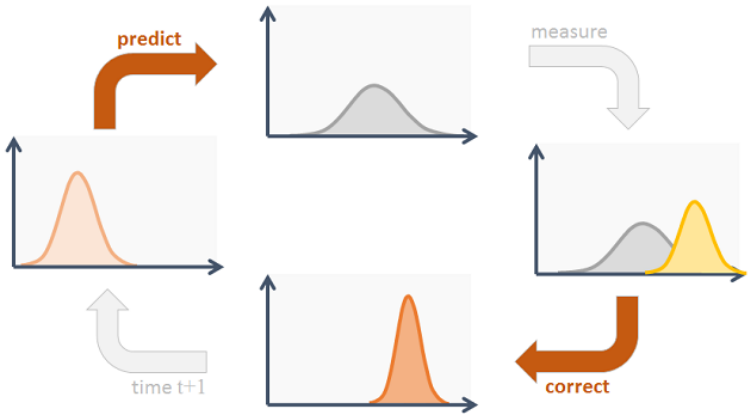


Figure 3.4: Illustration of the recursive Kalman filter steps

formulated as

$$\hat{\boldsymbol{x}}_k = \min(E\{[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k]^T[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k] \mid \boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_k, \boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_k\}) \tag{3.3}$$

The solution to this problem is given by the renowned KF equations, provided in Table 3.2. $\hat{\boldsymbol{P}}_k$ denotes the estimated state covariance matrix, and $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ denote the discrete-time covariance matrices of the process noise and the measurement noise respectively.

As seen in Table 3.2, the Kalman filter computes two different estimates of the state vector and the state covariance matrix in each sampling. The first estimates, $\hat{\boldsymbol{x}}_k^-$, is the propagation in time of the previous state estimate, using the model for the system dynamics. This is commonly referred to as the *prediction step*. In the second estimate, $\hat{\boldsymbol{x}}_k^+$, the measurements of the system output are taken into account and used to adjust the first estimates. This is commonly called the *correction step*. The state covariance matrix is estimated correspondingly.

The correction step represents when the estimator considers "new information" from the measurements. A question arising at this point is to what degree the new information from the measurements should be trusted compared to the modelled system dynamics. To resolve this problem, the Kalman gain, $\boldsymbol{K}_k$, is computed. The Kalman gain represents the optimal weighting of the new information from the measurements, based on the state and measurement covariances.

In conclusion, the Kalman filter provides a theoretically elegant and time-proven method for achieving an optimal estimate of the internal states of a dynamic system (Plett, 2004). It is a powerful sensor fusion tool taking advantage of redundancy by letting the contribution of each sensor input to the overall estimated state be a

Table 3.2: Linear Kalman filter equations

Linear state space model	
	$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k + \boldsymbol{w}_k$
	$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k$
Initialization	
	For $k = 0$, *set*
	$\hat{\boldsymbol{x}}_0^+ = E\{\boldsymbol{x}_0\}$
	$\hat{\boldsymbol{P}}_0^+ = E\{(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)^T\}$
Computation	
	For k = 1, 2, ..., compute
	State estimate time update:
	$\hat{\boldsymbol{x}}_k^- = \boldsymbol{A}_{k-1} \hat{\boldsymbol{x}}_{k-1}^+ + \boldsymbol{B}_{k-1} \boldsymbol{u}_{k-1}$
	Error covariance time update:
	$\hat{\boldsymbol{P}}_k^- = \boldsymbol{A}_{k-1} \hat{\boldsymbol{P}}_{k-1}^+ \boldsymbol{A}_{k-1}^T + \boldsymbol{Q}_{k-1}$
	Kalman gain:
	$\boldsymbol{K}_k = \hat{\boldsymbol{P}}_k^- \boldsymbol{H}_k^T (\boldsymbol{H}_k \hat{\boldsymbol{P}}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1}$
	State estimate measurement update:
	$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k (\boldsymbol{y}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^-)$
	Error covariance measurement update:
	$\hat{\boldsymbol{P}}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \hat{\boldsymbol{P}}_k^- (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)^T + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^T$

function of the sensor's accuracy and previous state knowledge. It estimates how the system develops, propagating not only its state but also the covariance related to each state (Solà, 2017).

3.4.2 Extended Kalman filter for nonlinear systems

The Kalman filter is specially designed for linear systems. However, many physical dynamic systems are more accurately described by nonlinear models. For nonlinear systems a common approach is to include a linearization process in the beginning of each time step. By doing this, a linear approximation of the system dynamics with local validity is achieved. This approach is known as the extended Kalman filter (EKF). Since an approximation of the system dynamics is utilized in this approach, the EKF may not necessarily be the optimal states estimator anymore. It has however proven to work well for many nonlinear system, and

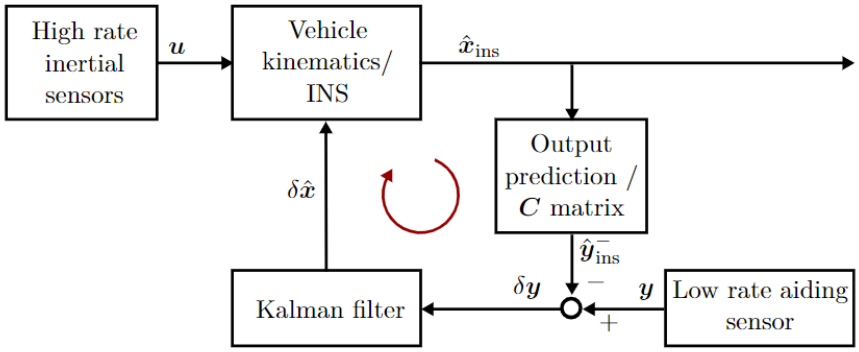


Figure 3.5: Structure of an ESKF

is recognized as a good approximation of the optimal solution (Valls et al., 2018; Plett, 2004).

3.5 Full state vs. error state Kalman filter

Kalman filters are widely used to combine the various sensors in an aided INS. In literature regarding aided inertial navigation, two classical principles for aided INS navigation filters are proposed:

1. Full state Kalman filter
2. Error state Kalman filter (ESKF)

The full state Kalman filter estimates the vehicle's full state, in line with the explanation in the previous section. This is the traditional design. In the error state formulation, the filter tries to estimate the error between the states measured by the low rate aiding sensors, and the corresponding estimates of the states obtained from the IMU integration (Bryne and Fossen, 2019). These errors, often referred to as the *error states*, are then added to the INS estimates to eliminate the drift that has occurred since the arrival of the previous low rate measurement. Figure 3.5 shows a common structure of an ESKF.

3.5.1 Advantages of the error state formulation

The ESKF is often preferred over the full state formulation, due to some key advantages. These are listed below. The list discloses some central differences between the two principles, and is based on the comparison carried out by Solà (2017).

- The ESKF is running in a feedback loop, as shown in Figure 3.5. It is therefore not directly a part of the control loop, like the full state system. This allows the navigation system to operate even if the KF block fails. This is because the integrator block connected to the main set of sensors is still capable of computing estimates. It is only the correction from the low rate sensors that is lost.
- The error state system always operates close to the origin, and thus far from possible parameter singularities (e.g. the Euler angle singularity of ± 90 degrees in pitch). This also strengthens the validity of the linearization.
- Since the error state is always small, all second order products are negligible. This allows easier and faster computation of the various Jacobians.
- The error state dynamics are slow. The ESKF architecture allows the high rate IMU data to be handled outside of the KF, in the INS block. The error states can therefore be computed at a lower rate, reducing computational cost and ensuring a high enough time step to complete all necessary KF calculations.
- The error state formulation facilitate a non-zero dynamic model for the estimates of the gyroscope and accelerometer biases, commonly included in the state vector in addition to the position, velocity and attitude (PVA).

Due to these advantages the ESKF is the chosen Kalman filter design for the state estimation system developed here.

3.6 The ESKF procedure

Furthermore, the behaviour of the ESKF is explained. This explanation is also to a large extent based on the literature of Solà (2017). It is recommended to look at Figure 3.5 while following the next paragraphs.

In the ESKF it is distinguished between the true state vector, $\boldsymbol{x}$, the nominal state vector, $\hat{\boldsymbol{x}}_{ins}$, and the error state vector, $\delta\boldsymbol{x}$. The true state is a composition of the nominal state and the error state. More specifically, the true state is either a linear sum of the nominal state and the error state or a rotation of the nominal state based on the error state. The rotation can either be carried out using a rotation matrix, as proposed by Farrell (2008), or with a quaternion product, as recognized in the design choice of Solà (2017). The idea of the ESKF is to consider the nominal state as a large signal while the error state is a small signal that is linearly integrable and suitable for linear Gaussian filtering.

On one side, IMU data arriving at a high frequency is integrated into the nominal states, $\hat{\boldsymbol{x}}_{ins}$. This data is contained in the input vector, $\boldsymbol{u}$, in Figure 3.5. The nominal

Table 3.3: Overview of the variables in the ESKF and their relations.

Variable type	True state	Nominal state	Error state	Composition	Measured input	Noise
Full state	$\boldsymbol{x}$	$\hat{\boldsymbol{x}}_{ins}$	$\delta\boldsymbol{x}$	$\boldsymbol{x}^n_{nb} = \hat{\boldsymbol{x}}^n_{ins} \otimes \delta\boldsymbol{x}^n_{nb}$		
Position	$\boldsymbol{p}^n_{nb}$	$\hat{\boldsymbol{p}}^n_{ins}$	$\delta\boldsymbol{p}^n_{nb}$	$\boldsymbol{p}^n_{nb} = \hat{\boldsymbol{p}}^n_{ins} + \delta\boldsymbol{p}^n_{nb}$		
Velocity	$\boldsymbol{v}^n_{nb}$	$\hat{\boldsymbol{v}}^n_{ins}$	$\delta\boldsymbol{v}^n_{nb}$	$\boldsymbol{v}^n_{nb} = \hat{\boldsymbol{v}}^n_{ins} + \delta\boldsymbol{v}^n_{nb}$		
Quaternion	$\boldsymbol{q}^n_{nb}$	$\hat{\boldsymbol{q}}^n_{ins}$	$\delta\boldsymbol{q}^n_{nb}$	$\boldsymbol{q}^n_{nb} = \hat{\boldsymbol{q}}^n_{ins} \otimes \delta\boldsymbol{q}^n_{nb}$		
Rotation matrix	$\boldsymbol{R}^n_b$	$\hat{\boldsymbol{R}}^n_{b,ins}$	$\delta\boldsymbol{R}^n_b$	$\boldsymbol{R}^n_b = \hat{\boldsymbol{R}}^n_{b,ins}\delta\boldsymbol{R}^n_b$		
Angles vector			$\delta\boldsymbol{\theta}^b_{nb}$	$\delta\boldsymbol{q}^n_{nb} = e^{\delta\boldsymbol{\theta}^b_{nb}/2}$		
Accelerometer bias	$\boldsymbol{b}^b_{acc}$	$\hat{\boldsymbol{b}}^b_{acc,ins}$	$\delta\boldsymbol{b}^b_{acc}$	$\boldsymbol{b}^b_{acc} = \hat{\boldsymbol{b}}^b_{acc,ins} + \delta\boldsymbol{b}^b_{acc}$		
Gyrometer bias	$\boldsymbol{b}^b_{ars}$	$\hat{\boldsymbol{b}}^b_{ars,ins}$	$\delta\boldsymbol{b}^b_{ars}$	$\boldsymbol{b}^b_{ars} = \hat{\boldsymbol{b}}^b_{ars,ins} + \delta\boldsymbol{b}^b_{ars}$		
Gravity vector	$\boldsymbol{g}^n_{nb}$	$\hat{\boldsymbol{g}}^n_{ins}$	$\delta\boldsymbol{g}^n_{nb}$	$\boldsymbol{g}^n_{nb} = \hat{\boldsymbol{g}}^n_{ins} + \delta\boldsymbol{g}^n_{nb}$		
Acceleretaions	$\boldsymbol{f}^b_{nb}$				$\boldsymbol{f}^b_{imu}$	$\boldsymbol{w}^b_{acc}$
Angular rates	$\boldsymbol{\omega}^b_{nb}$				$\boldsymbol{\omega}^b_{imu}$	$\boldsymbol{w}^b_{ars}$

states do not take noise terms into account. As a consequence, errors accumulate causing these estimates to drift. In parallel to this, the KF estimation of the error states aims to collect these errors by taking noises and other perturbations into account in its dynamic models.

The KF runs an iteration every time a low rate sensor reading arrives, resulting in a close to optimal estimate of the error states, $\delta\boldsymbol{x}$. The error states are then injected into the nominal states, ideally eliminating the drift since the previous error state injection. Lastly, the error states are reset to zero in the Kalman filter. This procedure is repeated once a new low rate measurement arrive.

In Table 3.3, an overview of all the variables of the ESKF are given. The composition equations utilized for injection of the error states into the nominal states are also provided here, showing how the various variables relate to each other. As Table 3.3 shows, the vehicle states of interest are the position, velocity and attitude along with the accelerometer and gyroscope biases.

3.7 Multiplicative extended Kalman filter

The vehicle's attitude describes the relative orientation of the axes of the body frame and the navigation frame. There are three ways to represent the attitude:

- The rotation or direction cosine matrix R_b^n
- The Euler angles roll, pitch and yaw, represented as ϕ, θ and ψ respectively. These are illustrated in Figure 3.6
- The four dimensional quaternion representation $\boldsymbol{q} = \begin{bmatrix} q_\eta & \boldsymbol{q}_\varepsilon \end{bmatrix}^T$

From Table 3.3 it is seen that the attitude of this navigation system is represented by a quaternion. The quaternion parametrization uses a four dimensional space to represent the attitude. It has a scalar or real part, and a three dimensional vector or imaginary part. For a quaternion defined as in the list above, q_η represents the scalar part and $\boldsymbol{q}_\varepsilon = \begin{bmatrix} q_i & q_j & q_k \end{bmatrix}^T$ represents the three dimensional vector part. This four dimensional representation may appear less intuitive. However, with the quaternion representation comes a number of convenient properties.

An example of a convenient property of the quaternion representation is the linearity of the quaternion differential equations. Compared to the Euler angles, an advantage is the complete avoidance of singularities. Another advantage is the reduction of trigonometric functions in the integration routine. Compared to the direction cosine integration, the number of parameters are significantly smaller using quaternions (Farrell, 2008).

When utilizing a quaternion to represent the attitude, there exist several different quaternion conventions that may be chosen. The most common are the Hamilton quaternion and the JPL quaternion. For this system the Hamilton convention is used, in accordance with Solà (2017). The most important property of this convention is the element order where the scalar part comes first. A comparison between the Hamilton and JPL quaternion can be found in Appendix B.

A Kalman filter, using unit quaternions as the nominal attitude parametrization, is commonly referred to as a multiplicative extended Kalman filter (MEKF). This is because the relation between the nominal attitude and the attitude error no longer is additive. Instead, the injection of the error state into the nominal state is carried out through computation of the quaternion product. This can be recognized in the composition equations in Table 3.3. Appendix B contain a section explaining fundamental quaternion mathematics. This section also covers the quaternion product.

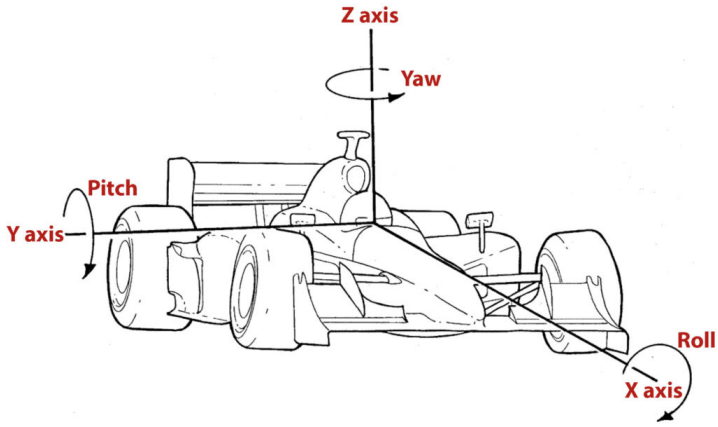


Figure 3.6: The roll, pitch and yaw angles in BODY frame

3.8 Locally defined angular rates and angular error

A design choice affecting the attitude estimation is how the angular rates and angular errors should be defined. These can either be defined locally or globally. In the literature of Solà (2017), an aided inertial navigation system designs is proposed for both choices. For the navigation system developed here, it is chosen to define both the angular rates and the angular errors locally, with respect to the nominal quaternion and the nominal orientation. This choice allows direct use of the gyrometer measurements from the IMU, as these measurements provide body referenced angular rates.

3.9 Stochastic instrument errors

As mentioned, a navigation system combines uncertain information related to a vehicle to accurately estimate its state. The uncertain information is mainly noise and imperfections in the sensor measurements. These are quantities that change with time in a non-deterministic manner. Stochastic processes are therefore important in navigation systems as they allow quantitative analysis of the propagation of uncertainties through a system as a function of time (Farrell, 2008).

Stochastic processes utilize probability theory to make statements about events or processes that are too complicated for detailed deterministic analysis. When a deterministic signal $x_d(t)$ is dependant on a parameter t, it means that the value of x_d can be accurately determined if t is known. For a stochastic signal $x_s(t)$, the parameter t only determines the probability distribution and statistics of x_s.

The error model of a navigation state is usually driven by various sources of sensor

errors. Parts of these error sources are classified as deterministic error sources that are modeled as known biases and compensated for. In this navigation system design, these biases are included in the state vector and estimated together with the rest of the vehicle states.

3.10 Inertial sensors specifications

The aided navigation system developed here utilizes the inertial sensors of the IMU as its main data source. The IMU is an integrated sensor package combining multiple accelerometers and gyroscopes to produce a 3-dimensional measurement of both specific force and and angular rate. Specific force is a measure of acceleration relative to free-fall, hence linear acceleration minus the gravity. Angular rate is a measure of the rate of rotation (VectorNav, 2020)

The IMU present on Atmos is manufactured by VectorNav. The sensor provides measurements at 100 Hz, which is regarded as a high output frequency. The documentation from the manufacturer contains information about the quality of the sensor. The sensor specifications relevant for examination of the various error sources are summarized in Table 3.4.

The sensor specifications from Table 3.4 can be used to achieve a representative error model for the inertial sensor inputs. The various error sources presented in this table introduce errors of different characters. To determine an appropriate error model each of the errors should be considered separately in the light of the expected behaviour of the vehicle.

In a technical report from 1997 published by the Norwegian Defence Research Establishment (FFI), Gade presents a thorough examination of the various error sources in the inertial sensors on an autonomous underwater vehicle (AUV). In the report, the contribution from each error source to the final error model is systematically put forward. In the following sections, the error sources present in

Table 3.4: IMU error specifications from manufacturer

Gyroscope		**Accelerometer**	
Error Source	*Value*	*Error Source*	*Value*
Alignment	$\pm 0.05°$	Alignment	$\pm 0.05°$
Linearity	$< 0.1\%$ FS*	Linearity	$< 0.5\%$ FS*
Bias Stability	$< 10°/\text{hr}$	Bias Stability	< 0.04 mg
Noise Density	$0.0035°/(\text{s}\sqrt{\text{Hz}})$	Noise Density	$< 0.14\ \text{mg}/\sqrt{\text{Hz}}$

* Full Scale. For the gyroscope this corresponds to $4000°/\text{s}$ and for the accelerometer the full scale is 32g.

the IMU on Atmos will be systematically investigated in a similar manner. The investigation is carried out with respect to the specifications in Table 3.4 and the expected behaviour of the target vehicle.

3.11 Gyroscope error model

When considering various error sources it is common to distinguish between low frequency errors and high frequency errors by collecting each of these in two separate terms in the error model. The gyroscope is hence modeled as follows.

$$\boldsymbol{\omega}_{imu}^{b} = \boldsymbol{\omega}_{nb}^{b} + \boldsymbol{b}_{ars}^{b} + \boldsymbol{w}_{ars}^{b} = \boldsymbol{\omega}_{nb}^{b} + \Delta\boldsymbol{\omega}_{nb}^{b} \tag{3.4}$$

$\Delta\boldsymbol{\omega}_{nb}^{b}$ is defined as the total gyroscope error model. $\boldsymbol{b}_{ars}^{b}$ denotes the gyroscope bias term or the standard deviation of the gyroscope, due to low frequency error sources. $\boldsymbol{w}_{ars}^{b}$ denotes high frequency measurement noise. Low frequency errors include the alignment errors, the in-run bias stability and the linearity errors. The high frequency term mainly includes white noise with the noise density given in Table 3.4. Moreover, each of the two terms in the error model will be modeled based on Table 3.4.

3.11.1 Modeling of the gyroscope bias: $\boldsymbol{b}_{ars}^{b}$

Biases are common to model as first order Gauss-Markov processes driven by Gaussian white noise (Gade, 1997). This yields

$$\dot{\boldsymbol{b}}_{ars}^{b} = -\frac{1}{T_{ars}}\boldsymbol{b}_{ars}^{b} + \boldsymbol{w}_{b,ars}^{b} \tag{3.5}$$

The term $\boldsymbol{w}_{b,ars}^{b}$ is referred to as process noise and represent white noise processes that drive the bias. This noise is different from $\boldsymbol{w}_{ars}^{b}$ from (3.4) which is measurement noise. The process noise is normally distributed with zero mean and variance equal to $\boldsymbol{Q}_{ars}\boldsymbol{\delta}(\tau)$. The gyroscope bias in each direction is assumed uncorrelated such that

$$\boldsymbol{Q}_{ars} = \begin{bmatrix} \sigma^2_{Q_{ars},x} & 0 & 0 \\ 0 & \sigma^2_{Q_{ars},y} & 0 \\ 0 & 0 & \sigma^2_{Q_{ars},z} \end{bmatrix} \tag{3.6}$$

To determine the value of $\sigma_{Q_{ars},i}$, where $i \in \{x, y, z\}$, the relation from Appendix D is utilized. This relation states that

$$\sigma^2_{b^b_{ars},i} = \sigma^2_{Q_{ars},i} \frac{T_{ars}}{2} \tag{3.7}$$

Hence, $\sigma_{Q_{ars},i}$ is dependant on the time constant and the standard deviation of the gyroscope bias term. The bias is expected to vary sowly and the time constants, T_{ars}, is thus set one hour (Gade, 1997).

$$T_{ars} = 3600\text{s} \tag{3.8}$$

$\sigma_{b^b_{ars},i}$ denotes the standard deviation of the gyroscope bias in each direction, and represents its uncertainty. This standard deviation contains contributions from each of the low frequency error sources present in Table 3.4, specifically the in-run bias stability error, the linearity error and the alignment error. Before being able to determine the value of $\sigma_{b^b_{ars},i}$, the expected contributions from each of the low frequency error sources must be determined based on the IMU specifications and the expected vehicle behaviour. This is carried out below.

Gyroscope in-run bias instability

The first low frequency error source of consideration is the in-run bias stability. This error source is straightforward to consider. Its contribution to $\sigma_{b^b_{ars},i}$, equal to its standard deviation, is directly given by the sensor specifications in Table 3.4. Conversion to the correct unit yields

$$\sigma_{\omega_{bias,i}} = 10 \cdot \frac{\pi}{180} \cdot \frac{1}{3600}\text{rad/s} = 4.85 \cdot 10^{-5}\text{rad/s} \tag{3.9}$$

Gyroscope linearity error

The linearity error is another low frequency error source, contributing to $\sigma_{b^b_{ars},i}$. For axis i the linearity error is equal to $\delta\boldsymbol{\omega}^b_{nb,i} \cdot \boldsymbol{\omega}^b_{nb,i}$, where $\delta\boldsymbol{\omega}^b_{nb,i}$ denotes the linearity error and $\boldsymbol{\omega}^b_{nb,i}$ denotes the angular rate. The latter is considered a deterministic signal while the linearity error is considered a stochastic signal. Utilizing that the expected value of a deterministic signal is the signal itself, the following expression for the standard deviation is achieved.

$$\sigma_{\omega_{lin,i}} = 0.1 \cdot 10^{-2} \cdot \boldsymbol{\omega}^b_{nb}\text{rad/s} \tag{3.10}$$

Here, the value for the linearity error from Table 3.4 is inserted. From the expression above it is observed that the linearity error contribution depends on the

Figure 3.7: The FSG 2019 Autocross and Trackdrive layout

angular rate. Hence, this error source only contributes to $\sigma_{b^b_{ars},i}$ when the vehicle is turning. The size of the contribution is proportional to the size of the turn rate. To properly determine the effect of this error contribution the expected behaviour of the vehicle must be studied.

The dynamic events at the competitions include three tracks, where the Acceleration track is a straight line, the Skidpad track is shaped as the number eight and the track used for both the Autocross and Trackdrive events has a layout that is uninformed in advance of the competition.

The standard deviation of the linearity error is equal to zero when the vehicle is driving straight, which is true for the entire Acceleration event. During the Skidpad event, the vehicle is turning during the entire race. The performance goals, stated in Chapter 2, correspond to 65.5°/s. This is in general a very high turn rate for a ground vehicle, and the linearity error source is expected to have a considerable effect on this race. The Autocross and Trackdrive events are expected to contain smaller turns in both directions. Figure 3.7 show the track layout for Autocross and Trackdrive at FSG 2019, which is regarded as a reasonable indicator of the expected amount and size of turns. The various turn rates and the duration of each turn are however impossible to properly determine in advance.

It is reasonable to treat the expected turn rate in the Skidpad event as the maximum expected turn rate of the vehicle. The turns in this event are also long-lasting,

inducing a larger overall contribution from the linearity errors. By inserting this turn rate into (3.10), the maximum contribution to $\sigma_{b^b_{ars},i}$ from the linearity error becomes

$$\begin{aligned}\sigma_{\omega_{lin,i}} &= 0.1 \cdot 10^{-2} \cdot \boldsymbol{\omega}^b_{nb}\,\text{rad/s} \\ &= 0.1 \cdot 10^{-2} \cdot 65.5 \cdot \frac{\pi}{180}\,\text{rad/s} \\ &= 1.14 \cdot 10^{-3}\,\text{rad/s}\end{aligned} \tag{3.11}$$

Gyroscope alignment error

The last low frequency error source is the alignment error. This error source leads to a slight disorientation of the axes in the reference system. Because of this, they are no longer exactly orthogonal preventing the misalignment to be described by a rotation matrix. The error along each axis must therefore instead be considered separately.

Figure 3.8 illustrates how this kind of misalignment of a single axis introduces an error in a measurement. The error along an axis will have the highest value when

$$\beta = 90^\circ + \delta\alpha, \tag{3.12}$$

using the notations from Figure 3.8. $\delta\alpha$ denotes the alignment error value specified in Table 3.4. Under the assumption of $\delta\alpha$ being small, the alignment error contribution to $\sigma_{b^b_{ars},i}$ can be expressed as

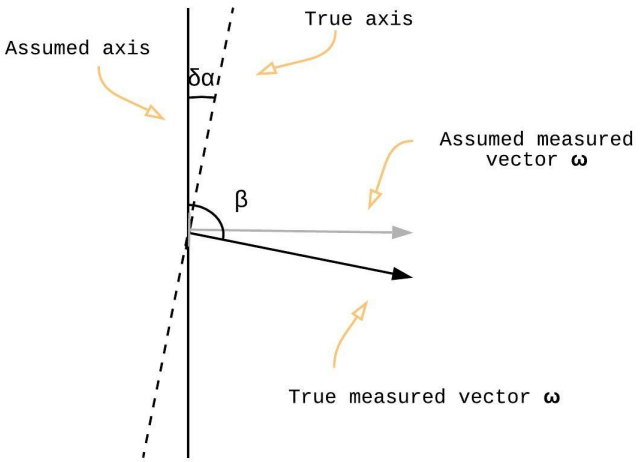


Figure 3.8: Illustration of the alignment error source

$$\sigma_{\omega_{align,i}} = \sin(\delta\alpha) \cdot |\boldsymbol{\omega}_{\boldsymbol{nb}}^{\boldsymbol{b}}| \approx \delta\alpha \cdot |\boldsymbol{\omega}_{\boldsymbol{nb}}^{\boldsymbol{b}}| \tag{3.13}$$

Note that this is correct with respect to units because [rad] is a dimensionless unit. It is considered a special case of the SI units. Insertion of the maximum expected turn rate and the alignment error value from Table 3.4, results in

$$\sigma_{\omega_{align,i}} = (0.05 \cdot \frac{\pi}{180}) \cdot (65.5 \cdot \frac{\pi}{180})\,\text{rad/s} = 9.98 \cdot 10^{-4}\,\text{rad/s} \tag{3.14}$$

With this, the contributions from all the low frequency error sources present in Table 3.4 are computed. The value of $\sigma_{b_{ars}^b,i}$ is calculated as

$$\sigma_{b_{ars}^b,i} = \sqrt{\sigma_{\omega_{align,i}}^2 + \sigma_{\omega_{lin,i}}^2 + \sigma_{\omega_{bias,i}}^2} = 0.0015\,\text{rad/s} \tag{3.15}$$

Moreover, all necessary information in order to determine the standard deviation of the driving noise in (3.5), $\sigma_{Q_{ars},i}$, is now achieved. Insertions of T_{ars} and $\sigma_{b_{ars}^b,i}$ into (3.7) yields

$$\sigma_{Q_{ars},i} = \sqrt{\frac{2}{T_{ars}} \cdot \sigma_{b_{ars}^b,i}^2} = 3.58 \cdot 10^{-5}\,\frac{\text{rad}}{\text{s}}\sqrt{\text{Hz}} \tag{3.16}$$

This completes the model for the gyroscope bias term, $\boldsymbol{b}_{ars}^b$.

3.11.2 Modeling of the gyroscope measurement noise: $\boldsymbol{w}_{ars}^b$

Next, the high frequency measurement noise component in (3.4) is studied. The measurement noise is modeled as a continuous white noise process with zero mean and variance equal to $\boldsymbol{R}_{ars}\boldsymbol{\delta}(\tau)$. Similar to the process noise for the bias drift, the measurement noise in each direction is assumed uncorrelated such that

$$\boldsymbol{R}_{ars} = \begin{bmatrix} \sigma_{R_{ars},x}^2 & 0 & 0 \\ 0 & \sigma_{R_{ars},y}^2 & 0 \\ 0 & 0 & \sigma_{R_{ars},z}^2 \end{bmatrix} \tag{3.17}$$

The standard deviation of the gyroscope measurement noise is given in the sensor specifications and can be directly read out from Table 3.4.

$$\sigma_{R_{ars},i} = 0.0035 \cdot \frac{\pi}{180}\,\frac{\text{rad}}{\text{s}\sqrt{\text{Hz}}} \tag{3.18}$$

3.12 Accelerometer error model

The accelerometer error model, $\Delta \boldsymbol{f}_{nb}^{b}$, is similar in appearance to the error model for the gyroscope. It has a bias term, collecting the low frequency errors, and a measurement noise term, collecting high frequency noise. The specific force arriving from the IMU is therefore expressed as

$$\boldsymbol{f}_{imu}^{b} = \boldsymbol{f}_{nb}^{b} + \boldsymbol{b}_{acc}^{b} + \boldsymbol{w}_{acc}^{b} = \boldsymbol{f}_{nb}^{b} + \Delta \boldsymbol{f}_{nb}^{b} \tag{3.19}$$

Each of the terms contained in $\Delta \boldsymbol{f}_{nb}^{b}$ will now be modeled.

3.12.1 Modeling of the accelerometer bias: $\boldsymbol{b}_{acc}^{b}$

Similar to the gyroscope bias, the accelerometer bias term is modeled as a first order Gauss-Markov process with a time constant equal to one hour.

$$\dot{\boldsymbol{b}}_{acc}^{b} = -\frac{1}{\boldsymbol{T}_{acc}} \boldsymbol{b}_{acc}^{b} + \boldsymbol{w}_{acc}^{b} \tag{3.20}$$

$$T_{acc} = 3600\text{s} \tag{3.21}$$

$\boldsymbol{w}_{acc}^{b}$ denotes the Gaussian white process noise that drives the biases, with zero mean and variance equal to $\boldsymbol{Q}_{acc}\boldsymbol{\delta}(\tau)$. $\boldsymbol{Q}_{acc}$ is written out in (3.22), using the assumption that the driving noise in each direction are uncorrelated.

$$\boldsymbol{Q}_{acc} = \begin{bmatrix} \sigma_{Q_{acc},x}^{2} & 0 & 0 \\ 0 & \sigma_{Q_{acc},y}^{2} & 0 \\ 0 & 0 & \sigma_{Q_{acc},z}^{2} \end{bmatrix} \tag{3.22}$$

The value of $\sigma_{Q_{acc},i}$ is achieved using the relation from Appendix D, in the same way as for the standard deviation of the driving noise in the gyroscope bias model. The relation states that

$$\sigma_{b_{acc}^{b},i}^{2} = \sigma_{Q_{acc},i}^{2} \frac{T_{acc}}{2} \tag{3.23}$$

where $\sigma_{b_{acc}^{b},i}$ denotes the standard deviation of the accelerometer bias in each direction, representing its uncertainty. $\sigma_{b_{acc}^{b},i}$ is composed by the contributions of each low frequency error source present in Table 3.4. To obtain $\sigma_{Q_{acc},i}$, a computation of $\sigma_{b_{acc}^{b},i}$ based on an investigation of the low frequency error sources in the accelerometer is required. This will now be carried out.

Accelerometer bias error

The contribution from the in-run bias stability to $\sigma_{b^b_{acc},i}$ is straightforward to consider here as well, with a standard deviation directly given by Table 3.4. Conversion to the correct unit yields

$$\sigma_{f_{bias,i}} = 0.04 \cdot 10^{-3} \cdot 9.81 \text{m/s}^2 = 3.92 \cdot 10^{-4} \text{m/s}^2 \tag{3.24}$$

Accelerometer linearity error

The expression for the standard deviation of the linearity errors in the accelerometer is provided below.

$$\sigma_{f_{lin}} = 0.5 \cdot 10^{-2} \cdot \boldsymbol{f}^b_{nb} \tag{3.25}$$

Here, the standard deviation of the linearity error depends on the specific force. In order to examine the maximum expected contribution of this error source to $\sigma_{b^b_{acc},i}$, the maximum expected specific force should be investigated. Since this state estimation system is developed for a ground vehicle, the horizontal plane will focused on.

In general, specific force arise when there is a change in the length or a change in the direction of the velocity vector. At competition, the largest linear acceleration is expected at the Acceleration event, where the vehicle is driving a 75 meter long straight track. Due to the performance goals from Chapter 2 the aim is to complete this event in 3.75s. This corresponds to an acceleration of 10.66 m/s^2 in the forward direction. If zero pitch angle is assumed, this value becomes equal to the specific force.

The maximum angular rate is expected at the Skidpad event. If the performance goals is to be met, an angular rate of 65.5°/s is required as already mentioned. The specific force caused by this angular rate is calculated as follows.

$$\boldsymbol{f}^b_{nb} = \boldsymbol{S}(\boldsymbol{\omega}^b_{nb})\boldsymbol{v}^b_{nb} - \boldsymbol{g}^b_{nb} \tag{3.26}$$

Here, $\boldsymbol{S}(\boldsymbol{\omega}^b_{nb})$ represents the skew-symmetric matrix of $\boldsymbol{\omega}^b_{nb}$. Assuming zero roll and pitch angles, the resulting maximum specific force in the horizontal direction becomes

$$\max\left(S\left(\begin{bmatrix}0\\0\\65.5\cdot\frac{\pi}{180}\end{bmatrix}\right)\begin{bmatrix}10.4\\0\\0\end{bmatrix}-\begin{bmatrix}0\\0\\9.81\end{bmatrix}\right)=11.89\text{m/s}^2 \tag{3.27}$$

using the desired average body velocity to reach the performance goals for the Skidpad event. This is higher than the specific force during the Acceleration event, and will be used to calculate the maximum expected standard deviation from the linearity error.

$$\sigma_{f_{lin,i}} = 0.50\cdot10^{-2}\cdot11.89\ \text{m/s}^2 = 5.94\cdot10^{-2}\ \text{m/s}^2 \tag{3.28}$$

Accelerometer alignment error

When investigating the standard deviation of the alignment errors, the same reasoning as for the gyroscope is used. The standard deviation can hence be approximated as

$$\sin(\delta\alpha)\cdot|\boldsymbol{f}_{nb}^b| \approx \delta\alpha\cdot|\boldsymbol{f}_{nb}^b| \tag{3.29}$$

using the notation of Figure 3.8. This error is also dependant on the specific force. In the previous section it was found that the maximum expected specific force in the horizontal plane is equal to 11.89 m/s^2, and is expected to find place during Skidpad. This yields

$$\sigma_{f_{align,i}} = 0.05\cdot\frac{\pi}{180}\cdot11.89\ \text{m/s}^2 = 1.04\cdot10^{-2}\ \text{m/s}^2 \tag{3.30}$$

All contributions from low frequency error sources in the accelerometer are now computed, enabling determination of the standard deviation of the accelerometer bias.

$$\sigma_{b_{ars}^b,i} = \sqrt{\sigma_{f_{bias,i}}^2+\sigma_{f_{lin,i}}^2+\sigma_{f_{align,i}}^2} = 0.0603\text{m/s}^2 \tag{3.31}$$

By inserting this value into (3.23) together with the time constant of one hours, it is now possible to compute $\sigma_{Q_{acc},i}$.

$$\sigma_{Q_{acc},i} = \sqrt{\frac{2}{T_{acc}}\cdot\sigma_{b_{acc}^b,i}^2} = 0.0014\,\frac{\text{m}}{\text{s}^2}\sqrt{\text{Hz}} \tag{3.32}$$

With this, the model for the accelerometer bias term, $\boldsymbol{b}_{acc}^b$, is complete.

3.12.2 Modeling of the accelerometer measurement noise: w_{acc}^{b}

The high frequency measurement noise component in (3.19) is modeled as continuous white noise, similar to the gyroscope measurement noise. This noise has a Gaussian normal distribution with zero mean and variance equal to $\boldsymbol{R}_{acc}\ \boldsymbol{\delta}(\tau)$. The measurement noise in each direction is assumed to be uncorrelated, such that

$$\boldsymbol{R}_{acc} = \begin{bmatrix} \sigma^2_{R_{acc},x} & 0 & 0 \\ 0 & \sigma^2_{R_{acc},y} & 0 \\ 0 & 0 & \sigma^2_{R_{acc},z} \end{bmatrix} \tag{3.33}$$

The standard deviation of the measurement noise, $\sigma_{R_{acc,i}}$, is read out from Table 3.4.

$$\sigma_{R_{acc,i}} = 0.14 \cdot 10^{-3} \cdot 9.81\ \mathrm{m/s^2} = 0.0014\ \frac{\mathrm{m}}{\mathrm{s^2}\sqrt{\mathrm{Hz}}} \tag{3.34}$$

3.13 Discretization of Gaussian white noise

The state space model consist of a set of process equations and a set of measurement equations. The process equations describe the physical behaviour of the internal states, and are naturally described in continuous time. The measurements, on the other hand, only arrive at certain time instants and it is consequently more correct to model the measurement equations in discrete time.

In the following, an investigation of how discretization of a white noise process can be carried out is presented. The process to be discretized here is the output measurement noise. Unlike continuous white noise, discrete white noise processes are physically realizable. This is due to the discrete Dirac delta function having a finite value at all time instances. Equation (3.35) show how the discrete time white measurement noise is defined.

$$\boldsymbol{w}_k = \frac{1}{h} \int_{t_{k-1}}^{t_k} \boldsymbol{w}(t) dt \tag{3.35}$$

h denotes the time step, or the difference in time between the arrival of each measurement. A complete discrete representation also require determination of

the covariance matrix of $\boldsymbol{w}_k$. This matrix is denoted $\boldsymbol{R}_d$.

$$\begin{aligned}\boldsymbol{R}_{d_k} = cov(\boldsymbol{w}_k) &= E\left\{[\boldsymbol{w}_k][\boldsymbol{w}_k]^T\right\} \\ &= \frac{1}{h^2}\int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{t_k} E\left\{[\boldsymbol{w}(t)][\boldsymbol{w}(\tau)]^T\right\} dtd\tau \\ &= \frac{1}{h^2}\int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{t_k} \boldsymbol{R}(t)\boldsymbol{\delta}(t-\tau)dtd\tau\end{aligned} \tag{3.36}$$

Under the assumption that the step size h is small compared to the change rate of $\boldsymbol{R}(t)$, the approximation of (3.37) holds. This will consequently be utilized in the discrete time error models in the state estimation system developed here.

$$\boldsymbol{R}_{d_k} = \frac{1}{h^2}\int_{t_{k-1}}^{t_k} \boldsymbol{R}(\tau)d\tau \approx \frac{\boldsymbol{R}(t_k)}{h} \tag{3.37}$$

4

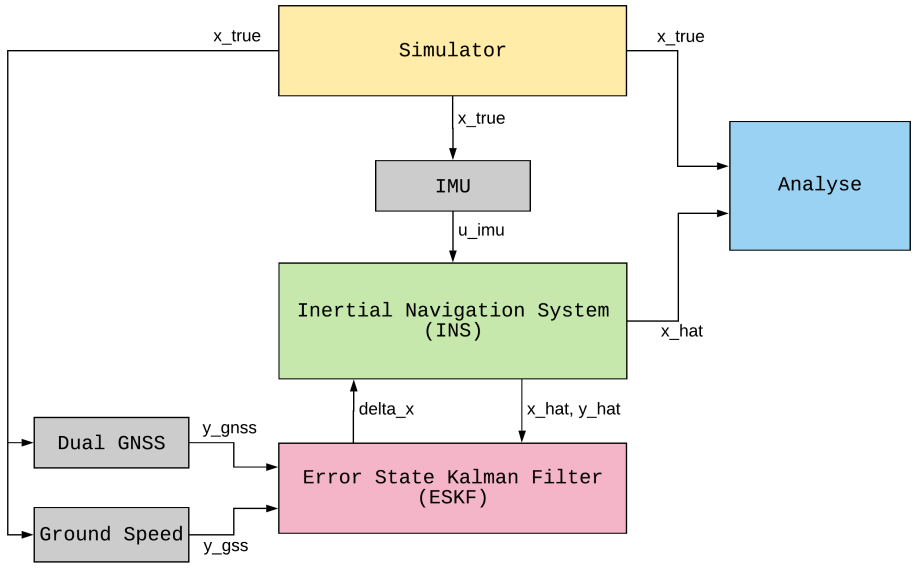


Figure 4.1: The general software structure of the navigation system

The State Estimation System

In this chapter the final design of the developed state estimation system is presented. The aim of this chapter is to systematically describe the implementation to an extent that enables the system to be reproduced. Thorough documentation is valuable for organizations like Revolve NTNU in order to learn from the experiences of past teams and always move forward in their solutions.

This state estimation system is developed in MATLAB. Initially, it was intended that the software, having reached a sufficient state of completeness, should be rewritten in C++ and integrated with ROS in order to run on the vehicle together with the rest of the total autonomous system. However, due to the limitations related to the COVID-19 situation, a simulator is instead developed in MATLAB and utilized for validation.

The structure of the developed state estimation software is illustrated in Figure 4.1. This is intended to provide an overview of the system and will be frequently referred to during the rest of this chapter. The chapter is structured such that each block in the block diagram of Figure 4.1 is explained in turn.

4.1 Simulator

The simulator block in Figure 4.1 simulates a certain behaviour of the race car. It outputs the true states along with the true specific forces and angular rates. These are useful both for validation, providing simulated ground truth, and for modeling of the various sensors.

4.1.1 True state space

The simulator block concerns the true state space and is in control of the true behaviour of the vehicle. This state space is defined by the true state vector, $\boldsymbol{x}$, which is given below.

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p}_{nb}^{n} \\ \boldsymbol{v}_{nb}^{n} \\ \boldsymbol{b}_{acc}^{b} \\ \boldsymbol{q}_{nb}^{n} \\ \boldsymbol{b}_{ars}^{b} \end{bmatrix} \tag{4.1}$$

The true state vector above has the following nomenclature:

$\boldsymbol{p}_{nb}^{n}$	True position.
$\boldsymbol{v}_{nb}^{n}$	True velocity.
$\boldsymbol{b}_{acc}^{b}$	True accelerometer bias.
$\boldsymbol{q}_{nb}^{n}$	True attitude quaternion.
$\boldsymbol{b}_{ars}^{b}$	True gyroscope bias.

All of the states are three-dimensional except for the attitude quaternion which has four dimensions.

4.1.2 True state kinematics

The true state kinematic equations are used in the simulator block to propagate the true vehicle states forward in time. The continuous time differential equations for each of these states are given below.

$$\dot{\boldsymbol{p}}_{nb}^{n} = \boldsymbol{v}_{nb}^{n} \tag{4.2a}$$

$$\dot{\boldsymbol{v}}_{nb}^{n} = \boldsymbol{R}_{b}^{n} \boldsymbol{a}_{nb}^{b} \tag{4.2b}$$

$$\dot{\boldsymbol{b}}_{acc}^{b} = -\frac{1}{T_{acc}} \boldsymbol{b}_{acc}^{b} + \boldsymbol{w}_{b,acc}^{b} \tag{4.2c}$$

$$\dot{\boldsymbol{q}}_{nb}^{n} = \frac{1}{2} \boldsymbol{q}_{nb}^{n} \otimes \boldsymbol{\omega}_{nb}^{b} \tag{4.2d}$$

$$\dot{\boldsymbol{b}}_{ars}^{b} = -\frac{1}{T_{ars}} \boldsymbol{b}_{ars}^{b} \boldsymbol{w}_{b,ars}^{b} \tag{4.2e}$$

The implementation of this state estimation system is carried out in the discrete time domain. This is convenient as both the high rate IMU measurements and the low rate aiding measurements only arrive at specific time instances. Hence, in order to include the kinematic equations in the implementation, a discretization of (4.2) is necessary. The resulting discrete update equations for the true states are

$$\boldsymbol{p}^n_{nb_k} = \boldsymbol{p}^n_{nb_{k-1}} + \boldsymbol{v}^n_{nb_{k-1}} h + \frac{1}{2}\boldsymbol{R}^n_{b_{k-1}} \boldsymbol{a}^b_{nb_{k-1}} h^2 \tag{4.3a}$$

$$\boldsymbol{v}^n_{nb_k} = \boldsymbol{v}^n_{nb_{k-1}} + \boldsymbol{R}^n_{b_{k-1}} \boldsymbol{a}^b_{nb_{k-1}} h \tag{4.3b}$$

$$\boldsymbol{b}^b_{acc_k} = e^{-\frac{1}{T_{acc}}h} \boldsymbol{b}^b_{acc_{k-1}} + \boldsymbol{w}_{b,acc_{k-1}} \tag{4.3c}$$

$$\boldsymbol{q}^n_{nb_k} = \boldsymbol{q}^n_{nb_{k-1}} \otimes \boldsymbol{q}\{\boldsymbol{\omega}^b_{nb_{k-1}}\} \tag{4.3d}$$

$$\boldsymbol{b}^b_{ars_k} = e^{-\frac{1}{T_{ars}}h} \boldsymbol{b}^b_{ars_{k-1}} + \boldsymbol{w}_{b,ars_{k-1}} \tag{4.3e}$$

In the quaternion update equation, the term $\boldsymbol{q}\{\boldsymbol{\omega}^b_{nb_{k-1}}\}$ is recognized. To explain this notation, the generic expression, $\boldsymbol{q}\{\boldsymbol{v}\}$, is used. $\boldsymbol{q}\{\boldsymbol{v}\}$ denotes the quaternion associated with the vector $\boldsymbol{v} = \phi\boldsymbol{u}$, representing a rotation of ϕ radians about the axis $\boldsymbol{u}$. To compute this quaternion, the conversion formula below is utilized (Solà, 2017).

$$\boldsymbol{q}\{\boldsymbol{v}\} = \boldsymbol{q}\{\phi\boldsymbol{u}\} = e^{\phi\boldsymbol{u}/2} = \begin{bmatrix} \cos(\phi/2) \\ \boldsymbol{u}\sin(\phi/2) \end{bmatrix} \tag{4.4}$$

Having the discrete kinematic equations, the simulator block i capable of propagating all the vehicle states and determine the true behaviour of the vehicle under the affection of a certain input sequence. The input sequence is constructed in order to provoke a behaviour corresponding to the choice of case study.

4.1.3 Choice of case study

The simulated case in focus is the Skidpad event at the FS competitions. This is chosen because it is the most complex dynamic event where the track layout is known in advance of the competition. This event also excites the dynamics of almost all the vehicle states of interest, making it a suitable case of study. To obtain sufficient excitement of roll and pitch, a slight deviation from a horizontal ground is assumed. From the examination of IMU errors in the previous chapter, it is also found that the Skidpad event is the event facing the largest effects from the error sources in the inertial sensors. As mentioned in the previous chapters, the

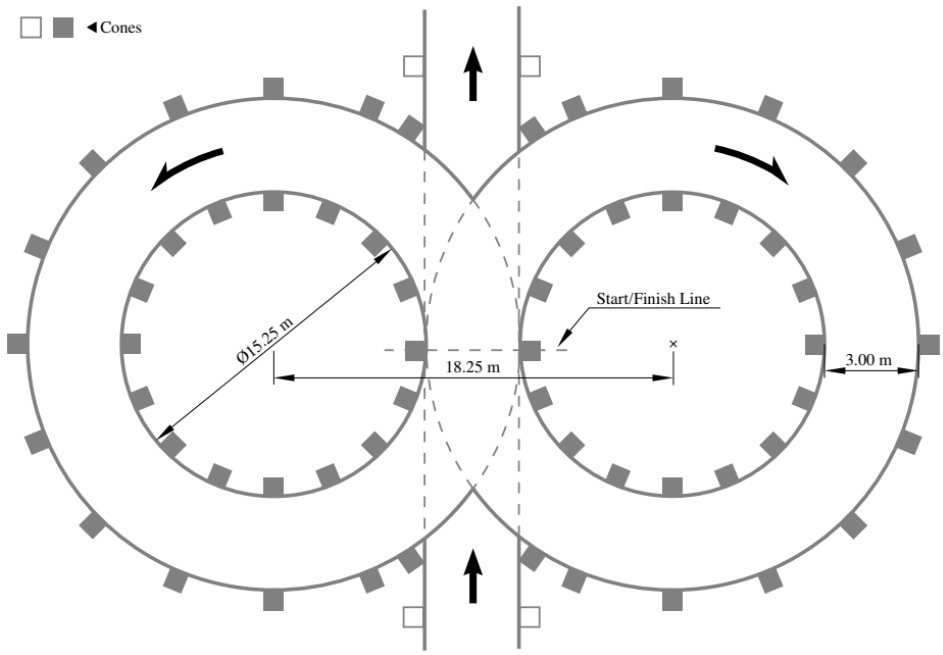


Figure 4.2: Skidpad track layout due to competition rules

Skidpad track is shaped as the number eight. The exact layout, according to the competition rules (FSG, 2020), is shown in Figure 4.2. A corresponding layout is used in the simulations.

4.1.4 The Skidpad procedure

The procedure of the Skidpad event is as follows. The vehicle starts by driving in a straight line, referred to as the "pit lane", and enters the track perpendicular to the figure eight. Then, two full laps on the right circle is taken, where the second lap is timed. Hereafter, the vehicle enters the left circle of the Skidpad track for the third lap and completes two laps on this circle as well. Also here, the second lap is timed.

To match the competition procedure, the pit lane is included in the simulated case and two laps are completed of each circle before entering the next. In addition it is assumed that the state estimation system is turned on for a short period of time while the vehicle remains stationary, waiting for permission to start the race.

The performance goal of this year's team at the Skidpad event is to finish each circle within an average time of 5.5 seconds, as stated in Ch.2. This corresponds to an absolute velocity of 10.4 m/s, which is the velocity applied in the simulations.

Moreover, it is recognized from Figure 4.1 that the true state vector, $\boldsymbol{x}$, from the simulator block is used as input to all the blocks in the block diagram representing sensors. As this state estimation system is an aided inertial navigation system, the main data source is the IMU. This is the next block to be presented.

4.2 Inertial measurement unit (IMU)

The IMU block in Figure 4.1 provides the high rate inertial sensor measurements used as the main set of sensors in this state estimation system. As mentioned in Ch.3, an IMU is composed of an accelerometer and a gyroscope. Equation (4.5) and (4.6) show the models for the specific forces, $\boldsymbol{f}^b_{imu}$, from the accelerometer and the angular rates, $\boldsymbol{\omega}^b_{imu}$, from the gyroscope respectively.

$$\boldsymbol{f}^b_{imu} = \boldsymbol{a}^b_{nb} + \boldsymbol{S}(\boldsymbol{\omega}^b_{nb})\boldsymbol{v}^b_{nb} - \boldsymbol{R}^{nT}_b \boldsymbol{g}^n_{nb} + \boldsymbol{b}^b_{acc} + \boldsymbol{w}^b_{acc} \tag{4.5}$$

$$\boldsymbol{\omega}^b_{imu} = \boldsymbol{\omega}^b_{nb} + \boldsymbol{b}^b_{ars} + \boldsymbol{w}^b_{ars} \tag{4.6}$$

The inertial sensor measurements are modeled as functions of the true specific forces and angular rates, assumed corrupted by various error sources. The error models used here correspond to the error models worked out in Ch.2, with regards to the sensor specifications provided by the manufacturer. The specific force and angular rate vectors make up the members of $\boldsymbol{u}_{imu}$ in Figure 4.1. This vector is further fed into the INS block.

4.3 Inertial navigation system (INS)

The INS block in Figure 4.1 performs the integration of the incoming high-rate IMU measurement outputs to obtain an estimate of the desired vehicle states. These states represent expected values of the vehicle states based on the IMU readings alone. Since error components have expected values equal to zero, no noise or perturbations are taken into account in this block.

4.3.1 Nominal state space

The estimated states obtained from integration of the high rate IMU measurements are referred to as the "nominal states" and make up the "nominal state space". The nominal state vector corresponds to the true state vector in (4.1), and is stated

below.

$$\hat{\boldsymbol{x}}_{ins} = \begin{bmatrix} \hat{\boldsymbol{p}}^n_{ins} \\ \hat{\boldsymbol{v}}^n_{ins} \\ \hat{\boldsymbol{b}}^b_{acc,ins} \\ \hat{\boldsymbol{q}}^n_{ins} \\ \hat{\boldsymbol{b}}^b_{ars,ins} \end{bmatrix} \tag{4.7}$$

In correspondence with the true state vector, the following nomenclature is used:

$\hat{\boldsymbol{p}}^n_{ins}$	Nominal position.
$\hat{\boldsymbol{v}}^n_{ins}$	Nominal velocity.
$\hat{\boldsymbol{b}}^b_{acc,ins}$	Nominal accelerometer bias.
$\hat{\boldsymbol{q}}^n_{ins}$	Nominal attitude quaternion.
$\hat{\boldsymbol{b}}^b_{ars,ins}$	Nominal gyroscope bias.

All nominal states are in three dimensions, with the exception of the nominal attitude quaternion which is four-dimensional.

4.3.2 Nominal state kinematics

The continuous kinematic equations for the nominal states are utilized to propagate these states forward in time based on the input from the IMU block. The kinematic equations for the nominal states used in this implementation are given below.

$$\dot{\hat{\boldsymbol{p}}}^n_{ins} = \hat{\boldsymbol{v}}^n_{ins} \tag{4.8a}$$

$$\dot{\hat{\boldsymbol{v}}}^n_{ins} = \hat{\boldsymbol{a}}^n_{ins} \tag{4.8b}$$

$$\dot{\hat{\boldsymbol{b}}}^b_{acc,ins} = -\frac{1}{T_{acc}}\hat{\boldsymbol{b}}^b_{acc,ins} \tag{4.8c}$$

$$\dot{\hat{\boldsymbol{q}}}^n_{ins} = \frac{1}{2}\hat{\boldsymbol{q}}^n_{ins} \otimes \hat{\boldsymbol{\omega}}^b_{ins} \tag{4.8d}$$

$$\dot{\hat{\boldsymbol{b}}}^b_{ars,ins} = -\frac{1}{T_{ars}}\hat{\boldsymbol{b}}^b_{ars,ins} \tag{4.8e}$$

The nominal acceleration, $\hat{\boldsymbol{a}}^n_{ins}$, is obtained due to (4.9) below. Here, it is seen that the gravitational force is added to the rotated specific force from the IMU, where the estimated accelerometer bias is subtracted. The nominal angular rate, $\hat{\boldsymbol{\omega}}^b_{ins}$, is calculated due to (4.10) by removing the estimated angular rate bias from the angular rates provided by the IMU.

$$\hat{\boldsymbol{a}}^n_{ins} = \hat{\boldsymbol{R}}^n_{b,ins}(\boldsymbol{f}^b_{imu} - \hat{\boldsymbol{b}}^b_{acc,ins}) + \hat{\boldsymbol{g}}^n_{ins} \tag{4.9}$$

$$\hat{\boldsymbol{\omega}}^b_{ins} = \boldsymbol{\omega}^b_{imu} - \hat{\boldsymbol{b}}^b_{ars,ins} \tag{4.10}$$

Similar to the true state kinematics, the equations of (4.8) are discretized before they are implemented in the software. The discrete dynamic model for the nominal state is given in (4.11).

$$\hat{\boldsymbol{p}}^n_{ins_k} = \hat{\boldsymbol{p}}^n_{ins_{k-1}} + \hat{\boldsymbol{v}}^n_{ins,k-1}h + \frac{1}{2}\hat{\boldsymbol{a}}^n_{ins_{k-1}}h^2 \tag{4.11a}$$

$$\hat{\boldsymbol{v}}^n_{ins_k} = \hat{\boldsymbol{v}}^n_{ins_{k-1}} + \hat{\boldsymbol{a}}^n_{ins_{k-1}}h^2 \tag{4.11b}$$

$$\hat{\boldsymbol{b}}^b_{acc,ins_k} = \hat{\boldsymbol{b}}^b_{acc,ins_{k-1}} \tag{4.11c}$$

$$\hat{\boldsymbol{q}}^n_{ins_k} = \hat{\boldsymbol{q}}^n_{nb_{k-1}} \otimes \hat{\boldsymbol{q}}\{\hat{\boldsymbol{\omega}}^b_{ins_{k-1}}\} \tag{4.11d}$$

$$\hat{\boldsymbol{b}}^b_{ars,ins_k} = \hat{\boldsymbol{b}}^b_{ars,ins_{k-1}} \tag{4.11e}$$

$\hat{\boldsymbol{q}}\{\hat{\boldsymbol{\omega}}^b_{ins_{k-1}}\}$ is computed using (4.4).

As seen in Figure 4.1, the INS block outputs estimates corresponding to the aiding sensor measurements, denoted $\hat{\boldsymbol{y}}$, in addition to the nominal state vector. These estimates correspond to the low rate aiding measurements, represented within the nominal state space. How the expressions of the various members of $\hat{\boldsymbol{y}}$ are constructed, become more evident when presented together with the models for the low rate aiding sensors. The models for the various aiding sensor and their corresponding estimates are therefore presented together in the next section.

4.4 Low rate aiding sensors

The dual GNSS block and the GSS block in Figure 4.1 make up the blocks representing the low rate aiding sensors. These take in the true state vector, $\boldsymbol{x}$, as input from the simulator block, and provide aiding measurements as output to the ESKF block. The low rate aiding sensor blocks provide four different quantities: the two GNSS position measurements, the ground speed measurement and the baseline vector between the GNSS antennas. These are collected in the measurement vector, $\boldsymbol{y}$.

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_{gnss1} \\ \boldsymbol{y}_{gnss2} \\ \boldsymbol{y}_{gss} \\ \boldsymbol{y}_{bl} \end{bmatrix} \tag{4.12}$$

Each measured quantity is modeled as a measurement functions of the true state, $\boldsymbol{x}$. In the following, each of these sensor models are presented along with the corresponding nominal estimates, $\hat{\boldsymbol{y}}$. The appearance of $\hat{\boldsymbol{y}}$ is equivalent to $\boldsymbol{y}$.

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{\boldsymbol{y}}_{gnss1} \\ \hat{\boldsymbol{y}}_{gnss2} \\ \hat{\boldsymbol{y}}_{gss} \\ \hat{\boldsymbol{y}}_{bl} \end{bmatrix} \tag{4.13}$$

4.4.1 Dual GNSS position measurements

The two GNSS antennas are placed a known distance from the CG of the car. The position measurements from each of these antennas are therefore modeled as displaced positions. With two of these measurements along with knowledge about the positions of the antennas relative to each other, the attitude of the vehicle becomes observable. The measurement functions used for the GNSS position measurements are

$$\boldsymbol{y}_{gnss,1} = \boldsymbol{p}^n_{nb} + \boldsymbol{R}^n_b \boldsymbol{r}^b_1 + \boldsymbol{w}_{gnss} \tag{4.14}$$

$$\boldsymbol{y}_{gnss,2} = \boldsymbol{p}^n_{nb} + \boldsymbol{R}^n_b \boldsymbol{r}^b_2 + \boldsymbol{w}_{gnss} \tag{4.15}$$

The true rotation matrix, $\boldsymbol{R}^n_b$, and the true position, $\boldsymbol{p}^n_{nb}$, are achieved from the simulator block. Furhtermore, $\boldsymbol{r}^b_1$ and $\boldsymbol{r}^b_2$ represent the known lever arms from the CG to each GNSS antenna. In addition, Gaussian white measurement noise is added to both of the measurement functions, as some disturbances are expected.

The corresponding nominal quantities are obtained from performing a similar displacement of the latest nominal position estimate obtained from the INS block.

$$\hat{\boldsymbol{y}}_{gnss,1} = \hat{\boldsymbol{p}}^n_{ins} + \hat{\boldsymbol{R}}^b_{n,ins} \boldsymbol{r}^b_1 \tag{4.16}$$

$$\hat{\boldsymbol{y}}_{gnss,2} = \hat{\boldsymbol{p}}^n_{ins} + \hat{\boldsymbol{R}}^b_{n,ins} \boldsymbol{r}^b_2 \tag{4.17}$$

$\hat{\boldsymbol{R}}_b^n$ denotes the nominal rotation matrix, computed from the latest nominal attitude quaternion. Similar to the nominal states, $\hat{\boldsymbol{x}}_{ins}$, no noise is included in $\hat{\boldsymbol{y}}$ as the expected values of these components are zero.

4.4.2 Ground speed velocity measurements

When modeling the ground speed sensor, there are two significant properties that have to be accounted for. Firstly, the sensor is not located exactly as CG, similar to the GNSS antennas. The measurements must therefore be modeled as displaced velocities. The displaced velocity is derived by differentiation of the displaced position:

$$\begin{aligned} \boldsymbol{p}_{gss}^n &= \boldsymbol{p}_{nb}^n + \boldsymbol{R}_b^n \boldsymbol{r}_{gss}^b && \text{(4.18a)} \\ \boldsymbol{v}_{gss}^n &= \dot{\boldsymbol{p}}_{gss}^n \\ &= \boldsymbol{v}_{nb}^n + \dot{\boldsymbol{R}}_b^n \boldsymbol{r}_{gss}^b + \boldsymbol{R}_b^n \cancel{\dot{\boldsymbol{r}}_{gss}^b} && \text{(4.18b)} \\ &= \boldsymbol{v}_{nb}^n + \boldsymbol{R}_b^n S(\boldsymbol{\omega}_{nb}^b) \boldsymbol{r}_{gss}^b \end{aligned}$$

Secondly, the GSS outputs the ground speed, which is a scalar value. This velocity corresponds to the absolute horizontal velocity of the vehicle given in BODY frame. When modeling this sensor, the aim is to arrive at a formulation that makes it possible to represent this measurement within the estimated state space. One of the challenges here is that the GSS outputs the absolute value of a two dimensional size. To resolve this, an allocation matrix is utilized enabling this sensor to be modeled as a function of the three dimensional velocity vector. This is shown in the third line of (4.19) where the allocation matrix is multiplied with the rotated three dimensional velocity vector in the first term.

$$\begin{aligned} \boldsymbol{y}_{gss} &= h_{gss}(\boldsymbol{x}_t) + \boldsymbol{w}_{gss} \\ &= \left\| \boldsymbol{v}_{gss}^b \right\|_2 + \boldsymbol{w}_{gss} \\ &= \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{R}_b^{nT} \boldsymbol{v}_{gss}^n \right\|_2 + \boldsymbol{w}_{gss} \end{aligned} \tag{4.19}$$

The allocation matrix multiplied with the rotation matrix yields the plane in the NED frame corresponding to the xy-plane in the BODY frame.

The corresponding estimated velocity $\hat{\boldsymbol{y}}_{gss}$ is obtained starting from the latest INS estimate of the velocity. The two dimensional mapping is then accounted for

following (4.19), with the displacement expressed as in (4.18b).

$$\begin{aligned}\hat{\boldsymbol{y}}_{gss} &= \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \hat{\boldsymbol{R}}_{b,ins}^{n\ T} (\hat{\boldsymbol{v}}_{ins}^{n} + \hat{\boldsymbol{R}}_{b,ins}^{n} \boldsymbol{S}(\boldsymbol{\omega}_{imu}^{b} - \hat{\boldsymbol{b}}_{ars,ins}^{b}) \boldsymbol{r}_{gss}^{b}) \right\|_2 + \boldsymbol{w}_{gss} \\ &= \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \hat{\boldsymbol{R}}_{b,ins}^{n\ T} \hat{\boldsymbol{v}}_{ins}^{n} + \boldsymbol{S}(\boldsymbol{\omega}_{imu}^{b} - \hat{\boldsymbol{b}}_{ars,ins}^{b}) \boldsymbol{r}_{gss}^{b} \right\|_2 + \boldsymbol{w}_{gss}\end{aligned} \tag{4.20}$$

4.4.3 Dual GNSS baseline vector

The vector between the two GNSS antennas, referred to as the "baseline", is included as the third low rate aiding measurement. This measurement is included in order to aid in determining accurate heading even when the vehicle is at rest. The GNSS baseline vector is denoted $\boldsymbol{y}_{bl}$ and its measurement function is calculated as the difference between $\boldsymbol{y}_{gnss1}$ and $\boldsymbol{y}_{gnss2}$.

$$\begin{aligned}\boldsymbol{y}_{bl} &= \boldsymbol{y}_{gnss2} - \boldsymbol{y}_{gnss1} \\ &= (\boldsymbol{p}_{nb}^{n} + \boldsymbol{R}_{b}^{n} \boldsymbol{r}_{2}^{b} + \boldsymbol{w}_{gnss}) - (\boldsymbol{p}_{nb}^{n} + \boldsymbol{R}_{b}^{n} \boldsymbol{r}_{1}^{b} + \boldsymbol{w}_{gnss}) \\ &= \boldsymbol{R}_{b}^{n} (\boldsymbol{r}_{2}^{b} - \boldsymbol{r}_{1}^{b})\end{aligned} \tag{4.21}$$

The corresponding estimated baseline has a similar expression, formulated within the nominal state space.

$$\hat{\boldsymbol{y}}_{bl} = \hat{\boldsymbol{R}}_{b,ins}^{n} (\boldsymbol{r}_{2}^{b} - \boldsymbol{r}_{1}^{b}) \tag{4.22}$$

With this, all the quantities in the measurement vector, $\boldsymbol{y}$, are modeled. The corresponding nominal quantities are specified as well. To remove contaminants, note that the aiding measurement vector, $\boldsymbol{y}$, is fed into the ESKF block from the dual GNSS block and the GSS block in Figure 4.1. Since $\hat{\boldsymbol{y}}$ calculates the corresponding values within the nominal state space this vector has to arrive from the INS block. This is also shown in Figure 4.1.

4.5 Error state Kalman filter

Next in line, is the ESKF block. This block is the most important and computationally heavy block in the state estimation system. The actual sensor fusion takes place in this block, utilizing the information from the low rate aiding sensors to eliminate the effects of the various error sources and uncertainties in the system. The ESKF block take input from the both the low rate aiding sensor blocks in addition to the INS block. An error state is then estimated with a Kalman filter using all of the collected information. This error state is subsequently fed back to the INS block to eliminate drift in the nominal states. This is repeated every time a new set of aiding measurements arrive.

The work carried out by the ESKF block at the arrival of a low rate measurement can be dived into five steps. These are linearization, KF prediction, KF correction, injection of the errors into the nominal states and finally the reset step. The way each of these steps are implemented is explained below. To begin with, the error states space and the error state kinematics are outlined.

4.5.1 Error state space

The error state space is described by the error state vector, which is provided below.

$$\delta \boldsymbol{x} = \begin{bmatrix} \delta \boldsymbol{p}^n_{nb} \\ \delta \boldsymbol{v}^n_{nb} \\ \delta \boldsymbol{b}^b_{acc} \\ \delta \boldsymbol{q}^n_{nb} \\ \delta \boldsymbol{b}^b_{ars} \\ \delta \boldsymbol{g}^n_{nb} \end{bmatrix} \tag{4.23}$$

The following nomenclature is used for the error state vector:

$\delta \boldsymbol{p}^n_{nb}$	Position error.
$\delta \boldsymbol{v}^n_{nb}$	Velocity error.
$\delta \boldsymbol{b}^b_{acc}$	Accelerometer bias error.
$\delta \boldsymbol{q}^n_{nb}$	Attitude quaternion error.
$\delta \boldsymbol{b}^b_{ars}$	Gyroscope bias error.
$\delta \boldsymbol{g}^n_{nb}$	Gravity error.

The error state vector does not correspond exactly to the true state vector and the nominal state vector. Firstly, the error state vector is expanded with one additional state being the gravitational error state $\delta \boldsymbol{g}^n_{nb}$. This is included because it is assumed that the gravity is calculated inaccurately. An alternative implementation is to omit the gravity error state and instead assume that it is a part of the accelerometer bias error state.

Secondly, the attitude error state is represented using the three parameter angular error, $\delta \boldsymbol{\theta}^b_{nb}$, instead of the four parameter unit quaternion. Hence, all states in this state space are three-dimensional. The four parameter unit quaternion makes the covariance matrix become rank deficient due to the unit constraint (Fossen, 2019). With the three parameter error state formulation this problem is resolved. Since the angular error is small, the singularities that may occur close to 90 degrees is not of concern. The angular error being small also leads to the following relation to the unit quaternion parametrization of the same error state:

$$\delta \boldsymbol{q}^n_{nb} \xrightarrow[\delta \boldsymbol{\theta} \to 0]{} \begin{bmatrix} 1 \\ \frac{1}{2} \delta \boldsymbol{\theta}^b_{nb} \end{bmatrix} \tag{4.24}$$

4.5.2 Error state kinematics

The error states relate to the nominal states following the composition equations from Table 3.3 in Ch.3. The kinematic equations of the error states are obtained starting from these composition equations and solving for the error states. Since the error states are small signals, all second order terms are assumed negligible. This approach yields the following error state dynamics, utilized in the Kalman filter equations.

$$\delta\dot{\boldsymbol{p}}^n_{nb} = \delta\boldsymbol{v}^n_{nb} \tag{4.25a}$$

$$\delta\dot{\boldsymbol{v}}^n_{nb} = -\hat{\boldsymbol{R}}^n_{b,ins}\delta\boldsymbol{b}^b_{acc} - \hat{\boldsymbol{R}}^n_{b,ins}\boldsymbol{S}(\boldsymbol{f}^b_{imu} - \hat{\boldsymbol{b}}^b_{acc,ins})\delta\boldsymbol{\theta}^b_{nb} + \delta\boldsymbol{g}^n_{nb} - \hat{\boldsymbol{R}}^n_{b,ins}\boldsymbol{w}^b_{acc} \tag{4.25b}$$

$$\delta\dot{\boldsymbol{b}}^b_{acc} = -\frac{1}{T_{acc}}\delta\hat{\boldsymbol{b}}^b_{acc} + \boldsymbol{w}^b_{b,acc} \tag{4.25c}$$

$$\delta\dot{\boldsymbol{\theta}}^b_{nb} = -\boldsymbol{S}(\boldsymbol{\omega}^b_{imu} - \hat{\boldsymbol{b}}^b_{ars,ins})\delta\boldsymbol{\theta}^b_{nb} - \delta\boldsymbol{b}^b_{ars} - \boldsymbol{w}^b_{ars} \tag{4.25d}$$

$$\delta\dot{\boldsymbol{b}}^b_{ars} = -\frac{1}{T_{ars}}\delta\hat{\boldsymbol{b}}^b_{acc} + \boldsymbol{w}_{b,ars} \tag{4.25e}$$

$$\delta\dot{\boldsymbol{g}}^n_{nb} = 0 \tag{4.25f}$$

4.5.3 EKF linearization

The error state kinematics are nonlinear as recognized in (4.25). A linearization process is therefore included in the filter as an additional initial step, following the principle of the EKF. The linearization step involve computation of the Jacobian matrices of (4.25) with respect to the error states, $\delta\boldsymbol{x}$, and the noise components, $\boldsymbol{w}_{imu}$. These matrices are denoted $\boldsymbol{A}$ and $\boldsymbol{E}$ respectively. Their expressions are given below.

$$\boldsymbol{A} = \left.\frac{\partial \boldsymbol{f}}{\partial \delta\boldsymbol{x}}\right|_{\delta\boldsymbol{x}=0} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & -\hat{\boldsymbol{R}}^b_{ins} & -\hat{\boldsymbol{R}}^b_{ins}\boldsymbol{S}(\boldsymbol{f}^b_{imu} - \hat{\boldsymbol{b}}^b_{acc,ins}) & \boldsymbol{0}_3 & \boldsymbol{I}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & -\frac{1}{T_{acc}}\boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & -\boldsymbol{S}(\boldsymbol{\omega}^b_{imu} - \hat{\boldsymbol{b}}_{ars,ins}) & -\boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & -\frac{1}{T_{ars}}\boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} \tag{4.26}$$

$$\boldsymbol{E} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}_{imu}}\bigg|_{\boldsymbol{w}=0} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ -\hat{\boldsymbol{R}}^b_{ins} & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & -\boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} \tag{4.27}$$

Furthermore, since the implementation of this state estimation system is carried out in discrete time, it is necessary to perform a discretization of the linearized dynamics. This is achieved as follows:

$$\boldsymbol{A}_d = \boldsymbol{I}_{18} + \boldsymbol{A}h \tag{4.28}$$

$$\boldsymbol{E}_d = \boldsymbol{E}h \tag{4.29}$$

h denotes the time step given is seconds. With this, the linearization step is completed and a discrete time linear approximation of (4.25) is achieved. The final discrete time linearized error state system on compressed form becomes

$$\delta \boldsymbol{x} \leftarrow \boldsymbol{A}_d(\boldsymbol{u}_{imu}, \boldsymbol{x}_{ins})\delta \boldsymbol{x} + \boldsymbol{E}_d(\boldsymbol{x}_{ins})\boldsymbol{w}_{imu} \tag{4.30}$$

with the discrete Jacobian matrices from (4.28) and (4.29), the nominal and error state vectors from (4.7) and (4.23) respectively, and the following input and perturbation vectors:

$$\boldsymbol{u}_{imu} = \begin{bmatrix} \boldsymbol{f}^b_{imu} \\ \boldsymbol{\omega}^b_{imu} \end{bmatrix} \qquad \boldsymbol{w}_{imu} = \begin{bmatrix} \boldsymbol{w}^b_{acc} \\ \boldsymbol{w}^b_{b,acc} \\ \boldsymbol{w}^b_{ars} \\ \boldsymbol{w}^b_{b,ars} \end{bmatrix} \tag{4.31}$$

4.5.4 ESKF prediction

The ESKF prediction step computes a time propagation of the error state vector, $\delta \boldsymbol{x}$, and its covariance matrix, $\hat{\boldsymbol{P}}$, making use of the Jacobian matrices $\boldsymbol{A}$ and $\boldsymbol{E}$ which represent the linearized dynamics.

$$\delta \boldsymbol{x}_k^- = \boldsymbol{A}_{d_{k-1}} \delta \boldsymbol{x}_{k-1}^+ \tag{4.32}$$

$$\hat{\boldsymbol{P}}_k^- = \boldsymbol{A}_{d_{k-1}} \hat{\boldsymbol{P}}_{k-1}^+ \boldsymbol{A}_{d_{k-1}}{}^T + \boldsymbol{Q}_d \tag{4.33}$$

$$\boldsymbol{Q}_d = \frac{1}{2}(\boldsymbol{A}_{d_{k-1}} \boldsymbol{E}_{k-1} \boldsymbol{Q} \boldsymbol{E}_{k-1}{}^T \boldsymbol{A}_{d_{k-1}}{}^T + \boldsymbol{E}_k \boldsymbol{Q} \boldsymbol{E}_k{}^T) h \tag{4.34}$$

Note that the error state $\delta \boldsymbol{x}$ is reset to zero each time a low rate aiding measurement arrive to the ESKF. Hence, (4.32) always returns zero and is consequently not included in the actual implementation. The discrete process noise covariance matrix, $\boldsymbol{Q}_d$, is defined in (4.34). Note that this matrix is computed using the continuous $\boldsymbol{E}$-matrix and not the discretized one. $\boldsymbol{Q}_d$ represent a discrete approximation of the continuous process noise covariance matrix. It may be approximated in simpler ways, for instance like it is done in the literature of Solà (2017). However, to increase preciseness it is here chosen to use an approximation where the $\boldsymbol{E}$-matrices from two consecutive time steps are included (Maybeck, 1982).

4.5.5 ESKF correction

In the ESKF correction step, the new information provided by the low rate measurements is taken into account and used to correct the predictions of $\delta \boldsymbol{x}$ and $\hat{\boldsymbol{P}}$. The Kalman gain, $\boldsymbol{K}_k$, is also computed and utilized in the correction equations to determine the optimal weighting of this new information. The ESKF correction equations are provided below.

$$\boldsymbol{K}_k = \hat{\boldsymbol{P}}_k^- \boldsymbol{H}_k^T (\boldsymbol{H}_k \hat{\boldsymbol{P}}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \tag{4.35}$$

$$\delta \boldsymbol{x}_k^+ = \boldsymbol{K}_k \delta \boldsymbol{y}_k \tag{4.36}$$

$$\hat{\boldsymbol{P}}_k^+ = (\boldsymbol{I}_{18} - \boldsymbol{K}_k \boldsymbol{H}_k) \hat{\boldsymbol{P}}_k^- (\boldsymbol{I}_{18} - \boldsymbol{K}_k \boldsymbol{H}_k)^T + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^T \tag{4.37}$$

Here, $\delta \boldsymbol{y}$ is the error between the aiding measurement vector, $\boldsymbol{y}$, and the corresponding estimates, $\hat{\boldsymbol{y}}$. Matrix $\boldsymbol{H}$ is the Jacobian matrix of the measurement functions of $\boldsymbol{y}$ with respect to the error state, $\delta \boldsymbol{x}$. Matrix $\boldsymbol{R}$ is the measurement covariance matrix, as mentioned in Ch.3. Next, the computation of $\boldsymbol{H}$ and $\delta \boldsymbol{y}$ will be thoroughly described in turn as both of these quantities play a central role in the Kalman filter.

4.5.6 The measurement Jacobian matrix $\boldsymbol{H}$

Matrix $\boldsymbol{H}$ is an important component in the ESKF as it represents the contribution of the aiding measurements to the error state and covariance estimates. As mentioned above, matrix $\boldsymbol{H}$ is the Jacobian matrix of the aiding measurements with respect to the error states.

The different measurement functions of the measurement vector $\boldsymbol{y}$ can generically be expressed as

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{w} \quad , \quad \boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{R}) \tag{4.38}$$

where $\boldsymbol{h}(\boldsymbol{x})$ denotes a linear or nonlinear function of the true state and $\boldsymbol{w}$ represent the expected noise component. Having an expression on this form, the Jacobian matrix $\boldsymbol{H}$ is defined as

$$\boldsymbol{H} = \left.\frac{\partial \boldsymbol{h}}{\partial \delta \boldsymbol{x}}\right|_{\delta \boldsymbol{x}=0} \tag{4.39}$$

This represents a linearization of the measurement function about $\delta \boldsymbol{x} = 0$. The computations of the Jacobian matrices of each low rate sensor model will now be investigated. These yield the various rows of the final $\boldsymbol{H}$ matrix used in this state estimation system.

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_{gnss1} \\ \boldsymbol{H}_{gnss2} \\ \boldsymbol{H}_{gss} \\ \boldsymbol{H}_{bl} \end{bmatrix} \tag{4.40}$$

Jacobian of the dual GNSS position measurement function $\boldsymbol{H}_{gnss}$

To compute the various Jacobians, each measurement function is reformulated as a functions of the nominal states and the error states, using the composition equations to substitute for the true states. The measurement function for the dual GNSS position measurements are given in (4.14) and (4.15). Staring from these models, the reformulation is carried out as follows.

$$\begin{aligned} \boldsymbol{y}_{gnss_i} &= \boldsymbol{p}^n_{nb} + \boldsymbol{R}^n_b \boldsymbol{r}^b_i + \boldsymbol{w}_{gnss} \\ &= (\hat{\boldsymbol{p}}^n_{ins} + \delta \boldsymbol{p}^n_{nb}) + \hat{\boldsymbol{R}}^n_{b,ins}(\boldsymbol{I}_3 + \boldsymbol{S}(\delta \boldsymbol{\theta}^b_{nb}))\boldsymbol{r}^b_i + \boldsymbol{w}_{gnss} \\ &= \hat{\boldsymbol{p}}^n_{ins} + \delta \boldsymbol{p}^n_{nb} + \hat{\boldsymbol{R}}^n_{b,ins}\boldsymbol{r}^b_i + \hat{\boldsymbol{R}}^n_{b,ins}\boldsymbol{S}(\delta \boldsymbol{\theta}^b_{nb})\boldsymbol{r}^b_i + \boldsymbol{w}_{gnss} \\ &= \hat{\boldsymbol{p}}^n_{ins} + \delta \boldsymbol{p}^n_{nb} + \hat{\boldsymbol{R}}^n_{b,ins}\boldsymbol{r}^b_i - \hat{\boldsymbol{R}}^n_{b,ins}\boldsymbol{S}(\boldsymbol{r}^b_i)\delta \boldsymbol{\theta} + \boldsymbol{w}_{gnss} \end{aligned} \tag{4.41}$$

with $i \in \{1, 2\}$. From here, the resulting Jacobian with respect to $\delta \boldsymbol{x}$ becomes

$$\boldsymbol{H}_{gnss,i} = \left.\frac{\partial h_{gnss,i}}{\partial \delta \boldsymbol{x}}\right|_{\delta \boldsymbol{x}=0} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & -\hat{\boldsymbol{R}}^n_{b,ins} S(\boldsymbol{r}^b_i) & \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} \tag{4.42}$$

Jacobian of the GSS velocity measurement function $\boldsymbol{H}_{gss}$

The model for the GSS measurements in (4.19), yields a convoluted reformulation and is omitted here due to space constraints. Disregarding the displacement of the GSS with respect to CG, the reformulation in order to achieve a function of $\delta \boldsymbol{x}$ becomes

$$\begin{aligned}
\boldsymbol{y}_{gss} &= \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{R}^n_b \boldsymbol{v}^n_{nb} \right\|_2 + \boldsymbol{w}_{gss} \\
&= \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} (\hat{\boldsymbol{R}}^b_{n,ins}(\boldsymbol{I}_3 + \boldsymbol{S}(\delta \boldsymbol{\theta}^b_{nb}))^T (\hat{\boldsymbol{v}}^n_{ins} + \delta \boldsymbol{v}^n_{nb})) \right\|_2 + \boldsymbol{w}_{gss} \\
&= \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} ((\boldsymbol{I}_3 - \boldsymbol{S}(\delta \boldsymbol{\theta}^b_{nb}) \hat{\boldsymbol{R}}^b_{n,ins}{}^T (\hat{\boldsymbol{v}}^n_{ins} + \delta \boldsymbol{v}^n_{nb})) \right\|_2 + \boldsymbol{w}_{gss} \\
&= \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \hat{\boldsymbol{R}}^b_{n,ins}{}^T (\hat{\boldsymbol{v}}^n_{ins} + \delta \boldsymbol{v}^n_{nb}) - \boldsymbol{S}(\delta \boldsymbol{\theta}^b_{nb}) \hat{\boldsymbol{R}}^b_{n,ins}{}^T (\hat{\boldsymbol{v}}^n_{ins} + \delta \boldsymbol{v}^n_{nb})) \right\|_2 + \boldsymbol{w}_{gss} \\
&= \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \hat{\boldsymbol{R}}^b_{n,ins}{}^T \hat{\boldsymbol{v}}^n_{ins} + \hat{\boldsymbol{R}}^b_{n,ins}{}^T \delta \boldsymbol{v}^n_{nb} + \boldsymbol{S}(\hat{\boldsymbol{R}}^b_{n,ins}{}^T (\hat{\boldsymbol{v}}^n_{ins} + \delta \boldsymbol{v}^n_{nb})) \delta \boldsymbol{\theta}^b_{nb} \right\|_2 + \boldsymbol{w}_{gss} \\
&= \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \hat{\boldsymbol{R}}^b_{n,ins}{}^T \hat{\boldsymbol{v}}^n_{ins} + \hat{\boldsymbol{R}}^b_{n,ins}{}^T \delta \boldsymbol{v}^n_{nb} + \boldsymbol{S}(\hat{\boldsymbol{R}}^b_{n,ins}{}^T \hat{\boldsymbol{v}}^n_{ins} + \hat{\boldsymbol{R}}^b_{n,ins}{}^T \delta \boldsymbol{v}^n_{nb}) \delta \boldsymbol{\theta}^b_{nb} \right\|_2 + \boldsymbol{w}_{gss}
\end{aligned} \tag{4.43}$$

The Jacobian matrix, $\boldsymbol{H}_{gss}$, is derived using the MATLAB Symbolic Math toolbox, and resulted in non-zero partial derivatives with regards to the velocity error, the attitude error and the gyroscope bias error. The full expression of $\boldsymbol{H}_{gss}$ is provided in Appendix C.

$$\boldsymbol{H}_{gss} = \left.\frac{\partial h_{gss}}{\partial \delta \boldsymbol{x}}\right|_{\delta \boldsymbol{x}=0} = \begin{bmatrix} \boldsymbol{0}_3 & \frac{\partial h_3}{\partial \delta \boldsymbol{v}^n_{nb}} & \boldsymbol{0}_3 & \frac{\partial h_3}{\partial \delta \boldsymbol{\theta}^b_{nb}} & \frac{\partial h_3}{\partial \delta \boldsymbol{b}^b_{ars}} & \boldsymbol{0}_3 \end{bmatrix} \tag{4.44}$$

Jacobian of the GNSS baseline measurement function $\boldsymbol{H}_{bl}$

The measurement function for the GNSS baseline vector from (4.21) is reformulated as a function of $\delta \boldsymbol{x}$ below.

$$\begin{aligned}\boldsymbol{h}_{bl} &= \boldsymbol{h}_{gnss2} - \boldsymbol{h}_{gnss1} \\ &= \boldsymbol{R}_b^n(\boldsymbol{r}_2^b - \boldsymbol{r}_1^b) + \boldsymbol{w}_{gnss} \\ &= \hat{\boldsymbol{R}}_{b,ins}^n(\boldsymbol{I}_3 + S(\delta\boldsymbol{\theta}_{nb}^b))(\boldsymbol{r}_2^b - \boldsymbol{r}_1^b) + \boldsymbol{w}_{gnss} \\ &= \hat{\boldsymbol{R}}_{b,ins}^n(\boldsymbol{r}_2^b - \boldsymbol{r}_1^b) - S(\hat{\boldsymbol{R}}_{b,ins}^n(\boldsymbol{r}_2^b - \boldsymbol{r}_1^b))\delta\boldsymbol{\theta}_{nb}^b + \boldsymbol{w}_{gnss}\end{aligned} \tag{4.45}$$

From this expression, the computation of the Jacobian matrix $\boldsymbol{H}_{bl}$ results in

$$\boldsymbol{H}_{bl} = \frac{\partial h_{bl}}{\partial \delta \boldsymbol{x}}\bigg|_{\delta\boldsymbol{x}=0} = \begin{bmatrix}\boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & -S(\hat{\boldsymbol{R}}_{b,ins}^n(\boldsymbol{r}_2^b - \boldsymbol{r}_1^b)) & \boldsymbol{0}_3 & \boldsymbol{0}_3\end{bmatrix} \tag{4.46}$$

4.5.7 Computation of $\delta\boldsymbol{y}$

$\delta\boldsymbol{y}$ is another important component in the ESKF. This quantity is straightforward to compute through a simple subtraction of the estimated aiding measurements from the actual aiding measurements. The resulting difference, or error, is an indicator of the drift that has occurred in the INS block since the previous arrival of low rate aids.

$$\delta\boldsymbol{y} = \boldsymbol{y} - \hat{\boldsymbol{y}} \tag{4.47}$$

In this implementation, it is chosen to include a self-diagnosis step is performed to validate $\delta\boldsymbol{y}$. More specifically, it is attempted to check if the incoming aiding measurement may be an outlier. This is further explained in the next section.

4.5.8 Outlier rejection

Robustness and reliability is crucial in state estimation systems. Any errors present in this system may propagate further down the line and cause performance limitations in the overall perception and navigation capabilities of the vehicle. A focus area during development of the current state estimation system is therefore self-diagnosis. Self-diagnosis in a navigation system refers to the capability of detecting and handling faults and vulnerabilities in its own system.

A major factor undermining the robustness of navigation systems is sensor faults. According to previous team members of Revolve NTNU, a pervasive problem related to the sensor data on Atmos has been sporadic outliers. This is especially relevant for the data streams from the two GNSS recievers (Skibelid, 2019).

To handle this problem, a probabilistic outlier detection method is applied to the state estimation system developed here. An outlier is a measurement that is

regarded too corrupted by errors such that it contains a low amount of valuable information for the state estimation system. Such measurements are likely to cause more confusion than benefit. It is hence advantageous to discard these.

The outlier detection method used here was firstly presented by Brumback and Srinath (1987) and has later had wide usage within navigation systems. It makes use of the innovation covariance and is therefore applicable with any sensor. The idea of this outlier rejection approach is to assess the likelihood of a measurement belonging to the error state distribution, by performing a chi-square (χ^2) hypothesis test. The χ^2-test requires the computation of a quantity $\boldsymbol{S}$, which is a function of both the error state covariance matrix, $\hat{\boldsymbol{P}}$, the sensor noise covariance, $\boldsymbol{R}$ and the Jacobian matrix of the measurement functions with respect to the error state, $\boldsymbol{H}$. This way both the sensor errors and the uncertainty of the states are taken into account in this method. The error state covariance is already calculated in the ESKF relieving some computational cost.

$$\boldsymbol{S} = \boldsymbol{H}\hat{\boldsymbol{P}}\boldsymbol{H}^T + \boldsymbol{R} \tag{4.48}$$

Once $\boldsymbol{S}$ is computed, the chi-squared test (χ^2-test) is performed by checking if (4.49) below hold. $\delta\boldsymbol{y}$ is defined as in (4.47).

$$\delta\boldsymbol{y}^T\boldsymbol{S}^{-1}\delta\boldsymbol{y} < \chi^2(\chi_i) \tag{4.49}$$

If the test in (4.49) fails, the sensor measurement is regarded as an outlier and discarded. The discarding of an outlier can be carried out in several ways. The simplest solution is to discard the entire $\delta\boldsymbol{y}$-vector once one of the members is classified as an outlier. However, this approach is sub-optimal as it throws away numerous non-corrupted measurements leading to loss of useful information. It is therefore here chosen to go with a somewhat different approach.

The implemented approach is as follows. Every time an outlier is detected, the relevant rows in the $\boldsymbol{H}$- and $\boldsymbol{R}$-matrices are cancelled out, meaning that all terms are set to zero. This yields zero contribution from this aiding measurement in the computation of the error state, $\delta\boldsymbol{x}$. This approach hence makes it possible to benefit from all the measurements that pass the χ^2-test, which is desirable. A drawback with this approach is that the matrices in the ESKF may have unnecessarily large dimensions at some time instances.

4.5.9 Injection of the observed error into the nominal state

After the ESKF correction step is performed, the final error state estimates are sent back to the INS block where they are injected into the nominal states due to the composition equations in Table 3.3. These equations are restated below, with $\boldsymbol{q}\{\delta\boldsymbol{\theta}^b_{nb}\}$ computed using (4.4) as previously.

$$\boldsymbol{p}^n_{nb} = \hat{\boldsymbol{p}}^n_{ins} + \delta\boldsymbol{p}^n_{nb} \tag{4.50a}$$
$$\boldsymbol{v}^n_{nb} = \hat{\boldsymbol{v}}^n_{ins} + \delta\boldsymbol{v}^n_{nb} \tag{4.50b}$$
$$\boldsymbol{b}^b_{acc} = \hat{\boldsymbol{b}}^b_{acc,ins} + \delta\boldsymbol{b}_{acc} \tag{4.50c}$$
$$\boldsymbol{q}^n_{nb} = \hat{\boldsymbol{q}}^n_{ins} \otimes \boldsymbol{q}\{\delta\boldsymbol{\theta}^b_{nb}\} \tag{4.50d}$$
$$\boldsymbol{b}^b_{ars} = \hat{\boldsymbol{b}}_{ars,ins} + \delta\boldsymbol{b}^b_{ars} \tag{4.50e}$$

4.5.10 ESKF reset

The final step of the ESKF block is the reset of the error state and covariance estimates. The reset equations implemented in this state estimation system follow the suggestion of Solà (2017).

$$\delta\boldsymbol{x} = 0 \tag{4.51a}$$
$$\hat{\boldsymbol{P}} = \boldsymbol{G}\hat{\boldsymbol{P}}\boldsymbol{G}^T \tag{4.51b}$$

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & (\boldsymbol{I}_3 - \boldsymbol{S}(\frac{1}{2}\delta\boldsymbol{\theta}^b_{nb})) & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix} \tag{4.52}$$

In most implementations of the ESKF the term $\frac{1}{2}\delta\boldsymbol{\theta}$ is approximated to zero reducing the $\boldsymbol{G}$-matrix to an identity matrix $\boldsymbol{I}_{18}$. However, Solà (2017) proposes this reset step as it has proven to increase the accuracy of the covariance matrix, $\hat{\boldsymbol{P}}$.

4.6 Pre-Race state estimation for faster convergence

At the competitions, a short waiting time is expected before every race. During this waiting time, the state estimation system is assumed turned on while the vehicle is at rest. In this stationary state the system is poorly excited, making it harder to

determine the various vehicle states. In order to maximize performance despite the poor excitation, some modifications are applied to the state estimation system for the special case of the vehicle standing still. This is intended to facilitate faster convergence, and eliminate most transients before the race is initiated.

The first modification is to reduce the complexity of the state estimation system when the vehicle is at rest. To maintain observability of the system, the estimation of the gravity error is omitted when the vehicle is stationary. In addition, the aiding from the GSS is omitted as this sensor is highly dependant on motion to provide reliable measurements of the speed of the car.

Secondly, additional aiding by the specific force from the accelerometer is introduced to system. This is the most important modification. The primary intention of including this aid is to improve the stationary attitude estimation. The model for the specific force measurements from the IMU is obtained directly from the accelerometer model in (4.5). The measurement function is hence

$$\boldsymbol{y}_{acc} = \boldsymbol{f}^b_{imu} = -\boldsymbol{R}^{nT}_b \boldsymbol{g}^n_{nb} + \boldsymbol{b}^b_{acc} + \boldsymbol{w}^b_{acc} \tag{4.53}$$

To compute the Jacobian matrix, $\boldsymbol{H}_{acc}$, the same procedure as earlier is repeated. The measurement function is firstly reformulated as a function of $\delta\boldsymbol{x}$. This reformulation is given below.

$$\begin{aligned}
\boldsymbol{y}_{acc} &= \boldsymbol{f}^b_{imu} \\
&= \cancel{\boldsymbol{a}^b_{nb}} + S(\cancel{\boldsymbol{\omega}^b_{nb}})\cancel{\boldsymbol{v}^b_{nb}} - \boldsymbol{R}^{nT}_b \boldsymbol{g}^n_{nb} + \boldsymbol{b}^b_{acc} + \boldsymbol{w}^b_{acc} \\
&= -\boldsymbol{R}^{nT}_b \boldsymbol{g}^n_{nb} + \boldsymbol{b}^b_{acc} + \boldsymbol{w}^b_{acc} \\
&= -(\boldsymbol{I} - S(\delta\boldsymbol{\theta}))\hat{\boldsymbol{R}}^n_{b,ins}{}^T(\hat{\boldsymbol{g}}^n_{nb} + \delta\boldsymbol{g}) + \hat{\boldsymbol{b}}^b_{acc,ins} + \delta\boldsymbol{b}^b_{acc} + \boldsymbol{w}^b_{acc}
\end{aligned} \tag{4.54}$$

In the expression above, it is utilized that the vehicle is known to be stationary such that the true body accelerations and velocities are equal to zero. From this reformulation, it can be seen that the estimate of the attitude appear. This substantiates that aiding from this measurement may improve the attitude estimation.

The Jacobian matrix of the reformulated measurement function has the following appearance.

$$\boldsymbol{H}_{acc} = \left.\frac{\partial h_{acc}}{\partial \delta\boldsymbol{x}}\right|_{\delta\boldsymbol{x}=0} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & -S(\hat{\boldsymbol{R}}^n_{b,ins}{}^T \boldsymbol{g}^n_{nb}) & \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} \tag{4.55}$$

The expression for the corresponding estimated specific forces is similar to (4.53),

with nominal quantities instead of the true.

$$\hat{\boldsymbol{y}}_{acc} = -\hat{\boldsymbol{R}}_{b,ins}^{n\ T}\hat{\boldsymbol{g}}_{nb}^{n} + \hat{\boldsymbol{b}}_{acc,ins}^{b} \tag{4.56}$$

A remark regarding the stationary case, is that the accelerometer bias estimates have reduced observability as a consequence of little excitation. By including the bias in (4.56), it was experienced that some of the errors in the accelerometer bias estimates propagated and had negative effects on the results. Consequently it was attempted to neglect this bias in both (4.53) and (4.56). This is a reasonable simplification under the assumption that the gravity is a significantly larger signal than the accelerometer bias. This adjustment improved the performance and is chosen in the final design. Hence, the final expressions for $\boldsymbol{y}_{acc}$, $\hat{\boldsymbol{y}}_{acc}$ and $\boldsymbol{H}_{acc}$ become slightly modified and are stated below.

$$\boldsymbol{y}_{acc} = -\boldsymbol{R}_{b,ins}^{n\ T}\boldsymbol{g}_{nb}^{n} + \boldsymbol{w}_{acc}^{b} \tag{4.57}$$

$$\hat{\boldsymbol{y}}_{acc} = -\hat{\boldsymbol{R}}_{b,ins}^{n\ T}\hat{\boldsymbol{g}}_{nb}^{n} \tag{4.58}$$

$$\boldsymbol{H}_{acc} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & -S(\hat{\boldsymbol{R}}_{b,ins}^{n\ T}\boldsymbol{g}_{nb}^{n}) & \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} \tag{4.59}$$

Apart from these modifications, the rest of the pre-race state estimation system follow the design presented throughout this chapter.

5 Results and Discussion

This chapter presents the results of the state estimation system developed here. The aim of this chapter is to examine the final performance and map the current strengths and weaknesses of the developed system. The chapter is divided into three main sections, structured as follows. The first section presents the results with the final design of the developed state estimation system. In the second section, it is decided to investigate the effects of some selected design choices and evaluate their necessities. Finally, the third section studies the contribution of each aiding measurement to the overall performance of this state estimation system.

The result of this state estimation system is graphically represented through a fairly large set of plots. For every aspect examined in this chapter, a new set of plots are deemed convenient to complement the discussion. This yields an excessive number of plots. Hence, for a tidier dissemination of the topics of this chapter, all plots not related to the final results are placed in appendices.

5.1 Results of the final design

5.1.1 Validation with simulated ground truth

The validation of this state estimation system is based on the case study presented in Ch.4.1.3, focusing on the Skidpad event at the FS competitions. Figure 5.1 show the position estimates from the final state estimation system, mapped onto the horizontal plane. The estimates are plotted against the true positions from the simulator mapped onto the same two dimensional plane. In this plot, the simulated Skidpad track layout is visible.

Figure 5.2 - 5.6 graphically illustrate the final overall performance of the developed state estimation system. Each figure concerns each of the vehicle states and illustrates the performance of the state estimations system using two different plots. In the first of these plots, the estimated states are plotted against ground truth, obtained from the simulator. The second plot is an error plot showing the deviation between the estimated states and ground truth.

In the plotted results, the following values are used for the tuning parameters $\boldsymbol{R}$ and $\boldsymbol{Q}$.

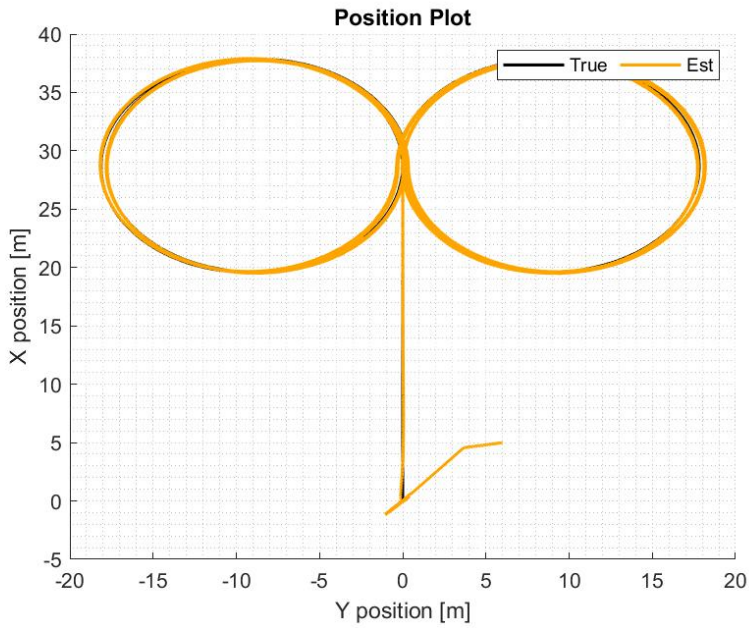


Figure 5.1: The true and estimated positions mapped onto the two dimensional horizontal plane.

$$\boldsymbol{R} = \text{diag}(\begin{bmatrix} 4, & 4, & 4, & 4, & 4, & 4, & 1, & 8, & 8, & 8, & 1, & 1, & 1 \end{bmatrix}) \tag{5.1}$$

$$\boldsymbol{Q} = \text{diag}(\begin{bmatrix} 0.001, & 0.001, & 0.001, & 1, & 1, & 1, & 0.01, & 0.01, & 0.01, & 0.001, & 0.001, & 0.001 \end{bmatrix}) \tag{5.2}$$

diag(·) represent the diagonal of the two square matrices. Due to time constraints, a limited amount of time is put into tuning of this state estimation system. Increased performance may therefore be achievable with a more comprehensive tuning process.

To properly validate the convergence of the state estimation system, multiple states are initialized with random values that differ from the true initial values. As shown in the various plots, the vehicle remains stationary for some time in the beginning of the simulations. During this period of stationary motion, the modified pre-race state estimation system is validated.

Figure 5.7 show the gravity estimation due to the inclusion of the gravitational error state in the state vector of the ESKF block.

5.1.2 RMS error

As a measure of accuracy, the root mean square (RMS) errors are computed and presented in Table 5.1. The transients in the very beginning of several of the plots, are omitted in these calculations. To better the ability to analyze the different levels of excitation, four different RMS values are computed and presented in Table 5.1.

In the first column from the left, the RMS errors capture the time period when the vehicle is standing still and the pre-race state estimation system acts on the vehicle. In the second column, the RMS errors are calculated for when the car is driving the pit lane. Thirdly, the RMS errors capture the period when the car is driving the eight-figured Skidpad track. Finally, in the rightmost column, the total RMS values are provided, containing contributions from the former three columns.

It should be noted that the time spent in the three different cases vary considerably. The amount of time spent in the various cases correspond to their weighting in the total RMS values. The time spent in the pit lane, is for instance significantly smaller than the other two cases, yielding very little contribution to the total RMS. The time period at rest before the start of the race lasts very long compared to the pit lane, but still only approximately half of the time spent in the Skidpad track.

5.1.3 Discussion of the results with the final design

From the graphical representations of the results along with the RMS errors, the performance of this state estimation system can be discussed. The discussion is structured such that each of the three case mentioned above, representing three different levels of excitation, are discussed separately. Firstly, the performance of pre-race state estimation system acting on the vehicle at stand-still is investigated.

Performance at stand-still

From Figure 5.2 - 5.6 it is observed that the estimations of the position, velocity, and gyroscope bias, converge towards the true values while the vehicle remains stationary. This is evident when observing the error plots, where it is shown that the deviations from the true states rapidly move towards the zero-reference and stay close to zero for the rest of the pre-race waiting time.

Considering the attitude estimation in the stationary condition, a small deviation between the estimated states and the true states is present for the roll and pitch angles. Here, the error plots provide a better basis for analysis. The magnitudes of the deviations are revealed by the error plots, and are approximately 3° for the roll angle and 2° for the pitch angle. The reason for these deviations are not trivial, but may be related to the poor performance of the accelerometer bias estimation at stand-still, which is further discussed later. These deviations are

Table 5.1: RMS errors of the final state estimation system

	Axis	RMS				Unit
		Stand-still	*Pit Lane*	*Skidpad*	*Total Track*	
Position (NED)	x	0.0016	0.0034	0.0075	0.0046	[m]
	y	0.0025	0.0032	0.0065	0.0040	
	z	0.0003	0.0010	0.006	0.0004	
Velocity (NED)	x	0.0001	0.0096	0.0106	0.0065	[m/s]
	y	0.0001	0.0120	0.0111	0.0068	
	z	0.0000	0.0024	0.0036	0.0022	
Accelerometer Bias (BODY)	x	0.1647	0.0262	0.0172	0.0528	[m/s^2]
	y	0.2940	0.3938	0.0072	0.0969	
	z	0.0144	0.0164	0.0015	0.0052	
Attitude (BODY)	roll	0.1837	0.1765	0.0047	0.0600	[°]
	pitch	0.0980	0.0167	0.0024	0.0299	
	yaw	0.0237	0.1313	4.9828	2.9422	
Gyroscope Bias (BODY)	roll	0.0000	0.0000	0.0078	0.0046	[°]
	pitch	0.0001	0.0054	0.0043	0.0040	
	yaw	0.0023	0.0002	0.0037	0.0028	

however small fractions of the total range of these states. With regards to the poor level of excitation the accuracy of these estimations are considered satisfactory.

The pre-race estimation of the yaw angle stand out from the other two angles. Looking at the relevant error plot in Figure 5.5, an oscillating behaviour is observed. The oscillations are however bounded and the deviations stay within a range of ±5° from the zero reference. From Figure 5.5 it is also possible to conclude that the mean of this estimation during the stationary condition is closer to zero than for the roll and pitch angles.

As mentioned, the accelerometer bias estimation also lacks capability of converging towards the true bias values while the vehicle is standing still. Figure 5.4 show that a deviation is present for all three biases, with different magnitudes. The deviations have magnitudes of approximately $0.41 m/s^2$, $0.54 m/s^2$ and $0.12 m/s^2$ for the x-, y- and z-directions respectively. These are relatively large in the x-direction and y-direction. The presence of these deviations supports the reasoning behind why the accelerometer bias is omitted in the aiding from the accelerometer specific force measurement in the pre-race state estimation system.

An additional overall observation from the various plots is that the position, velocity and accelerometer bias estimation have initial transients. However, these oscillations decay quickly and a nearly stationary value is reached in approximately 10 seconds, while the vehicle remains still. These transients therefore do not introduce any problems during the race.

Performance during pit lane

The stationary period, waiting for the race to start, lasts for a little bit more than one minute in the simulations. After this, the vehicle enters the pit lane, which is a 30 meter long straight line going perpendicular towards the figure eight shape of the Skidpad track. The pit lane marks the transition from a stationary condition to dynamic motion. At this point, all the vehicle states except the attitudes and the accelerometer biases have converged to a value in the vicinity of ground truth. These states remain close to their true values during the driving of the pit lane. One general observation is that the error plots now show a slightly more oscillatory behaviour about the zero reference line. This is probably due to increased complexity in the motion, leading to increased effects from the uncertainties. The states are still estimated with high accuracy.

Furthermore, it is interesting to consider the states where convergence towards ground truth was not achieved at stand-still. Figure 5.4 shows that the translational movement in the pit lane enables the state estimation system to converge towards the true accelerometer bias value in the x-direction, while the deviations related to the other two directions remain at the same level. This is better revealed by the RMS errors in Table 5.1. The direction of travel coincides with the direction of improved bias estimation. For the attitude estimates, Figure 5.5 shows that no significant improvements are present for the roll and yaw angles. The deviation in pitch angle, on the other hand, rapidly diminishes. Also here, the RMS errors in Table 5.1 provide a clearer image of the improvements. The pitch angle represent the rotation about the x-axis, also coinciding with the direction of travel.

Here, it is possible to establish a link between the improvements in both the accelerometer bias and the attitude estimation, and the direction of driving. Driving in the x-direction led to convergence of the accelerometer bias in the same direction as well as convergence of the vehicle's angle about that same axis. This observation also underpins the previously made statement saying that the poor accelerometer bias estimation at stand-still may have been the reason for the constant deviations present in the attitude estimates.

Performance during Skidpad

When the vehicle enters the Skidpad track, the complexity of the motion increase even further, and all vehicle states are now greatly excited. This part of the

simulations is the most important one, as it represents the actual competition, in which the performance of the race car is measured against the other participating vehicles.

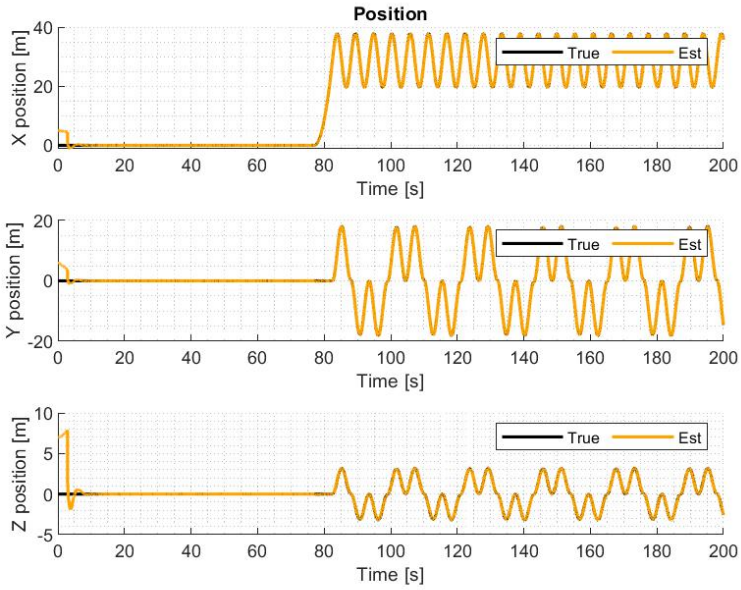
Prior to this part of the simulations, all vehicle states have converged close to their true values expect for the accelerometer biases in the y- and z-directions and the roll angle. However, due to the sufficient excitation of the system, the deviations present in these states are eliminated in few seconds. The already converged states also remain close to ground truth during the driving of the Skidpad track. All desired vehicle states are hence estimated with high accuracy during the entire Skidpad event.

Observing the Skidpad-column in Table 5.1, all RMS errors remain at a low level. It is however noticeable that the errors in the yaw angle estimation stay at a higher level than the other errors. The yaw angle faces large variations during the Skidpad track due to the layout of the track, which highly excites the whole range of this state. Extensive dynamics lead to larger uncertainties when information about the relevant state is available at specific time instances only. The results are hence not surprising. Looking at the error plot in Figure 5.5 the yaw angle error has a magnitude below 3° during the entire Skidpad run, and the mean is close to zero. This result is therefore still satisfactory.

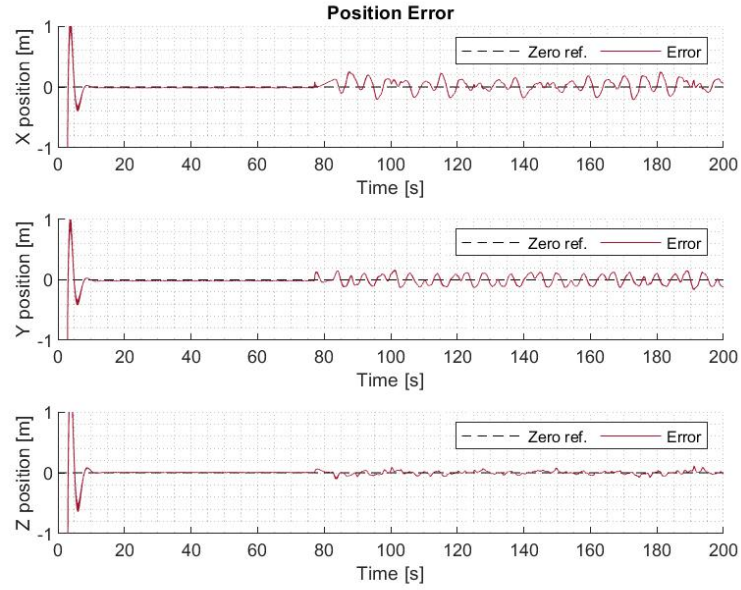
Overall performance of final design

To summarize, the state estimation system developed here is capable of determining all the desired vehicle state with high accuracy, given that the motion of the vehicle sufficiently excites the various states. The main aim of this system is to obtain an accurate determination of the PVA, and the bias estimates themselves are therefore of lower interest. However, as observed from the results, the different vehicle states are closely connected and accurate bias estimations are hence necessary to obtain the desired accuracy in the PVA estimation.

From the results it is observed that this state estimation system manages to accurately achieve both the positions and the velocities while the vehicle is stationary. In other word, the state estimation performs well for these states, even without excitation of the system. The attitudes are estimated with slight deviations when the vehicle is stationary. This is likely to be a consequence of poor accelerometer bias estimation. Once the vehicle starts turning, the attitude estimates rapidly converge towards their true values along with the accelerometer bias estimates.

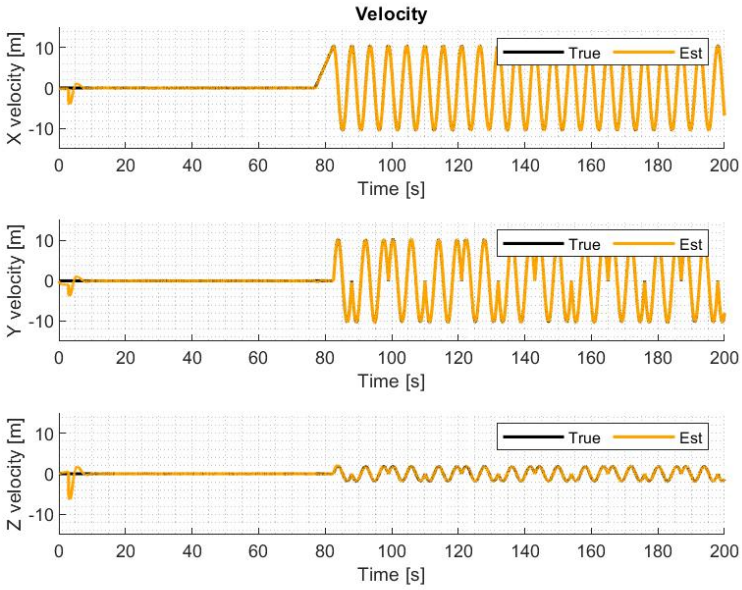


(a) Position estimates vs true state

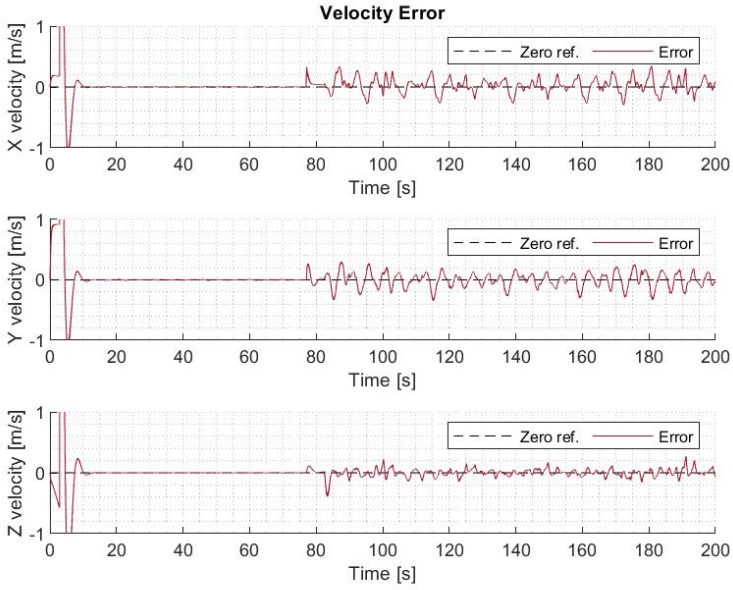


(b) Error plot

Figure 5.2: Results of position estimation

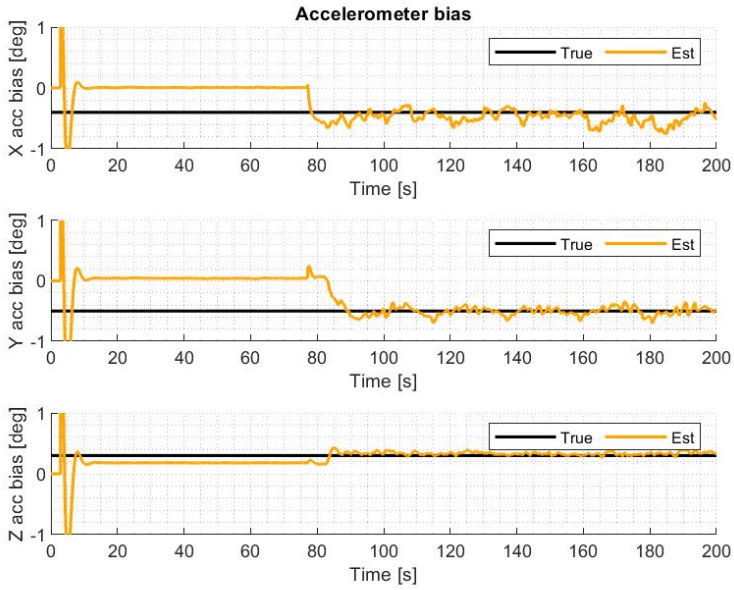


(a) Velocity estimates vs true state

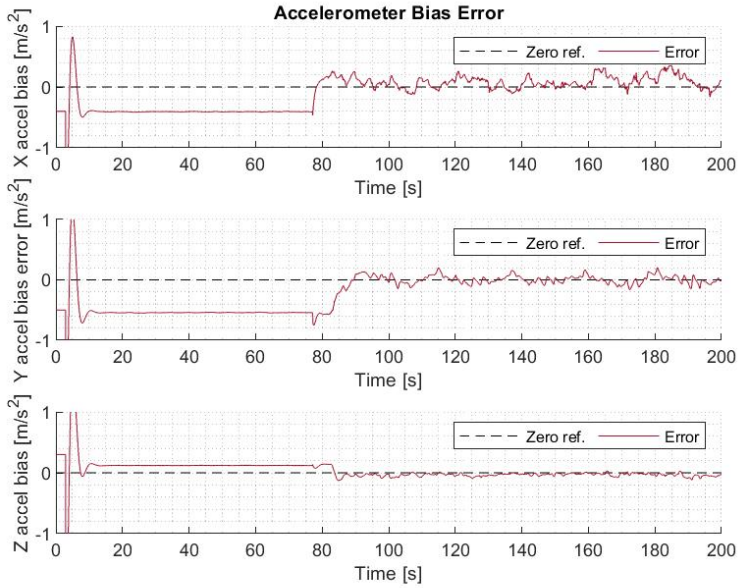


(b) Error plot

Figure 5.3: Results of velocity estimation

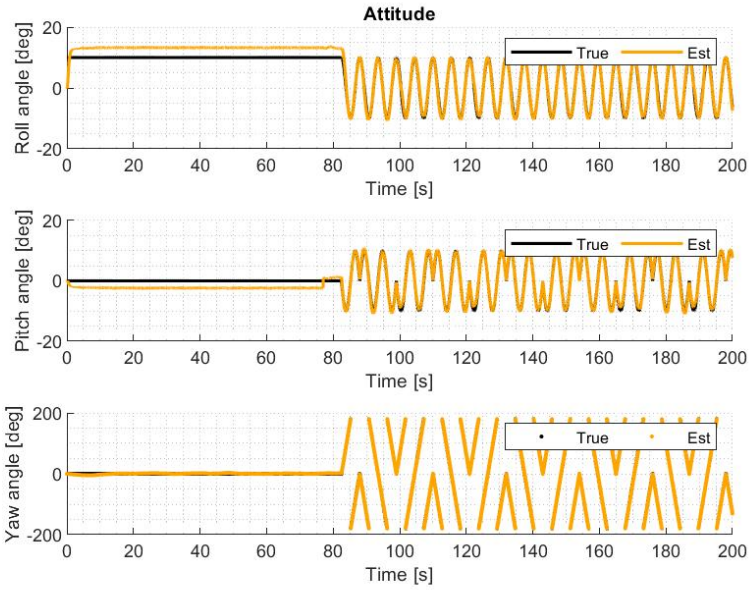


(a) Accelerometer bias estimates vs true state

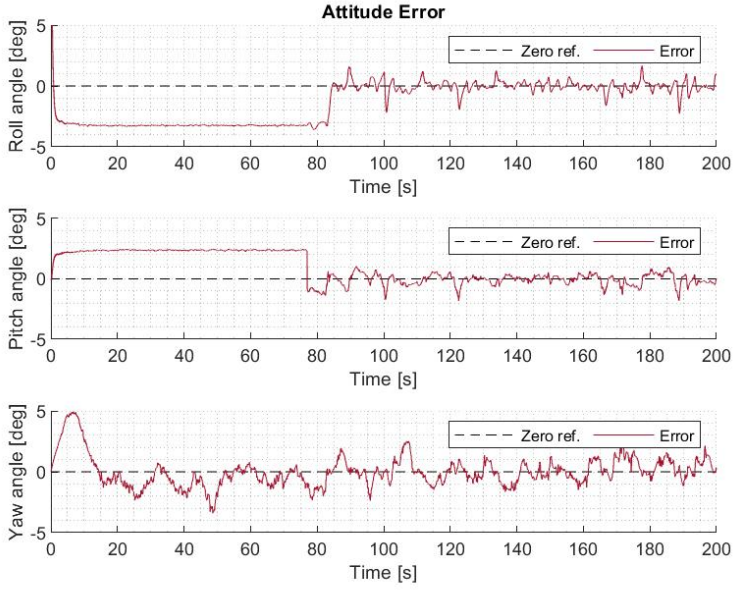


(b) Error plot

Figure 5.4: Results of accelerometer bias estimation

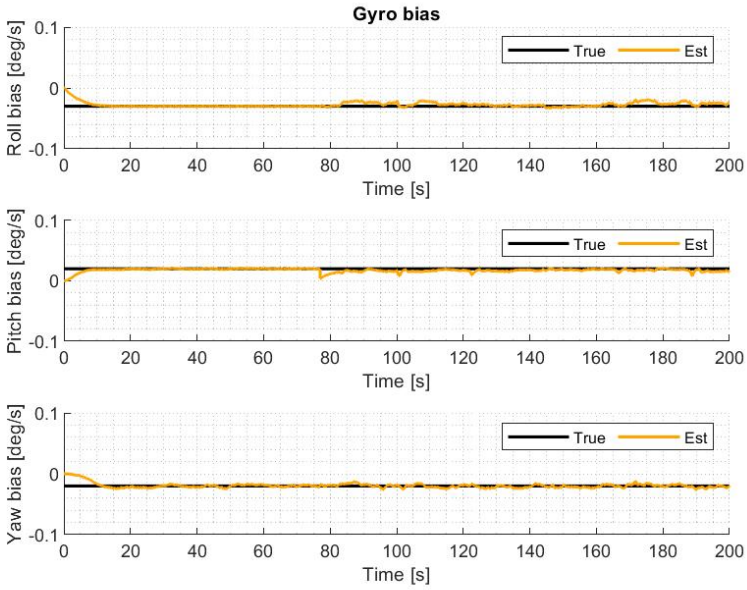


(a) Attitude estimation vs true state

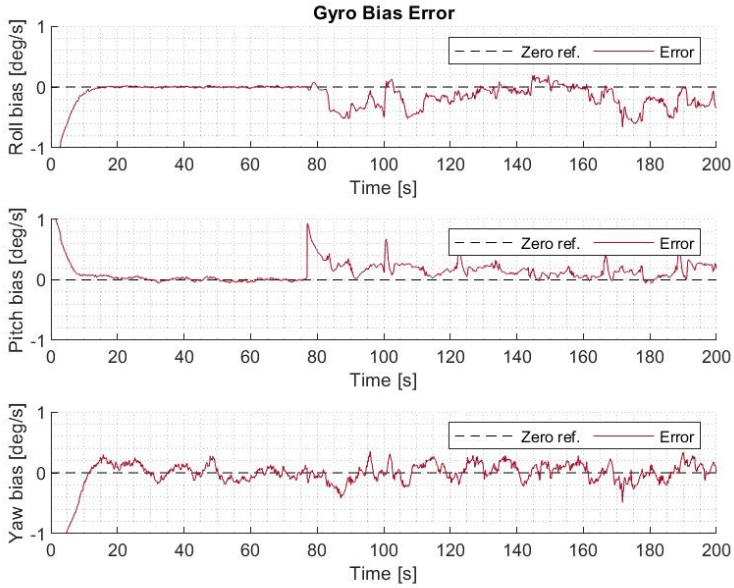


(b) Error plot

Figure 5.5: Results of attitude estimation



(a) Gyroscope bias estimates vs true state



(b) Error plot

Figure 5.6: Results of gyroscope bias estimation

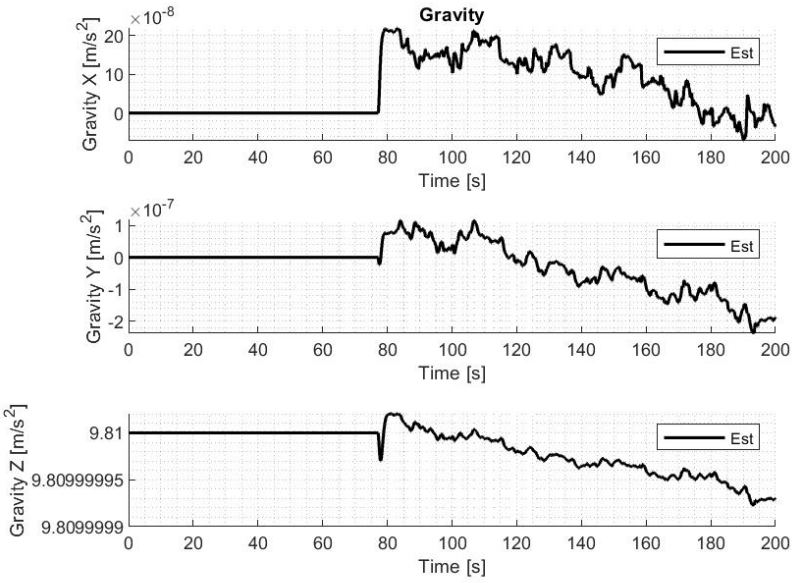


Figure 5.7: Results of gravity estimation

5.2 Investigation of selected design choices

For this section it is desired to examine two essential design choices. Firstly, the pre-race modifications are reviewed. Here, one of the subjects of analysis is the exclusion of the accelerometer bias from the measurement functions for the specific forces. This exclusion represents a relatively radical simplification, and is hence an interesting topic to review. In addition, the effects of having pre-race modifications at all will be examined. The second design choice investigated in this section is the gravity estimation, where it is of interest to determine its necessity.

5.2.1 Omittance of the accelerometer bias in the specific force aiding

In the previous chapter, it is put forward that the accelerometer bias is omitted in the design of the specific force aiding, introduced in the pre-race state estimation system. As discussed above, the accelerometer bias estimation shows mediocre performance in the stationary condition, and a deviation between the estimated states and the true states is present in all three directions. These deviations introduce a risk of error propagation, that may harm the performance of the accelerometer aiding.

Appendix E contains the results when the accelerometer biases are included in the measurement models used to implement the specific force aiding. The limits of the y-axes in these plots are kept similar to Figure 5.2 - 5.6 for easier comparison.

From the comparison between these sets of plots, the most prominent differences are observed in the estimations of the attitudes and the accelerometer biases. Both of these states are subject to significantly larger deviations and less consistency than in the final design, yielding a reduction in performance of the state estimation system.

For the remaining states, namely the positions, velocities and gyroscope biases, convergence towards ground truth is still achieved. However, it takes more time for the estimates to converge and a slightly more oscillating behaviour is present in the beginning of the simulations.

From this, it is confirmed that the omitting of the accelerometer bias improves the performance of the pre-race state estimation system, making this an expedient simplification.

5.2.2 The pre-race modifications

Next, it is of interest to compare the performances with and without the entire pre-race modifications. In the plots presented in Appendix F, the pre-race modi-

fications are removed such that the state estimation system described in the first part of Ch. 4 acts on the vehicle during the entire simulation.

In general, a significant reduction in the performance is observed. When the pre-race modifications are omitted, high frequency transients with large amplitudes are present in the estimation of several of the states. In addition, the attitude and accelerometer bias estimations struggle to determine the correct states leading to great deviations between the estimated states and ground truth.

The poor performance during the stationary condition of the vehicle also seem to propagate into the first period of dynamic motion. The deviations shown in the various error plots remain large for several laps of the Skidpad track. Eventually, the state estimation system is capable of retrieving a somewhat sufficient performance. The overall performance is however highly affected in a negative way by withholding the pre-race modifications.

5.2.3 Gravity estimation

In the final state estimation design, the error state vector in the ESKF block is expanded with one additional state compared to the nominal state vector. The additional state is the gravity error state, included in order to more precisely realize the dynamics of the velocity error state. However, as mentioned in Ch.4 this is not a necessity.

A common alternative approach is to exclude this state in the error state vector and rather treat it as a part of the accelerometer bias. Appendix G show the results when this alternative approach is realized. The pre-race modifications are reintroduced in these plots such that the only difference from the final design is the exclusion of the gravity estimation.

The plots show that exclusion of this additional error state yields a result that is nearly identical to the results of the final design. From this, it is concluded that having the gravitational error state adds little value to the system performance. Instead, it adds complexity and computational cost. The estimation of the gravitational error is omitted in the pre-race state estimation system. Due to the discoverings of this section, this additional error state might as well have been omitted entirely.

5.3 Which aiding measurements are most important?

In this section it is desired to unfold which aiding sensors are the most critical ones. It is also of interest to investigate if there are any aiding sensors that have an insignificant contribution to the overall performance of this state estimation

system. To study the individual contributions, the different low rate aiding sensors are removed from the state estimation system one by one. The most interesting discoverings from this analysis are presented in the following.

5.3.1 Dual GNSS most important

From the mentioned investigation it is found that the dual GNSS aiding is the most critical low rate aiding. In this study, the aiding by the baseline vector between the GNSS receivers and the aiding from the GSS was first removed individually and then removed both together. All the three cases gave very similar results.

The figures in Appendix H show the performance of the state estimation system when only the GSS is removed from the sensor configuration. From here it is seen that the removal of this sensor has little effect on the performance, which is almost identical to when the GSS is included. The necessity of this sensor, considering the hardware complications and increase of run-time it may introduce, may therefore be questionable. However, for the target application redundancy is very important and increases robustness of the system. If the GNSS loses signal during a race, it is important to still have other available aiding measurements.

Appendix I provides the results when the baseline measurement is removed as an aiding measurement. Also here, the performance is almost identical to the final design, and the dual GNSS position measurements seems to carry most of the responsibility for the desirable performance. In this case, the aiding baseline measurement arrive from the same sensor as the GNSS position measurements. This aiding measurement does therefore not provide redundancy in the same way as the GSS.

Hence, the conclusion is that the satisfactory performance of the final state estimation system is mainly obtained from the contribution of the dual GNSS position aiding. The two remaining aiding measurements adds little value to the state estimation system, as long as the dual GNSS position measurements are available. If the GNSS signals fall out during a race, the GSS measurements provide critical redundancy.

5.3.2 Reduced performance from baseline measurement

Another highly interesting discovering from this investigation is the effect on the yaw angle and yaw bias estimations induced by the aiding from the baseline vector. Comparison of the plots in Appendix I and the final results in Figure 5.2 - 5.6, surprisingly shows that both of these estimations are improved. When the baseline aiding is removed from the state estimation system, the estimations of the yaw angles and yaw biases become smoother and have remarkably lower variances. The errors between the true and the estimated states stay closer to

zero in both of estimations. Excluding this measurement therefore increases the performance in addition to reduced complexity and run-time.

The reason for this results is hard to determine with certainty. The lever arms used to model the displaced GNSS position measurements already provide observable heading. Hence, it can be questioned if this additional measurement of the baseline alone has a conflicting effect. It is also possible that this result is related to insufficient tuning of the measurement noise covariance matrix, $\boldsymbol{R}$.

5.3.3 Poor performance with single GNSS

The dual GNSS consists of two displaced position measurements, with different lever arms. It is important to notice that it is both of these measurements together that yield the desired performance. During the analysis of this section, the performance of this states estimation system is also tested when only one of the GNSS measurements are available.

Having only a single GNSS receiver severely reduces the performance of this state estimation system, as shown in Appendix J. In the plots in this appendix, the ground speed measurements are reintroduced. The baseline vector is not included in this scenario as it is dependant of both GNSS receivers and cannot exist when one of the GNSS receivers are removed. With aiding from only one GNSS position measurement, the system is capable of determining all the desired vehicle states for a limited period of time, but starts to diverge after approximately 70 seconds of simulated vehicle motion.

The state that seems to diverge first is the attitude. This is foreseeable as none of the low rate aids provide direct measurement of the heading anymore. Now that one of the GNSS antennas is removed, the observability of the attitude follows along. As already observed, inaccuracies in one state quickly propagate to other states in the state estimation system. It is hence important to facilitate high performance in the estimation of all states with use of appropriate low rate aids. It is also of concern if the implemented outlier rejection step increases the probability of divergence. If some of the sensor signals become unavailable for a short period of time this may mislead the χ_2-test to classify non-corrupted measurements as outliers when the lost sensor signals are retrieved.

6 Conclusion

In this thesis, an aided inertial navigation system is developed for an autonomous Formula Student race car. The system estimates the PVA of the vehicle combining information from high rate inertial sensors along with dual GNSS and radar-based ground speed measurements arriving at a lower rate. This is carried out with the use of an indirect multiplicative extended Kalman filter (MEKF), based on a purely kinematic model. The system is validated offline in a simulated environment where the expected excitation of the vehicle at the racetrack is applied. The simulated input to the state estimation system is disturbed by a stochastic bias model and additive Gaussian white noise.

Despite the perturbations and extensive excitation of the vehicle, the state estimation system is able to robustly estimate the desired vehicle states with fast convergence and rejection of outliers. Central design choices are examined as well as the contribution of the various sensors. Here, it is found that simplifying the state estimation system and including additional accelerometer aiding during stand-still has a significant positive effect on the performance. On the other hand, the estimation of the gravity during dynamic motion has negligible effects on the results. Concerning the contribution of the various aiding sensors, the dual GNSS position measurements stand out as the most important. The baseline vector and ground speed measurement have little influence on the system performance, but the latter provide useful redundancy.

Even though the results from the offline validation show promising potential, the results are not obtained from experiments entirely representable for what the race car is subject to in an actual race. Such experiments was unfortunately restricted due to the COVID-19 pandemic. Online validation with real-time sensor measurements therefore still remains and may introduce challenges not accounted for by the current system. Further testing and implementation is thus necessary before the developed state estimation system can be seen as a viable product for the intended use.

7 Future Work

The state estimation system has been developed from scratch in this thesis. The developed system shows promising potential but is still subject to vulnerabilities and should be further developed. The aim of this chapter is to provide some suggestions of reasonable next steps in the further development of this state estimation system.

7.1 From offline to online estimation

This state estimation system is developed in a simulated environment and validated offline with simulated data streams. Furthermore, this system should be implemented to run in real-time, using measurements directly from the sensors. Integration with hardware may introduce errors not currently handled in the state estimation system. These errors may root from expected error sources with a larger impact than foreseen, or they may arise from unexpected error sources. Examples of expected errors are uncertainties in the measurements of the lever arms and misalignments due to the mounting of the various sensors.

It may be possible to reduce the impact of these additional error sources by putting an increased effort into tuning of the system parameters. As of now, the time spent on tuning is moderate. Finer tuning may in any case be useful in order to fetch as much as possible of the potential performance of this state estimation system.

7.2 Compensation of measurement delays

A property of the Kalman filter approach is that it can only take into account measurements from the current state (Valls et al., 2018). In many practical systems there is a delay in some of the sensors. These introduce problems with the current design. Another suggestion for further work is therefore to compensate for these measurement delays and prevent them from limiting the performance of the state estimation system.

A trivial solution to this problem is to keep a buffer of previous state distributions and measurements. At every iteration the state is then propagated forward and corrected with all newer measurements up to the current time. This solution is simple in principle, but has a high computational cost.

An approximate approach to this solution is proposed by Valls et al. (2018) and proven successful for a similar application. The idea is to use the Kalman filter to estimate the states up until the most delayed measurement. All measurements

newer than the most delayed one are then taken into account by executing a steady state approximation of the extended Kalman filter (SSKF).

The SSKF is a simplification of the EKF where the following assumptions are made for the time interval from the most delayed measurement up until the current time:

- Constant (or slowly varying) covariance matrix
- Close to linear measurement models
- Stationary measurement noise and process noise

These assumptions leads to a constant Kalman gain, avoiding the matrix inversion step and the need for calculating the covariance matrix. This approach represents a trade-off between the accuracy of the EKF and the run-time of the SSKF. Alternative solutions for handling delays in discrete Kalman filters are proposed by Larsen et al. (1998).

7.3 Iterative extended Kalman filter

In Chapter 3, a method for detecting outliers in the measurement data using a χ_2-test is presented. Once an outlier is detected, there are several possible ways to discard this measurement, as already mentioned. The current approach is to cancel all terms in the measurement Jacobian matrix $\boldsymbol{H}$ related to the outlier. This leads to zero contribution from this measurement in the rest of the KF steps where the error state is computed. A disadvantage here is that the dimensions of the matrices in the filter become unnecessarily large, leading to unnecessary computational cost, especially since the Kalman filter includes a matrix inversion step.

Another possibility is to input the aiding measurements individually into the ESKF block, unless they are classified as outliers. All the steps of the KF are then performed for each measurement individually. This design is recognized as an iterative extended Kalman filter (IEKF). In the IEKF, an error state vector is computed for each aiding sensor separately and the error state covariance matrix is updated accordingly. This way the properties of each sensor are better captured in the covariance matrix.

Performing the KF steps for each sensor in the system may seem computationally costly. However, the matrices become remarkably smaller with this approach. Hence, the costs related to the increased number of KF iterations are balanced out by the reduction of cost per iteration. Smaller matrices yield a lower computational cost of the KF calculations, especially the calculation containing a matrix inversion.

The difference in the total computational cost between the iterative approach and the standard approach is therefore small and not of significant concern.

The iterative approach may improve the performance of the state estimation system due to more frequent error state generation and injection. In addition, the continuous update of the covariance matrix is advantageous. This is therefore also suggested as a topic of research for future work.

7.4 Modification of sensor configuration

From the results of this thesis, it is concluded that the dual GNSS aiding is the most critical. If one of the GNSS antennas falls out, the performance is radically reduced, and the state estimation system is only capable of determining all the states for a short period of time.

It is well-known that GNSS receivers are prone to environmental disturbances. Loss of GNSS frequently occur due to limited sight of satellite signals or signal interference. It is therefore expected that periods of weak or no access to GNSS data may be present during a competition race. A suggestion for future work is therefore to include other sensors that can provide redundancy together with the GSS, and compensate for the lost performance if one or both of the GNSS antennas fall out. This would increase robustness and reliability.

The results also showed that not all states are capable of converging when the vehicle is stationary. Here, it is interesting to research alternative sensor configurations that may enable entire convergence before the start of a race. The currently developed system facilitates this research by providing a simulation environment where it is uncomplicated to include various sensor models and achieve an indication of their contribution to the overall performance. For instance, it could be interesting to check if introduction of a compass to the sensor configuration might improve the attitude estimation.

7.5 Fusion with a separate LiDAR-based state estimation pipeline

The last suggestion for future work is aimed at a long-term perspective. It is inspired by the state estimation solution developed by a research group of the Technical University of Munich (TUM) taking part in the Roborace competition in 2019 (Wischnewski et al., 2019). The clever solution of this research group can be regarded as an extension of the state estimation solution developed here. The idea is to fuse two separate state estimation pipelines. One pipeline is based on GPS, like the one developed here, and the second pipeline is LiDAR-based. The fusion

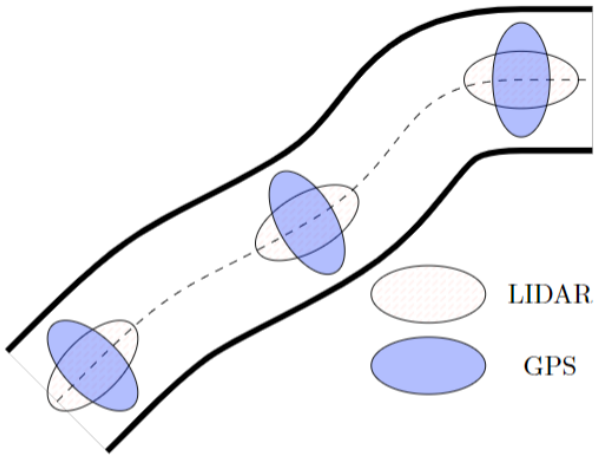


Figure 7.1: Comparison of the covariance properties of a LiDAR vs. GPS

is carried out with a Kalman filter.

The key advantage of this solution is the exploitation of the covariance properties of each sensor source. Figure 7.1 shows how the uncertainty ellipses for LiDAR and GPS related to three different vehicle positions may be resembled. As shown, the LiDAR covariance in the lateral direction is smaller than the GPS covariance. In the longitudinal direction, the opposite holds. Hence, fusion of these two state estimation pipelines can be carried out with adaptive weighting based on their covariances.

A similar solution requires application of advanced concepts of control theory and is believed to be an exciting challenge for the future. This suggestion is included both to illustrate a potential direction to carry the developed state estimation system, and to encourage further development in general.

Atmos Sensor Configuration

A

Atmos is configured with a high number of sensors in order to monitor various vehicle states. To enable autonomous perception and navigation the most relevant sensors are the INS combining GNSS and IMU, the camera, the two LiDARs and the optical encoders used for the four wheels and for the steering wheel. In this appendix, a more detailed explanation of the mentioned sensors and their usages are provided.

A.1 VectorNav VN-300 dual antenna GNSS/INS

The VectorNav VN-300 is a combined sensor consisting of an IMU and two GNSS antennas. In general, an IMU is an integrated sensor package combining multiple accelerometers, gyroscopes and sometimes additional inertial sensors to produce a 3-dimensional measurement of both specific force and and angular rate. Specific force is a measure of acceleration relative to free-fall. The actual acceleration of the vehicle can be extracted from the specific force by subtracting the gravitational acceleration. Angular rate is a measure of the rate of rotation. (VectorNav, 2020)

The VectorNav VN-300 utilizes the latest of micro-electro-mechanical systems (MEMS)-based inertial sensors famous for being low in cost and high in accuracy. VN-300 incorporates a 3-axis accelerometer, a 3-axis gyroscope, a 3-axis magnetometer and a barometric pressure sensor making up a 10-axis MEMS IMU (VectorNav, 2020).

The two GNSS recievers follow the U.S. system frequently known as GPS. This is an Earth-satellite based navigation system made up of a synchronized network constellation of at least 24 satellites placed in the Earth's orbit by the U.S Department of Defense (DoD). The satellite signals can be processed in a GPS receiver to estimate its current position and velocity. Since these states are determined di-

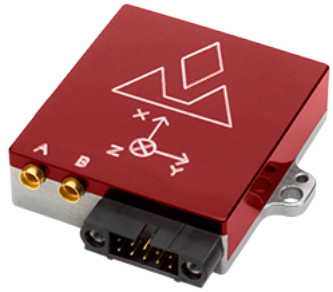


Figure A.1: The Vectornav VN-300 INS

rectly, the GPS receivers are not dependant of initial conditions or previous states. They are therefore considered as stand-alone sensors. However, these receivers commonly suffer from time delay and disturbances in the environment shadowing for the satellite signals and preventing these sensors from providing navigation data at a high enough rate for e.g. automotive applications.

The VN-300 incorporates two on-board 72-channel, L1, GPS receivers (VectorNav, 2020). Having two such antennas enable estimation of true heading with respect to true north in both static and dynamic conditions.

This sensor is the main information source for the navigation system being the subject for this thesis. Combining data from IMU and GNSS are in many applications sufficient for robust state estimation. The VectorNav VN-300 already contains an on-board filter merging data from its two sensor sources, in order to filter out as much noise as possible and provide navigation data. However, experience from earlier team members witness that the navigation solution provided by VectorNav still suffers from drift due to noise (Skibelid, 2019).

A.2 Pegasem GSS15 radar-based ground speed sensor

The Pegasem radar sensors provide non-contact ground speed sensing. This compact and light-weight sensor scans the road surface with a 24 GHz radar beam outputting 100 pulses per meter. The high gain narrow beam antennas create Doppler signals with good noise margin, allowing measurements even in very low speed ranges. The raw Doppler signals are converted to an analogue speed voltage by the internal processor of the sensor (Pegasem, 2019). The GSS is included as a low rate aiding measurement in the aided inertial navigation system developed in this thesis. Figure A.2 shows an image of the sensor mounted on the back of a vehicle. Due to the COVID-19 outbreak this sensor did not arrive from the manufacturer, as originally planned.

Figure A.2: The Pegasem GSS15 ground speed sensor.

A.3 Basler acA1300-200uc USB 3.0 camera

Atmos is equipped with two digital stereo cameras, manufactured by Basler. These are used for the development of a stereo camera system for where the cameras attempt to retrieve 3D information from the 2D scene. The desired 3D information is the presence of cones in the vehicle's environment, their color and their position relative to the vehicle.

The two stereo cameras are synchronized and rectified. The cameras being synchronized means that they receive the trigger signal to acquire an image at the exact same time. Image rectification is a transformation process used to project two or more images onto a common image plane. In computer vision with stereo cameras this is done by finding corresponding points in the images.

The stereo camera system of Atmos aim to detect cones at distances up to 20 meters.

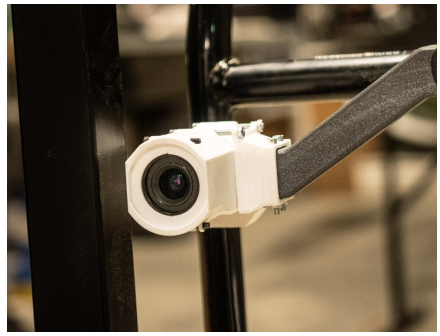


Figure A.3: The Basler acA1300-200uc USB 3.0 Camera

To achieve this, deep learning is applied using the well-known convolutional neural network, YOLOv3. This neural network perform the actual task of object detection. For improving the performance of the camera system, a number of preprocessing steps are applied to the images beforehand.

The Basler USB 3.0 cameras are renowned for being fast with a data throughput rate up to 350 MB/s. The manufacturer also ensures low CPU load, latency and jitter. In addition, these cameras are light-weight and have simple integration with most image processing libraries, making them suitable for the current application. (Basler, 2020)

A.4 Hesai Pandar 40-channel and 20-channel LiDAR

LiDAR is an active detection that uses light for measurement. It uses a pulsed laser to send out a signal and waits for the echo to arrive. The distance to any obstacles are then calculated as a function of the time between the laser emission and reception.

Atmos is equipped with two LiDARs placed as shown in Figure A.5. The rear LiDAR has 40 vertical channels, while the front LiDAR has 20 channels. The main difference between these two is thei vertical field of view. The Hesai Pandar 40-channel LiDAR has 40° vertical field of view, while the 20-channel LiDAR has 30°. Apart from this, both LiDARs have a 200 meter measurement range and a 0.33° minimum vertical resolution. Both weigh approximately 1.5 kg and provide measurement at 20 Hz. (Hesai, 2020a,b)

Autonomous vehicles commonly use LiDARs for obstacle detection and avoidance to safely navigate through their environments. Point cloud output from a LiDAR sensor provide the necessary data for perception software to determine where

Figure A.4: The Hesai Pandar 40-channel LiDAR

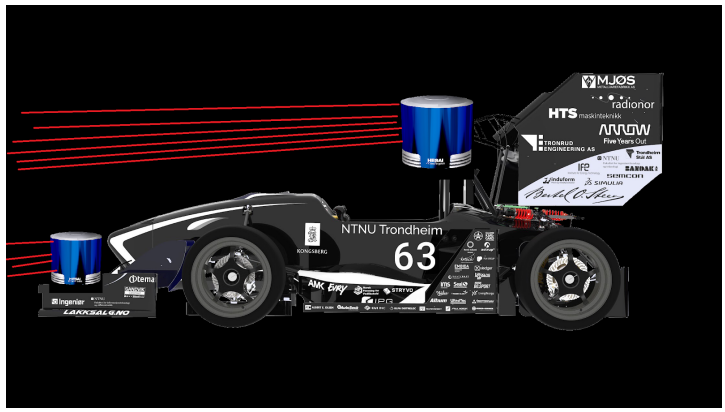


Figure A.5: Both Hesai Pandar LiDAR placed on Atmos

obstacles may be located. The LiDAR detection system on Atmos is intended to detect cones during competitions.

A.5 Optical encoders

Optical encoders are used on Atmos to measure the angle of the steering wheel, referred to as the Steering Wheel Position Sensor (SWPS) within the team, and to measure the rate of rotation of the four wheels. Optical encoders consist of a light source, a rotating grating which is a plastic or glass disk marked with a certain pattern of holes and optical light detectors. The light source sends light through the grating that is partly blocked by the pattern on the rotating disk. The light detectors detect the light that passes through the disc and generates an electrical pulse. Knowledge about the pattern on the grating enable the electrical pulse to be decoded into the rotation rate and angle of the disc.

Optical encoders are known to be high resolution, accurate and repeatable, as well as being very compact. A weakness, on the other hand, is that the accuracy is highly affected by environmental ruggedness, such as vibrations and dust. Due to these vulnerabilities, the optical encoders are omitted from the navigation solution developed here. For future iterations of this state estimation system, these may come in handy, and for instance provide redundancy.

B Quaternions

B.1 Quaternion convention

There are several ways to define the quaternion. The determination relies on four binary choices.

1. The order of its elements[1]: Real part first or last?
$$\boldsymbol{q} = \begin{bmatrix} q_\eta \\ \boldsymbol{q}_\varepsilon \end{bmatrix} \quad \text{vs.} \quad \boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_\varepsilon \\ q_\eta \end{bmatrix}$$
2. The multiplication formula: Right-handed or left-handed?
$$ij = -ji = k \quad \text{vs.} \quad ji = -ij = k$$
3. The function of the rotation operator: Passive or active?
Rotating frames vs. Rotating vectors
4. The direction of operation (in the *passive* case): Local-to-global or global-to-local?
$$\boldsymbol{x}_{global} = \boldsymbol{q} \otimes \boldsymbol{x}_{local} \otimes \boldsymbol{q}^* \quad \text{vs.} \quad \boldsymbol{x}_{local} = \boldsymbol{q} \otimes \boldsymbol{x}_{global} \otimes \boldsymbol{q}^*$$

This variety of choices yields 12 possible combinations. Historical developments has favored some certain combinations. The two most commonly used conventions for quaternions are known as the Hamilton quaternion and the JPL quaternion. Table B.1 compares the two types with respect to the four binary choices. The JPL convention is mainly used in the aerospace domain, while the Hamilton convention has been applied frequently in other engineering areas. In this state estimation implementation the Hamilton convention is chosen. This is right-handed and coincides with the Marine Systems Simulator (MSS) Toolbox developed by Thor I. Fossen, a professor at NTNU, which has been utilized during development of this navigation system. This is also the convention used in several software libraries widely used in robotic applications such as Eigen, ROS and Google Ceres (Solà, 2017).

[1] q_η represents the real, scalar part and $\boldsymbol{q}_\varepsilon = \begin{bmatrix} q_i & q_j & q_k \end{bmatrix}^T$ represents the vector part

Quaternion type	Hamilton	JPL
1. Element order	Real part first	Real part last
2. Multiplication formula	Right-handed	Left-handed
3. Function of rotation operator	Passive	Passive
4. Direction of operation	Local-to-global	Global-to-local

Table B.1: Comparison of the Hamilton and JPL quaternion conventions

B.2 Fundamental quaternion mathematics

In this section, a walk-through of some elementary mathematical operations with Hamilton quaternions is provided. The section is based on the literature of Solà (2017). For a more comprehensive coverage of the mathematical properties of quaternions, it is suggested to directly look at his work.

B.2.1 The quaternion sum

The sum of two quaternions is intuitive.

$$\boldsymbol{p} \pm \boldsymbol{q} = \begin{bmatrix} q_\eta \\ \boldsymbol{q}_\varepsilon \end{bmatrix} \pm \begin{bmatrix} q_\eta \\ \boldsymbol{q}_\varepsilon \end{bmatrix} = \begin{bmatrix} p_\eta \pm q_\eta \\ \boldsymbol{q}_\varepsilon \pm \boldsymbol{q}_\varepsilon \end{bmatrix} \tag{B.1}$$

The sum is both *commutative* and *associative*.

$$\boldsymbol{p} + \boldsymbol{q} = \boldsymbol{q} + \boldsymbol{p} \tag{B.2}$$

$$\boldsymbol{p} + (\boldsymbol{q} + \boldsymbol{r}) = (\boldsymbol{p} + \boldsymbol{q}) + \boldsymbol{r} \tag{B.3}$$

B.2.2 The quaternion product

The quaternion product is denoted $\otimes$ in this thesis, similar to Solà (2017). The product written out in vector form gives

$$\boldsymbol{p} \otimes \boldsymbol{q} = \begin{bmatrix} p_\eta q_\eta - p_i q_i - p_j q_j - p_k q_k \\ p_\eta q_i + p_i q_\eta + p_j q_k - p_k q_j \\ p_\eta q_j - p_i q_k + p_j q_\eta + p_k q_i \\ p_\eta q_k + p_i q_j - p_j q_i + p_k q_\eta \end{bmatrix} \tag{B.4}$$

A more compressed formulation of the same expression can be achieved in terms of η and ε.

$$\boldsymbol{p} \otimes \boldsymbol{q} = \begin{bmatrix} p_\eta q_\eta - \boldsymbol{p}_\varepsilon^T \boldsymbol{q}_\varepsilon \\ p_\eta \boldsymbol{q}_\varepsilon + q_\eta \boldsymbol{p}_\varepsilon + \boldsymbol{p}_\varepsilon \times \boldsymbol{q}_\varepsilon \end{bmatrix} \tag{B.5}$$

The vector cross-product reveals that the quaternion product is not commutative. Hence,

$$\boldsymbol{p} \otimes \boldsymbol{q} \neq \boldsymbol{q} \otimes \boldsymbol{p} \tag{B.6}$$

This is a general rule. However, there exist cases where the quaternion product is commutative. This is if one of the quaternions is real, such that $\boldsymbol{p} = p_\eta$ or $\boldsymbol{q} = q_\eta$, or if the vector parts are parallel, such that $\boldsymbol{p}_\varepsilon || \boldsymbol{q}_\varepsilon$. All of these yield $\boldsymbol{p}_\varepsilon \times \boldsymbol{q}_\varepsilon = 0$.

The quaternion product is associative.

$$(\boldsymbol{p} \otimes \boldsymbol{q}) \otimes \boldsymbol{r} = \boldsymbol{p} \otimes (\boldsymbol{q} \otimes \boldsymbol{r}) \tag{B.7}$$

It is also distributive over the sum.

$$\boldsymbol{p} \otimes (\boldsymbol{q} + \boldsymbol{r}) = \boldsymbol{p} \otimes \boldsymbol{q} + \boldsymbol{p} \otimes \boldsymbol{r} \quad \text{and} \quad (\boldsymbol{p} + \boldsymbol{q}) \otimes \boldsymbol{r} = \boldsymbol{p} \otimes \boldsymbol{r} + \boldsymbol{q} \otimes \boldsymbol{r} \tag{B.8}$$

B.2.3 The identity quaternion

The identity quaternion, $\boldsymbol{q}_I$ is defined such that $\boldsymbol{q}_I \otimes \boldsymbol{q} = \boldsymbol{q} \otimes \boldsymbol{q}_I = \boldsymbol{q}$ holds.

$$\boldsymbol{q}_I = 1 = \begin{bmatrix} 1 \\ \mathbf{0}_\varepsilon \end{bmatrix} \tag{B.9}$$

B.2.4 The quaternion conjugate

The conjugate of a quaternion is defined as follows.

$$\boldsymbol{q}^* = q_\eta - \boldsymbol{q}_\varepsilon = \begin{bmatrix} q_\eta \\ -\boldsymbol{q}_\varepsilon \end{bmatrix} \tag{B.10}$$

The quaternion conjugate has the following properties:

$$\boldsymbol{q} \otimes \boldsymbol{q}^* = \boldsymbol{q}^* \otimes \boldsymbol{q} = q_\eta^2 + q_i^2 + q_j^2 + q_k^2 = \begin{bmatrix} q_\eta^2 + q_i^2 + q_j^2 + q_k^2 \\ \mathbf{0}_\varepsilon \end{bmatrix} \tag{B.11}$$

$$(\boldsymbol{p} \otimes \boldsymbol{q})^* = \boldsymbol{q}^* \otimes \boldsymbol{p}^* \tag{B.12}$$

B.2.5 The norm of a quaternion

The norm of a quaternion is defined as

$$||\boldsymbol{q}|| = \sqrt{\boldsymbol{q} \otimes \boldsymbol{q}^*} = \sqrt{\boldsymbol{q}^* \otimes \boldsymbol{q}} = \sqrt{q_\eta^2 + q_i^2 + q_j^2 + q_k^2} \tag{B.13}$$

It has the property that $||\boldsymbol{p} \otimes \boldsymbol{q}|| = ||\boldsymbol{q} \otimes \boldsymbol{p}|| = ||\boldsymbol{p}|| \cdot ||\boldsymbol{q}||$

The unit quaternion is a quaternion where $||\boldsymbol{q}|| = 1$.

B.2.6 The quaternion inverse

The inverse of a quaternion, $\boldsymbol{q}^{-1}$ is defined such that the quaternion times its inverse results in the identity quaternion.

$$\boldsymbol{q} \otimes \boldsymbol{q}^{-1} = \boldsymbol{q}^{-1} \otimes \boldsymbol{q} = \boldsymbol{q}_I \tag{B.14}$$

The is computed as follows.

$$\boldsymbol{q}^{-1} = \frac{\boldsymbol{q}^*}{||\boldsymbol{q}||^2} \tag{B.15}$$

B.2.7 Rotation with quaternions

The quaternion rotation action of a vector $\boldsymbol{v}$ is carried out through the following formula.

$$r(\boldsymbol{v}) = \boldsymbol{q} \otimes \boldsymbol{v} \otimes \boldsymbol{q}^* \tag{B.16}$$

For a detailed derivation of this formula, the literature of Solà (2017) can be studied.

C

Jacobian of the Ground Speed Measurement Function: $\boldsymbol{H}_{gss}$

This appendix provides the complete expression for the Jacobian matrix of the ground speed measurement function, linearized about about $\delta \boldsymbol{x} = 0$

The measurement function of the ground speed aiding measurement is:

$$\boldsymbol{y}_{gss} = \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{R}_b^n \boldsymbol{v}_{gss}^n \right\|_2 + \boldsymbol{w}_{gss} \tag{C.1}$$

where

$$\boldsymbol{v}_{gss}^n = \boldsymbol{v}_{nb}^n + \boldsymbol{R}_b^n S(\boldsymbol{\omega}_{nb}^b) \boldsymbol{r}_{gss}^b \tag{C.2}$$

accounting for the displacement of the GSS.

The same measurement function reformulated as a function of the error state vector $\delta \boldsymbol{x}$ is achieved using the relation $\boldsymbol{x} = \hat{\boldsymbol{x}}_{ins} \otimes \delta \boldsymbol{x}$. This results is:

$$\boldsymbol{y}_{gss} = \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \hat{\boldsymbol{R}}_{n,ins}^{b\ T} \hat{\boldsymbol{v}}_{ins}^n + \hat{\boldsymbol{R}}_{n,ins}^{b\ T} \delta \boldsymbol{v}_{nb}^n + \boldsymbol{S}(\hat{\boldsymbol{R}}_{n,ins}^{b\ T} \hat{\boldsymbol{v}}_{ins}^n + \hat{\boldsymbol{R}}_{n,ins}^{b\ T} \delta \boldsymbol{v}_{nb}^n) \delta \boldsymbol{\theta}_{nb}^b \right\|_2 + \boldsymbol{w}_{gss} \tag{C.3}$$

From this, the linearization is carried out by computing the Jacobian matrix of $\boldsymbol{y}_{gss}$ with respect to the error state, $\delta \boldsymbol{x}$, about $\delta \boldsymbol{x} = 0$.

$$\boldsymbol{H}_{gss} = \left. \frac{\partial \boldsymbol{h}_{gss}}{\partial \delta \boldsymbol{x}} \right|_{\delta \boldsymbol{x}=0} \tag{C.4}$$

The expression above is solved using MATLAB's Symbolic Math toolbox. The resulting partial derivatives has non-zero expressions with respect to the velocity error, the attitude error and the gyroscope bias error. All partial derivatives are provided below:

$$\boldsymbol{H}_{gss_{pos}} = \left. \frac{\partial \boldsymbol{h}_{gss}}{\partial \delta \boldsymbol{p}} \right|_{\delta \boldsymbol{p}=0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{C.5}$$

$$\boldsymbol{H}_{gss_{acc}} = \left. \frac{\partial \boldsymbol{h}_{gss}}{\partial \delta \boldsymbol{b}_{acc}} \right|_{\delta \boldsymbol{b}_{acc}=0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{C.6}$$

$$\boldsymbol{H}_{gss_{vel}} = \left.\frac{\partial \boldsymbol{h}_{gss}}{\partial \delta v}\right|_{\delta v=0} = \begin{bmatrix} \frac{R_{12}\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{Y_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)+R_{11}\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)}{\sqrt{\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)^2+\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)^2}} \\ \frac{R_{22}\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)+R_{21}\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)}{\sqrt{\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)^2+\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)^2}} \\ \frac{R_{32}\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)+R_{31}\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)}{\sqrt{\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)^2+\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)^2}} \end{bmatrix} \tag{C.7}$$

$$\boldsymbol{H}_{gss_{att}} = \left.\frac{\partial \boldsymbol{h}_{gss}}{\partial \theta}\right|_{\delta\theta=0} = \begin{bmatrix} \frac{\left(R_{13}\,v_{X_n}+R_{23}\,v_{Y_n}+R_{33}\,v_{Z_n}\right)\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)}{\sqrt{\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)^2+\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)^2}} \\ -\frac{\left(R_{13}\,v_{X_n}+R_{23}\,v_{Y_n}+R_{33}\,v_{Z_n}\right)\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)}{\sqrt{\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)^2+\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)^2}} \\ -\frac{2\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}\right)\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)-2\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}\right)\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)}{2\sqrt{\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)^2+\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)^2}} \end{bmatrix} \tag{C.8}$$

$$\boldsymbol{H}_{gss_{ars}} = \left.\frac{\partial \boldsymbol{h}_{gss}}{\partial \delta b_{ars}}\right|_{\delta b_{ars}=0} = \begin{bmatrix} \frac{r_{Z_b}\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)}{\sqrt{\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)^2+\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)^2}} \\ -\frac{r_{Z_b}\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)}{\sqrt{\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)^2+\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)^2}} \\ \frac{r_{X_b}\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)-r_{Y_b}\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)}{\sqrt{\left(R_{12}\,v_{X_n}+R_{22}\,v_{Y_n}+R_{32}\,v_{Z_n}-b_{arsz,b}\,r_{X_b}+b_{arsx,b}\,r_{Z_b}-o_{X_{imu,b}}\,r_{Z_b}+o_{Z_{imu,b}}\,r_{X_b}\right)^2+\left(R_{11}\,v_{X_n}+R_{21}\,v_{Y_n}+R_{31}\,v_{Z_n}+b_{arsz,b}\,r_{Y_b}-b_{arsy,b}\,r_{Z_b}+o_{Y_{imu,b}}\,r_{Z_b}-o_{Z_{imu,b}}\,r_{Y_b}\right)^2}} \end{bmatrix} \tag{C.9}$$

D

The Variance of a First Order Gauss-Markov Process

A first order Gauss-Markov process is given by:

$$\dot{x} = -\frac{1}{T}x + \gamma \tag{D.1}$$

where γ is a white noise process with power spectral density (PSD) Q, such that $E(\gamma(t)\gamma(\tau) = Q * \delta(t - \tau)$.

Equation D.1 yield the following solution.

$$x(t) = \exp -\frac{1}{T}(t - t_0 * x(t_0) - \int_{t_0}^{t} \exp -\frac{1}{T}(t - \tau\gamma(\tau)d\tau \tag{D.2}$$

Letting $t_0 = 0$ and $x(t_0) = 0$, the expected value and variance of x follows Equation D.3 below.

$$E(x(t)) = 0 \tag{D.3a}$$

$$\sigma_x^2 = E(x^2(t)) = E\left(\int_0^t \int_0^t \exp -\frac{1}{T}(t - \tau_1) \exp -\frac{1}{T}(t - \tau_2)\gamma(\tau_1)\gamma(\tau_2)d\tau_1 d\tau_2\right) \tag{D.3b}$$

Utilizing that the expected value operator $E()$ and the integral can swap places, the following reformulation is valid:

$$\sigma_x^2 = E(x^2(t)) = \int_0^t \int_0^t \exp -\frac{1}{T}(t - \tau_1) \exp -\frac{1}{T}(t - \tau_2)E\left(\gamma(\tau_1)\gamma(\tau_2)\right) d\tau_1 d\tau_2 \tag{D.4}$$

As mentioned the PSD of γ is equal to Q. Hence, we obtain:

$$\begin{aligned} \sigma_x^2 &= \int_0^t \int_0^t \exp -\frac{1}{T}(t - \tau_1) \exp -\frac{1}{T}(t - \tau_2)Q\delta(\tau_1 - \tau_2)d\tau_1 d\tau_2 \\ &= Q \int_0^t \exp -\frac{1}{T}(t - \tau_1) \int_0^t \exp -\frac{1}{T}(t - \tau_2)\delta(\tau_1 - \tau_2)d\tau_1 d\tau_2 \end{aligned} \tag{D.5}$$

Since $\delta(\tau_1 - \tau_2) = 0$ when $\tau_1 \neq \tau_2$ (and the integrals have the same limits), the following is obtained:

$$\sigma_x^2 = Q \int_0^t \exp -\frac{1}{T}(t - \tau_2) \exp -\frac{1}{T}(t - \tau_2) d\tau_2 = Q \int_0^t \exp -\frac{2}{T}(t - \tau_2) d\tau_2 \quad \text{(D.6)}$$

Furthermore, this yields:

$$\sigma_x^2 = Q \left[\tfrac{T}{2} \exp -\tfrac{2}{T}(t - \tau_2) \right]_0^t = Q(\frac{2}{T}(1 - \exp -\frac{2}{T}t) \quad \text{(D.7)}$$

Letting $t \to \infty$, the resulting stationary variance becomes:

$$\sigma_x^2 = Q\frac{T}{2} \quad \text{(D.8)}$$

Results when the Accelerometer Bias is Omitted from the Specific Force Aiding

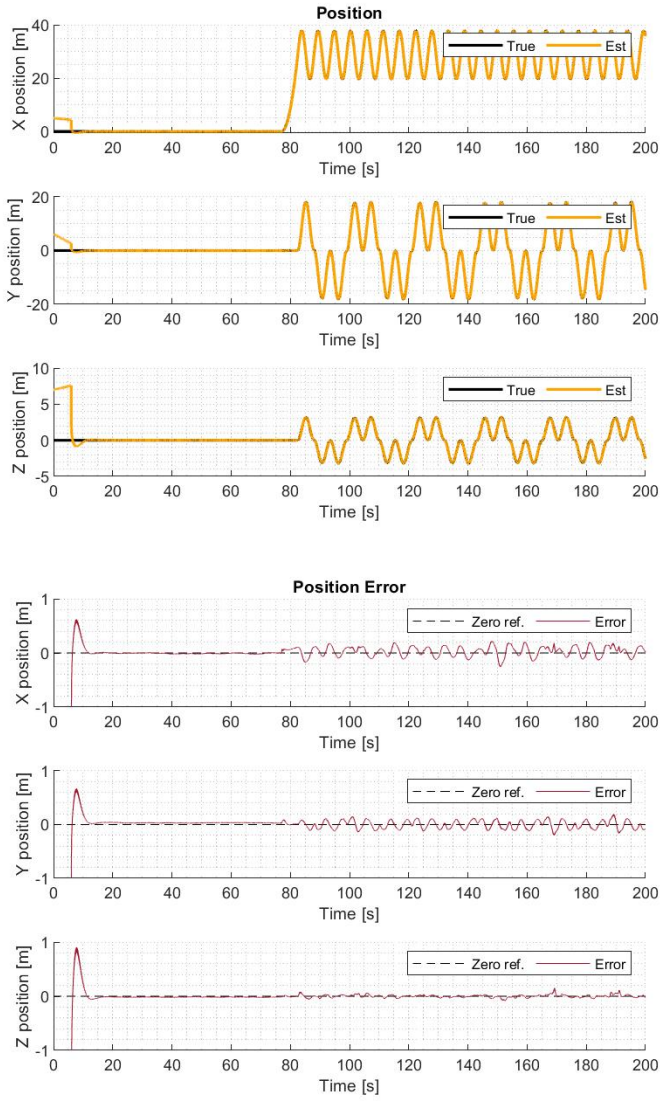


Figure E.1: Results of position estimation when the accelerometer bias is omitted from the specific force aiding

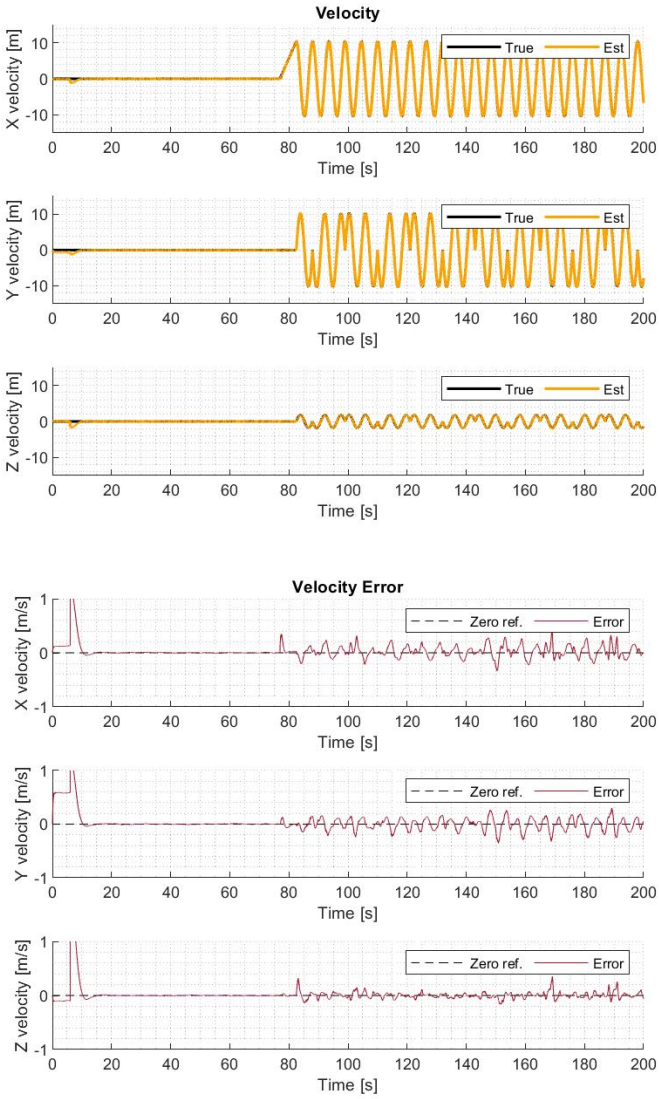


Figure E.2: Results of velocity estimation when the accelerometer bias is omitted from the specific force aiding

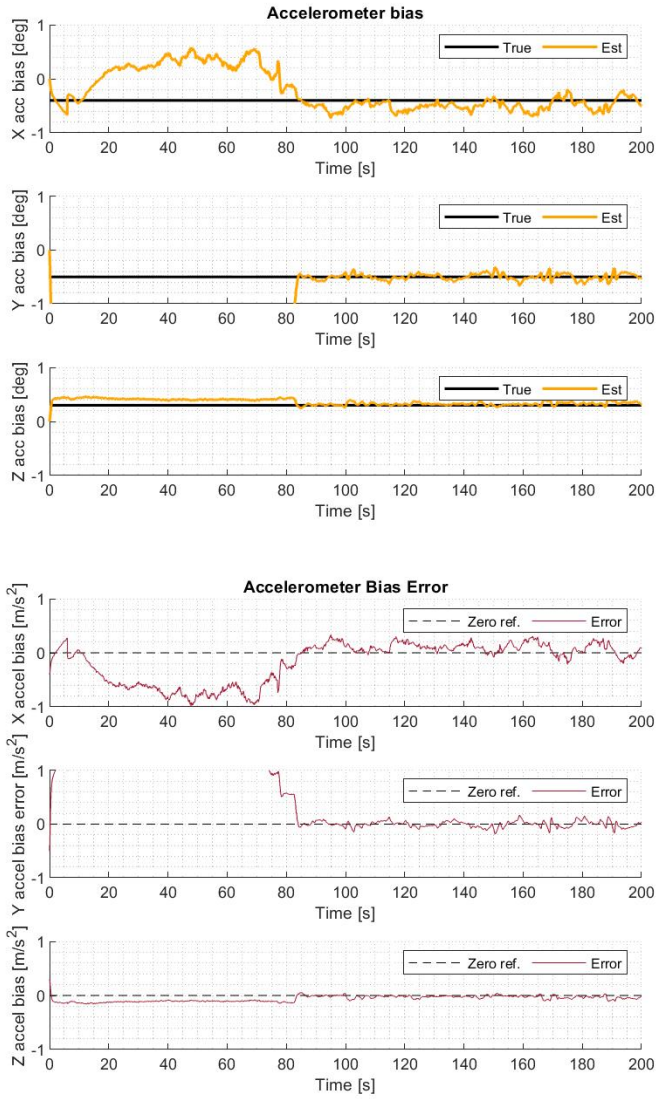


Figure E.3: Results of accelerometer bias estimation when the accelerometer bias is omitted from the specific force aiding

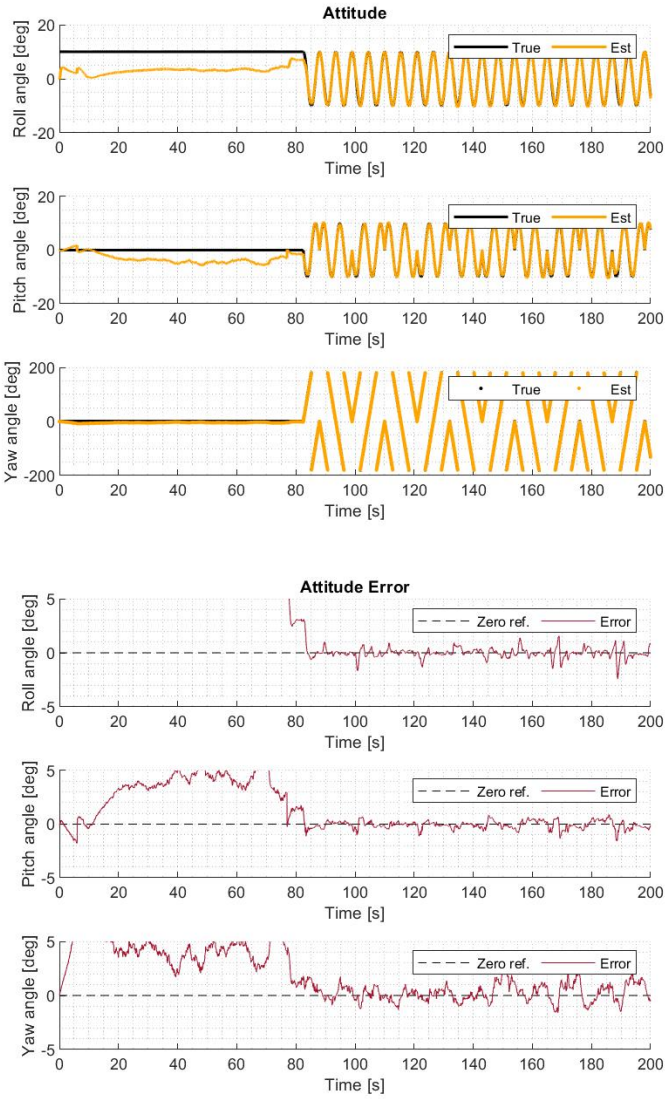


Figure E.4: Results of attitude estimation when the accelerometer bias is omitted from the specific force aiding

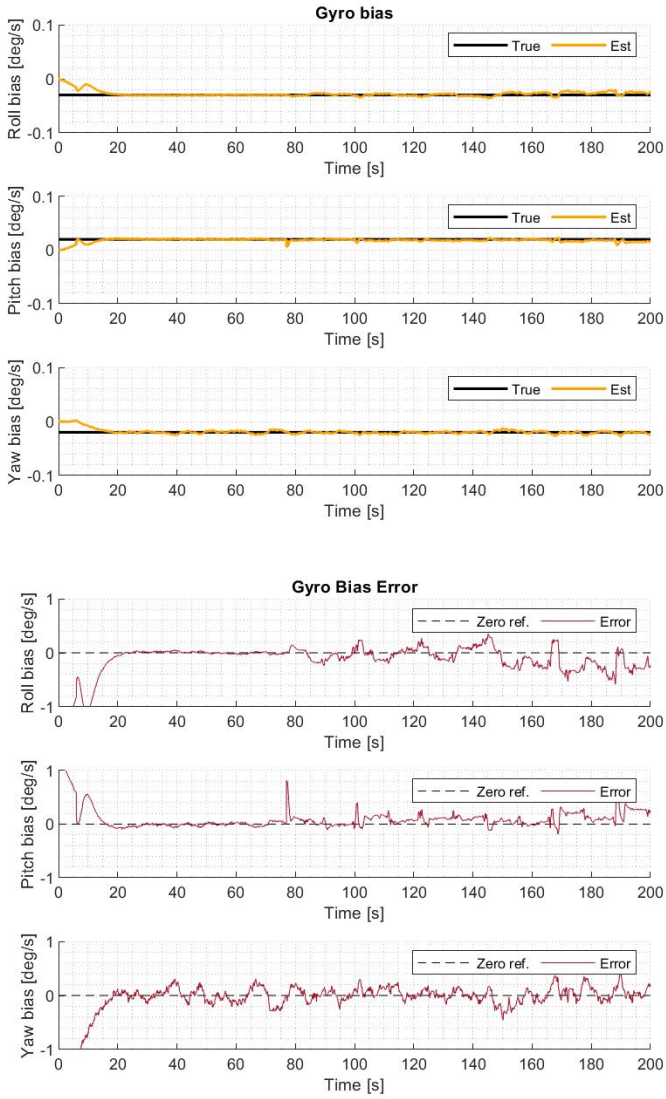


Figure E.5: Results of gyroscope bias estimation when the accelerometer bias is omitted from the specific force aiding

Results when no Pre-Race Modifications are Applied

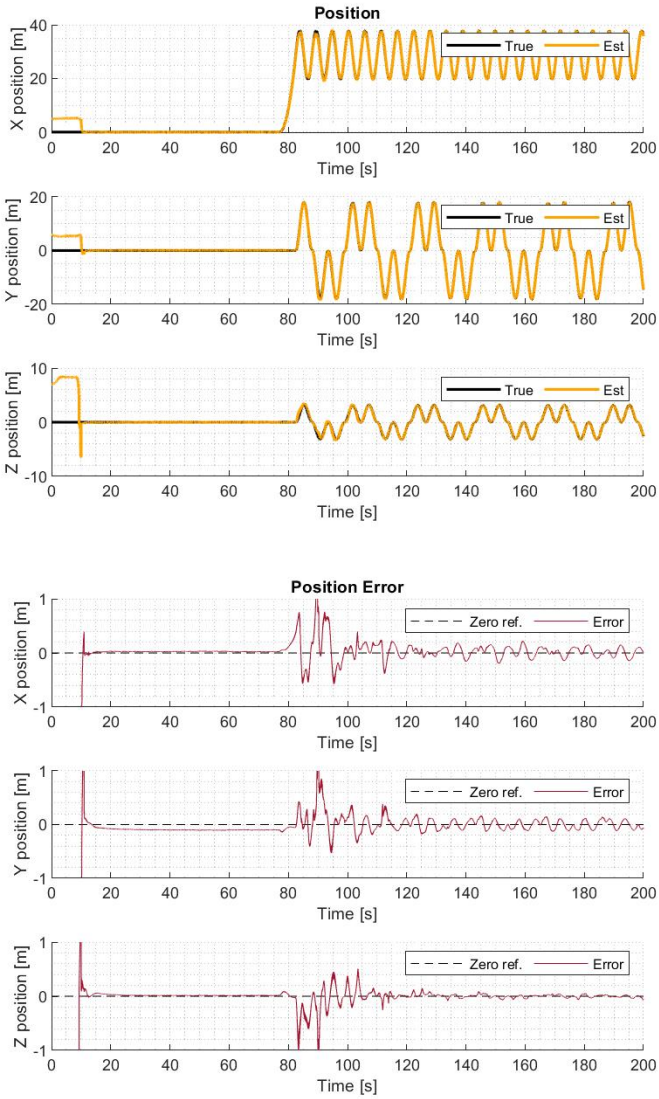


Figure F.1: Results of position estimation when the pre-race modifications are removed

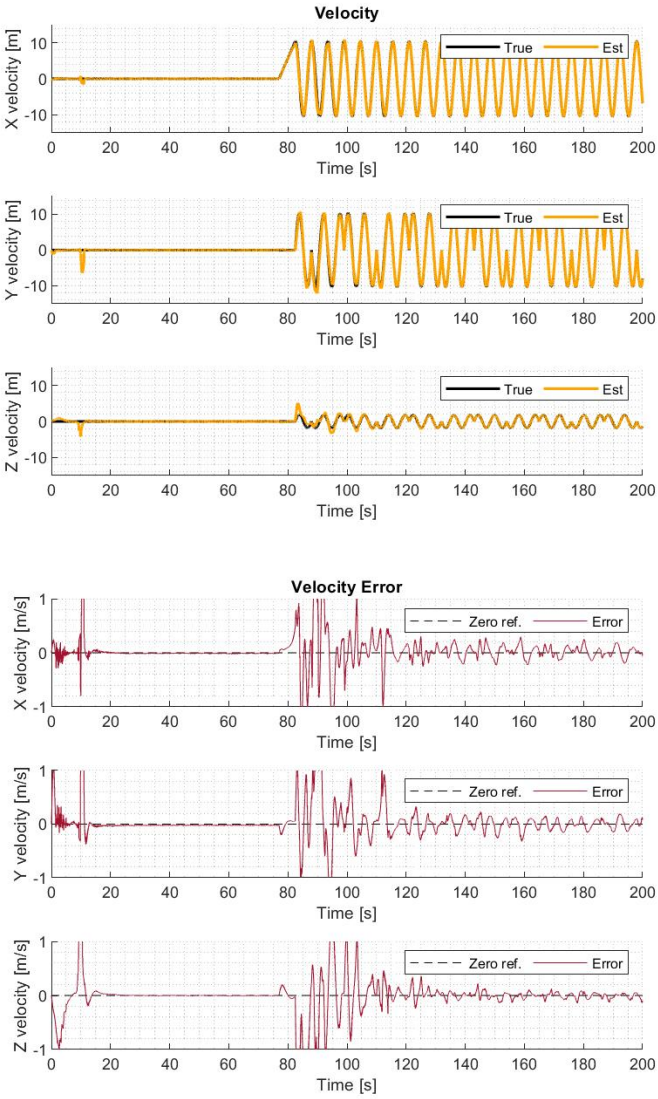


Figure F.2: Results of velocity estimation when the pre-race modifications are removed

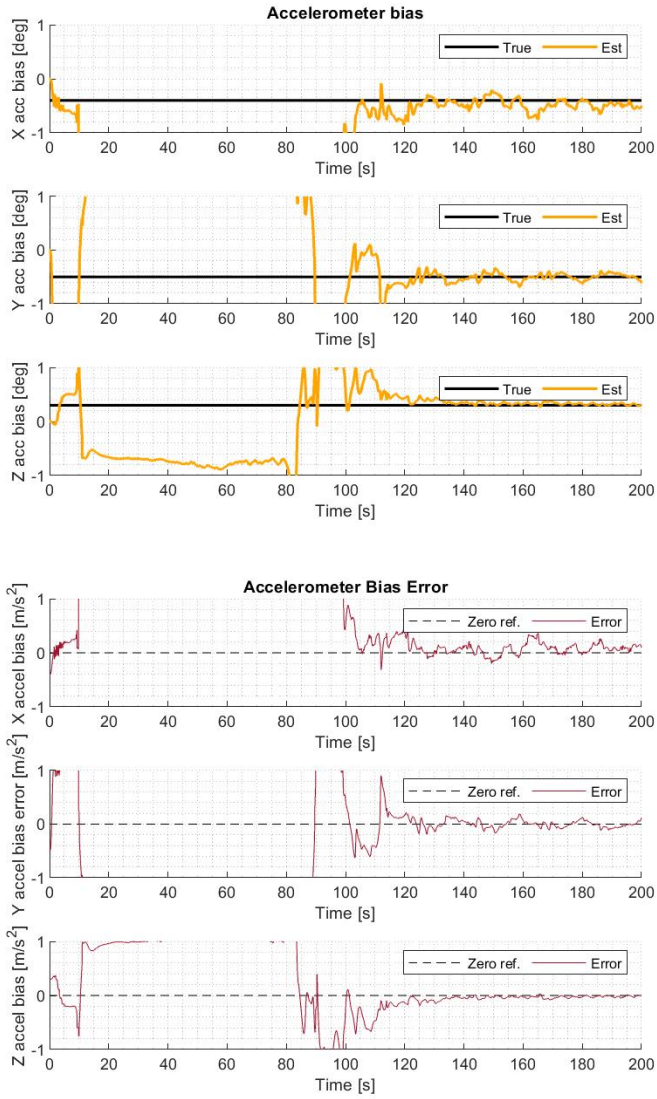


Figure F.3: Results of accelerometer bias estimation when the pre-race modifications are removed

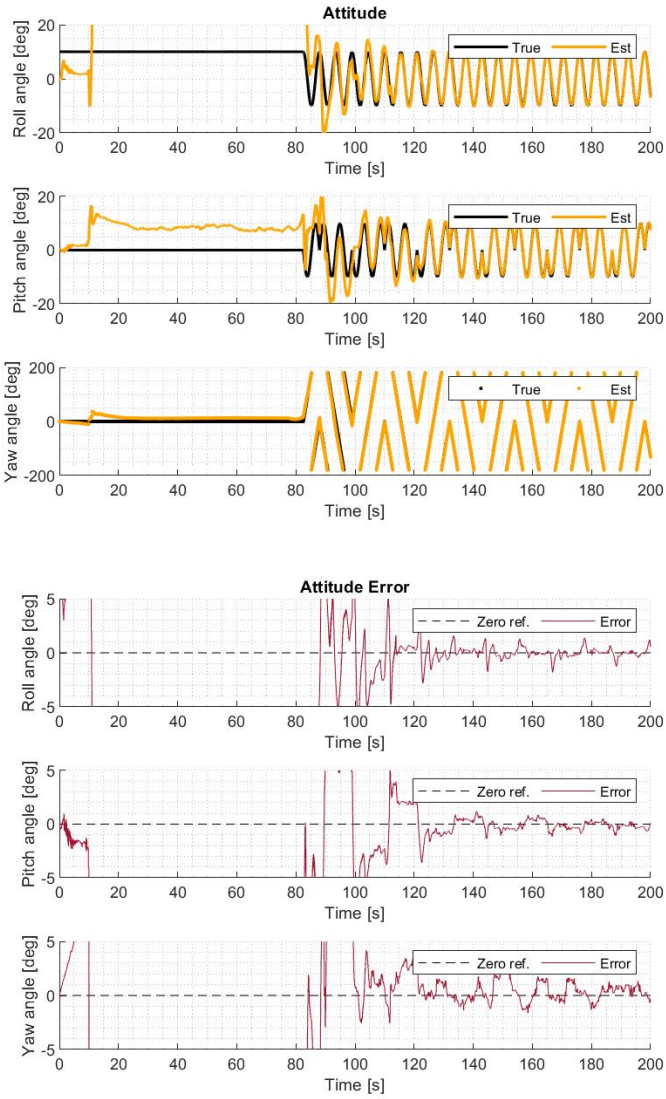


Figure F.4: Results of attitude estimation when the pre-race modifications are removed

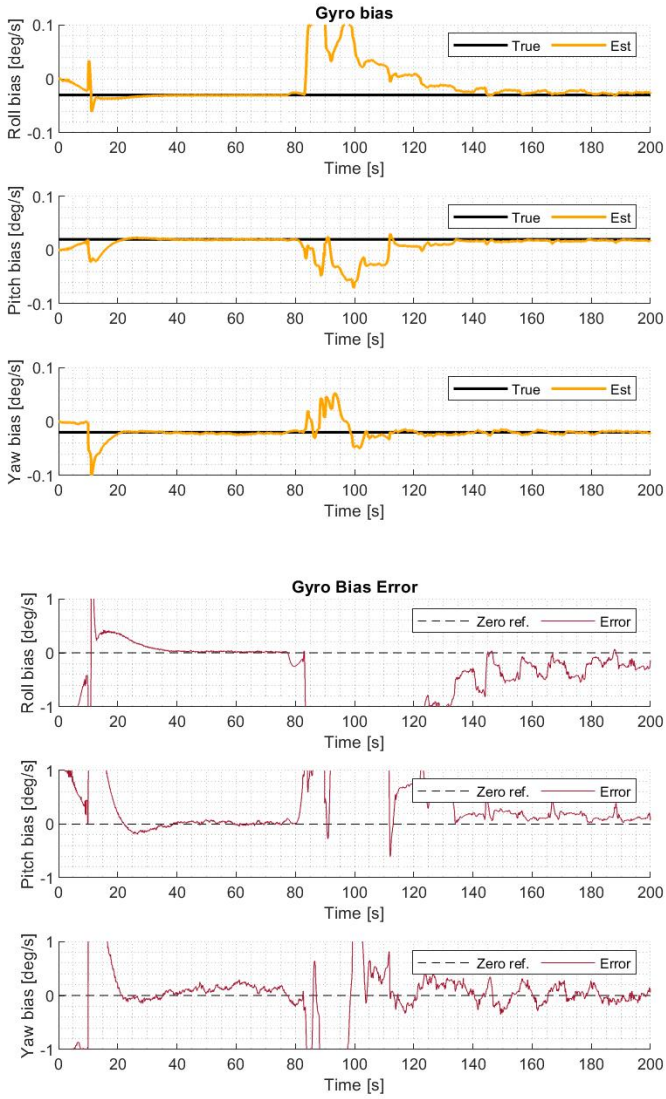


Figure F.5: Results of gyroscope bias estimation when the pre-race modifications are removed

Results without Gravity Estimation

G

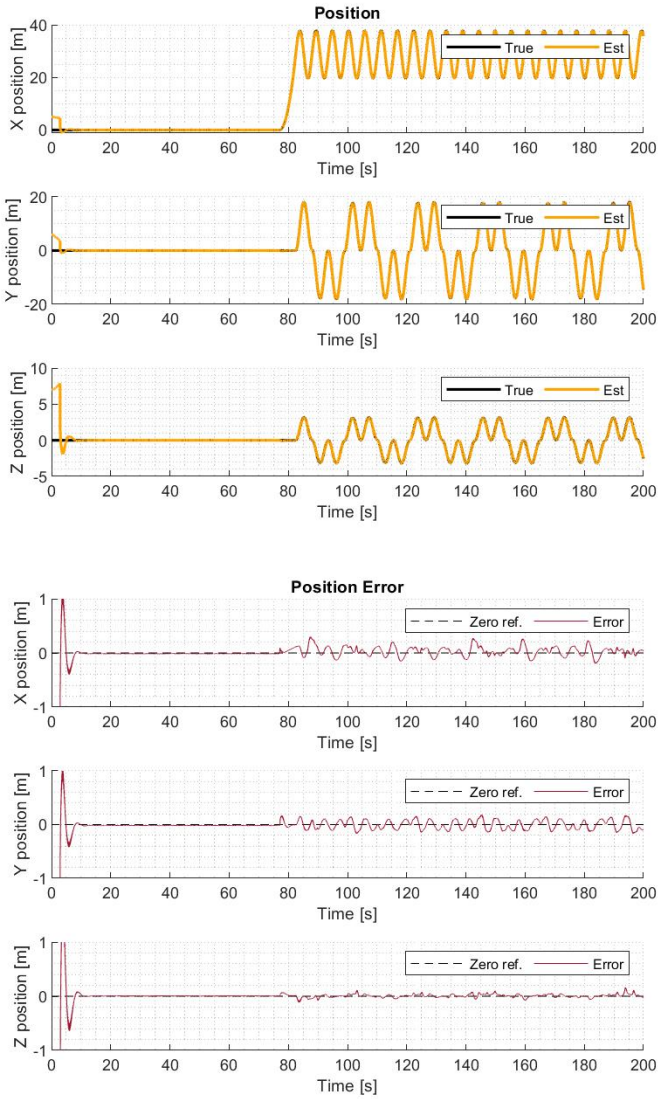


Figure G.1: Results of position estimation when the gravity estimation is omitted

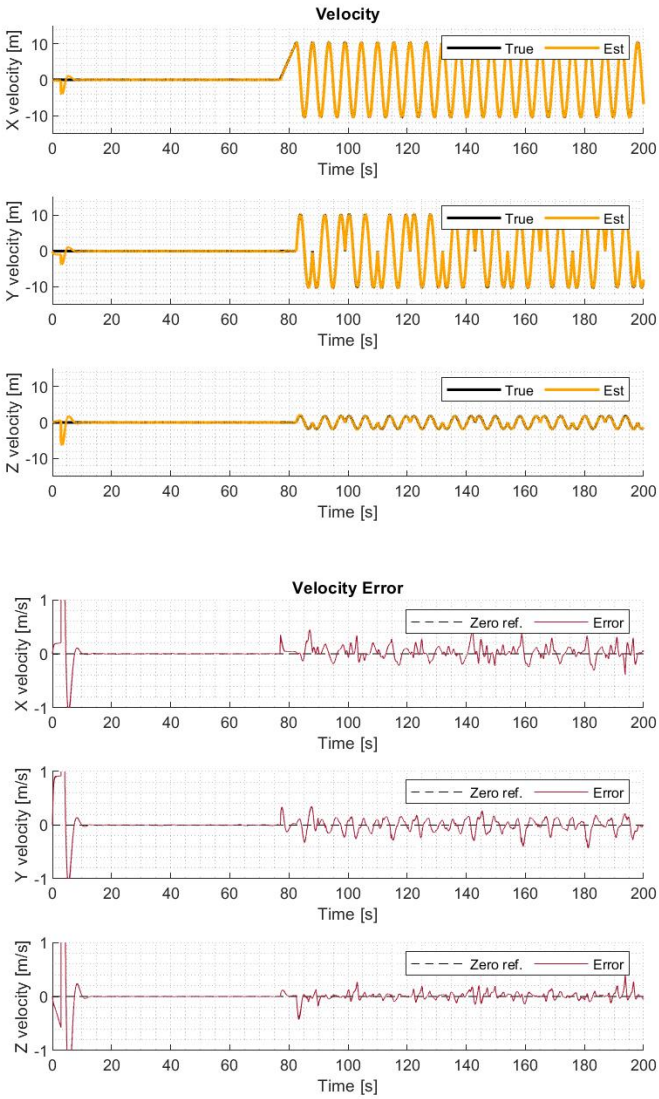


Figure G.2: Results of velocity estimation when the gravity estimation is omitted

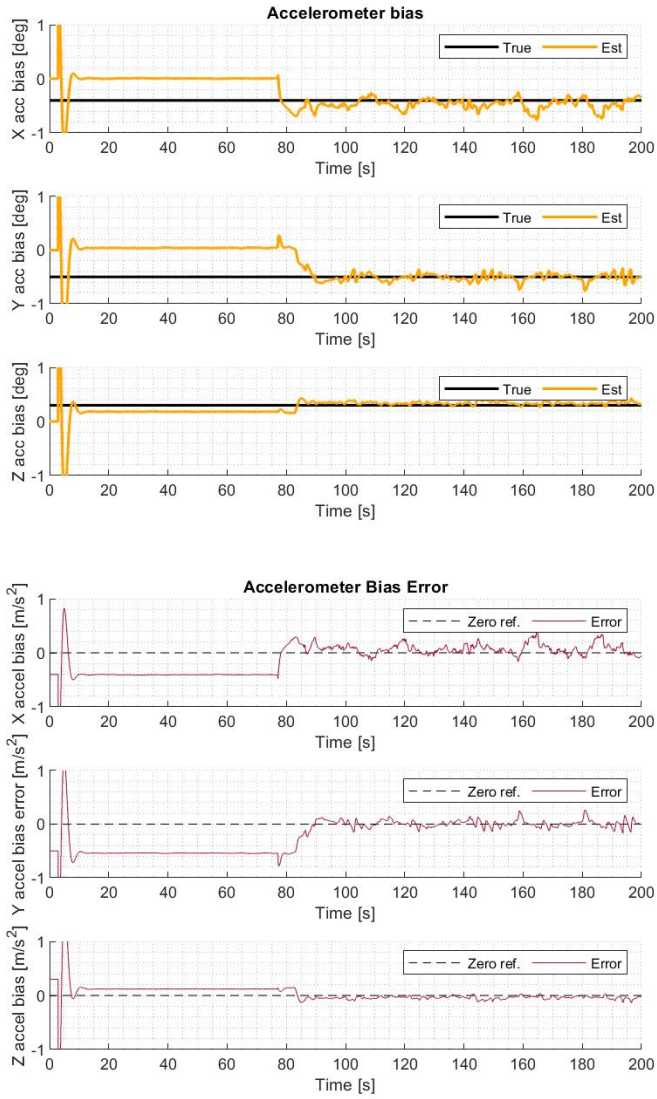


Figure G.3: Results of accelerometer bias estimation when the gravity estimation is omitted

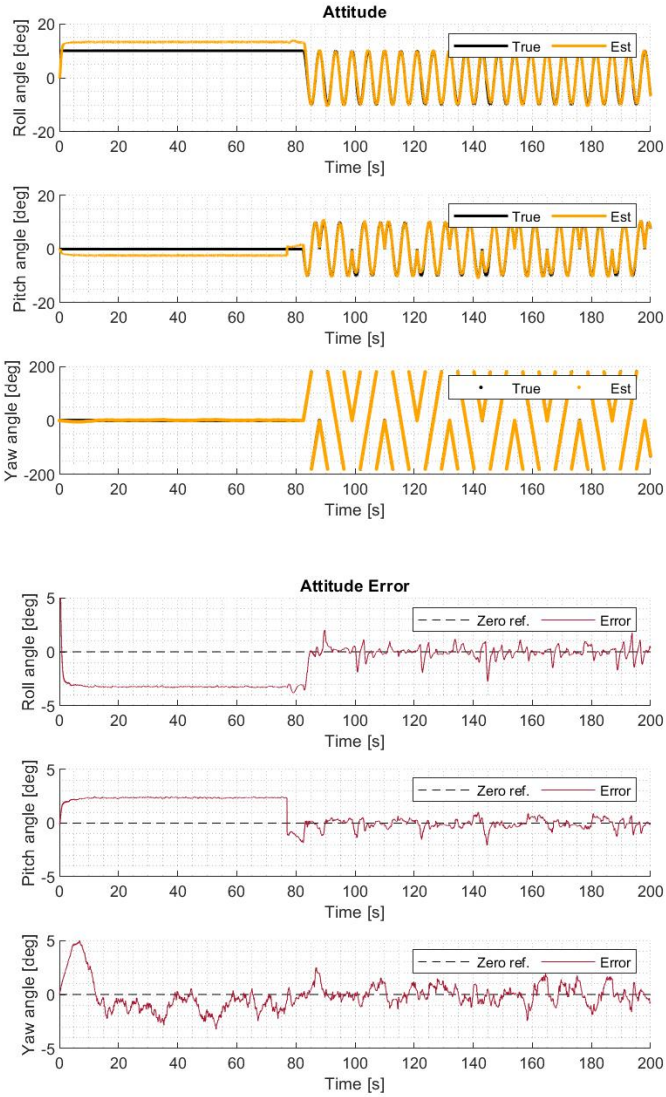


Figure G.4: Results of attitude estimation when the gravity estimation is omitted

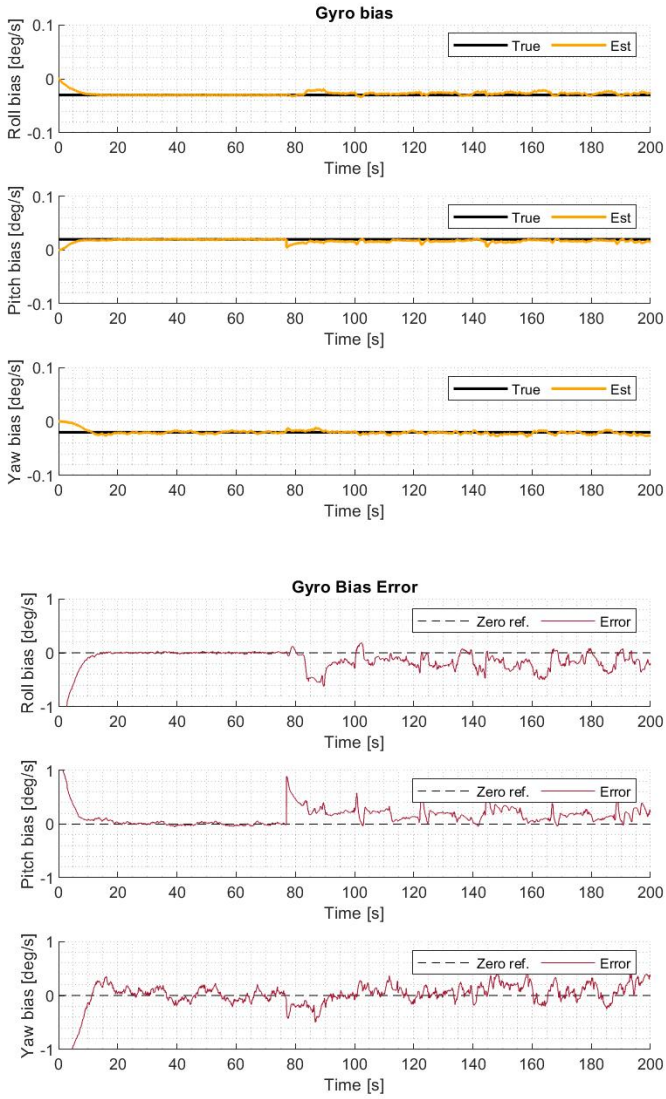


Figure G.5: Results of gyroscope bias estimation when the gravity estimation is omitted

Results without Ground Speed Aiding

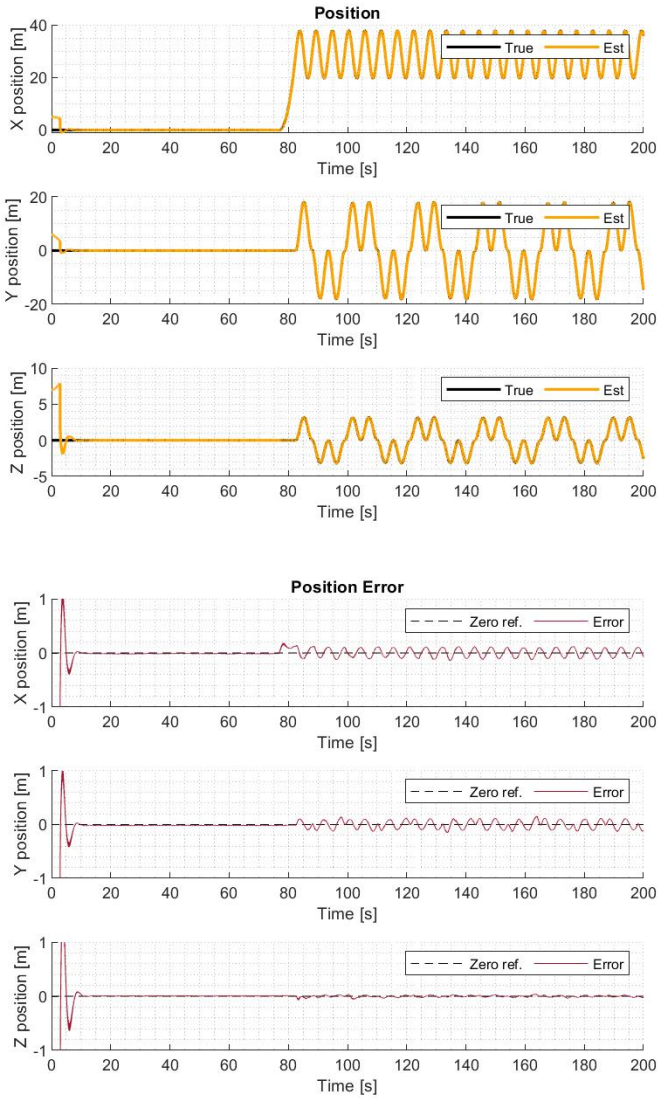


Figure H.1: Results of position estimation when the gravity estimation is omitted

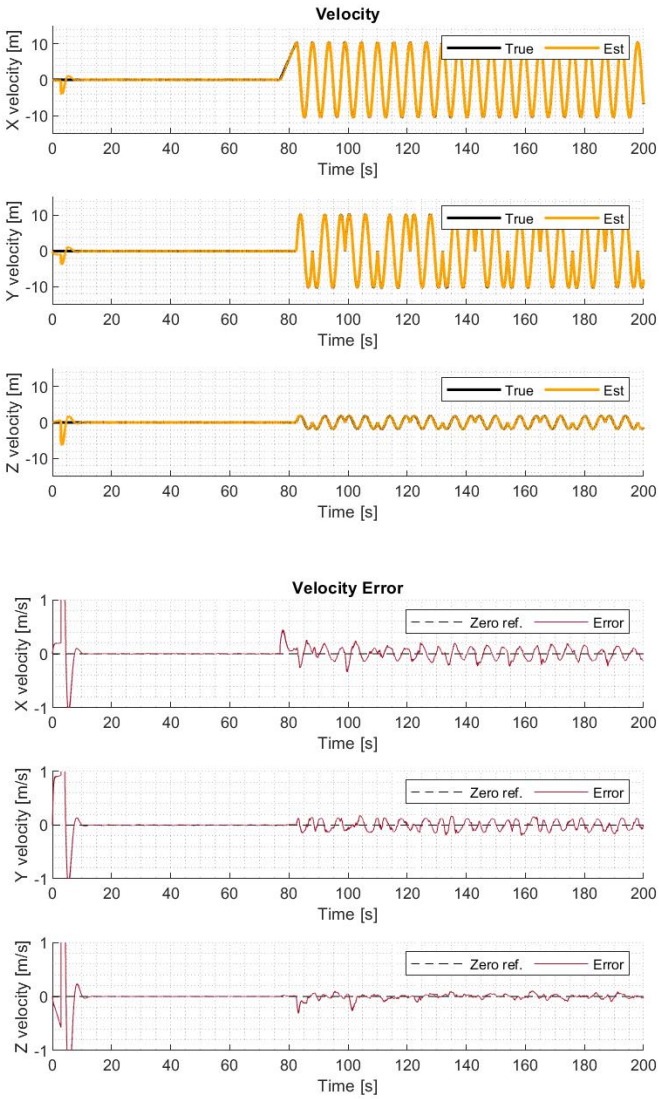


Figure H.2: Results of velocity estimation when the gravity estimation is omitted

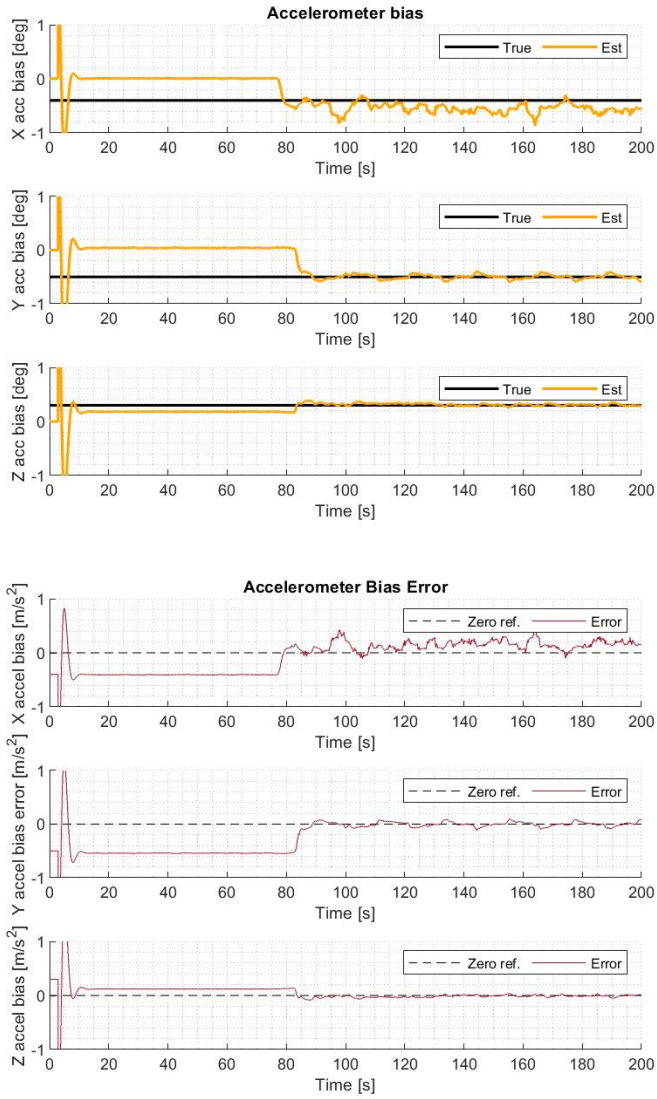


Figure H.3: Results of accelerometer bias estimation when the gravity estimation is omitted

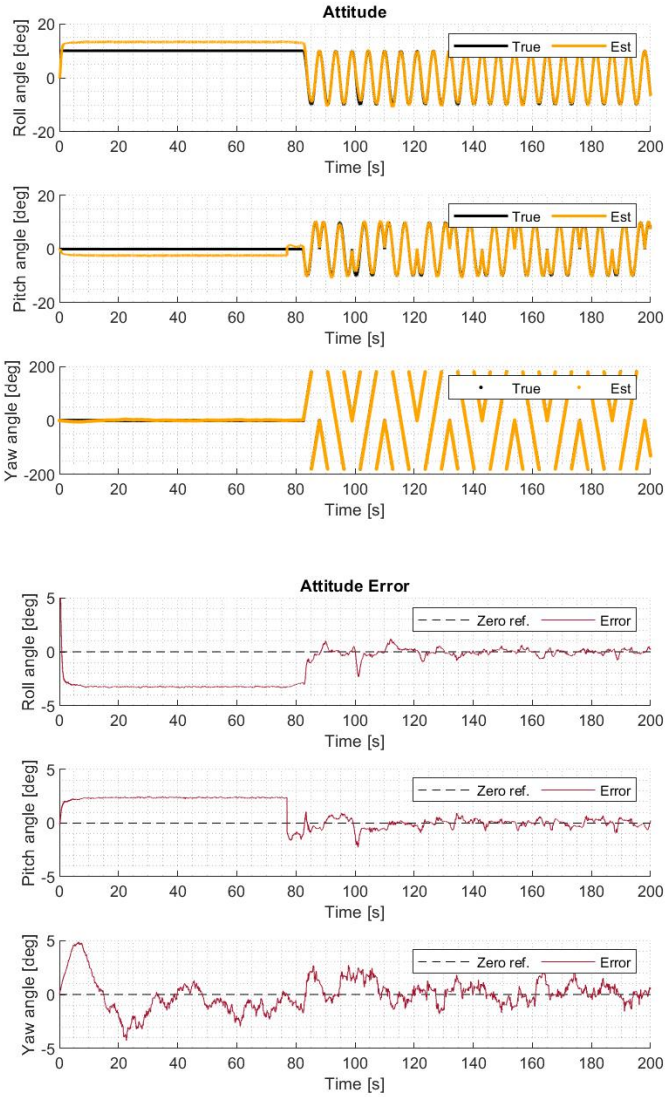


Figure H.4: Results of attitude estimation when the gravity estimation is omitted

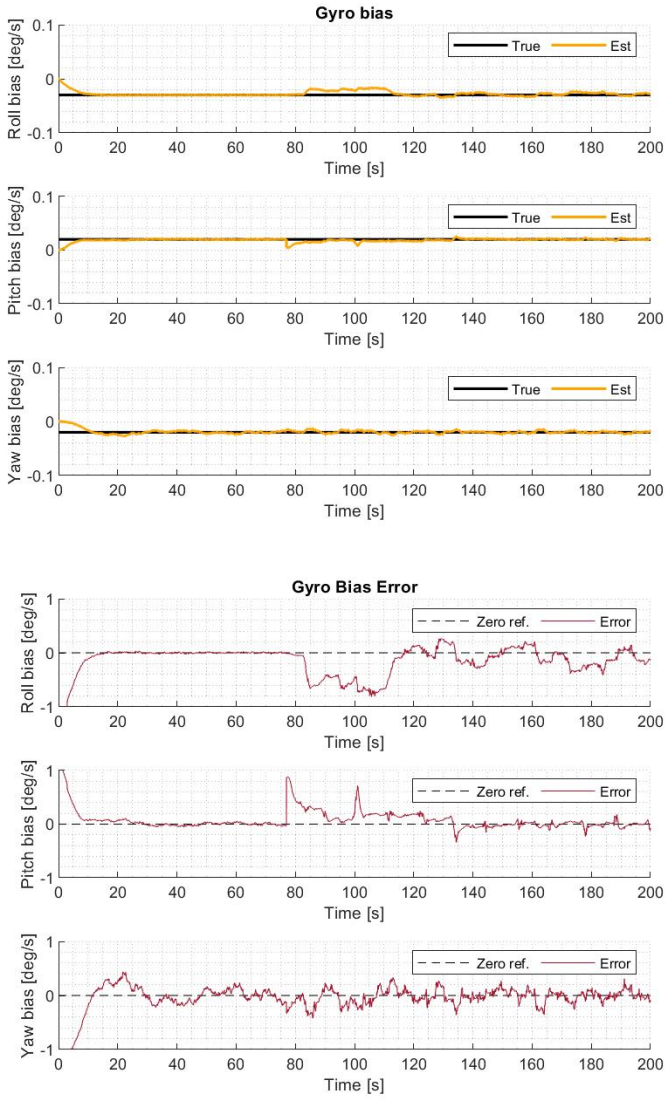


Figure H.5: Results of gyroscope bias estimation when the gravity estimation is omitted

I Results without Vector-Measurement Aiding

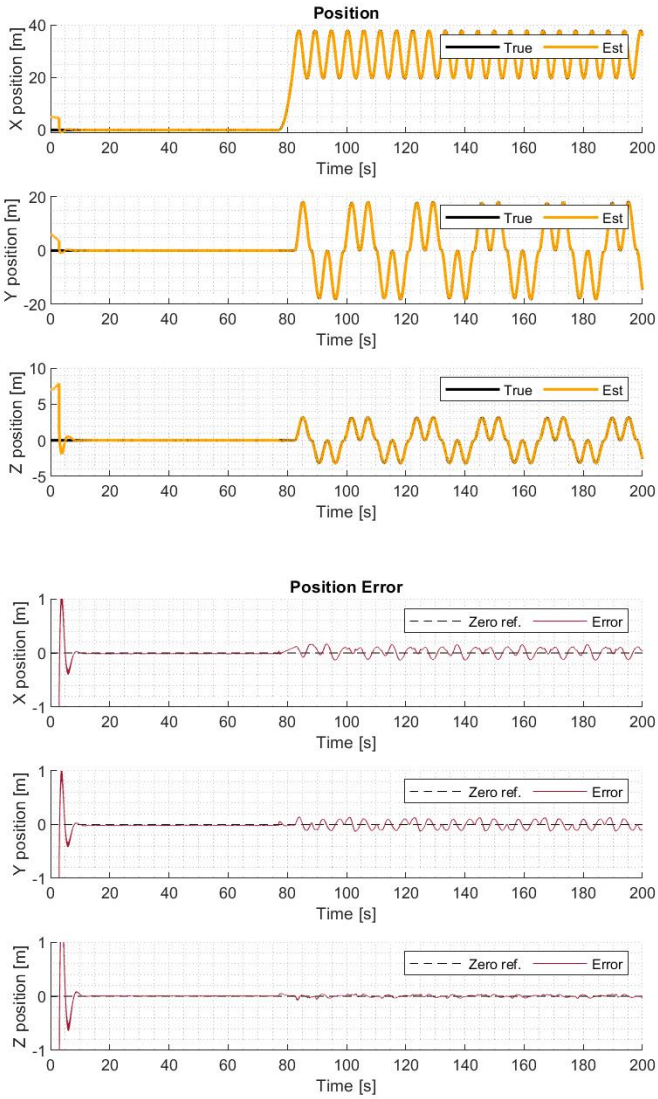


Figure I.1: Results of position estimation when the gravity estimation is omitted

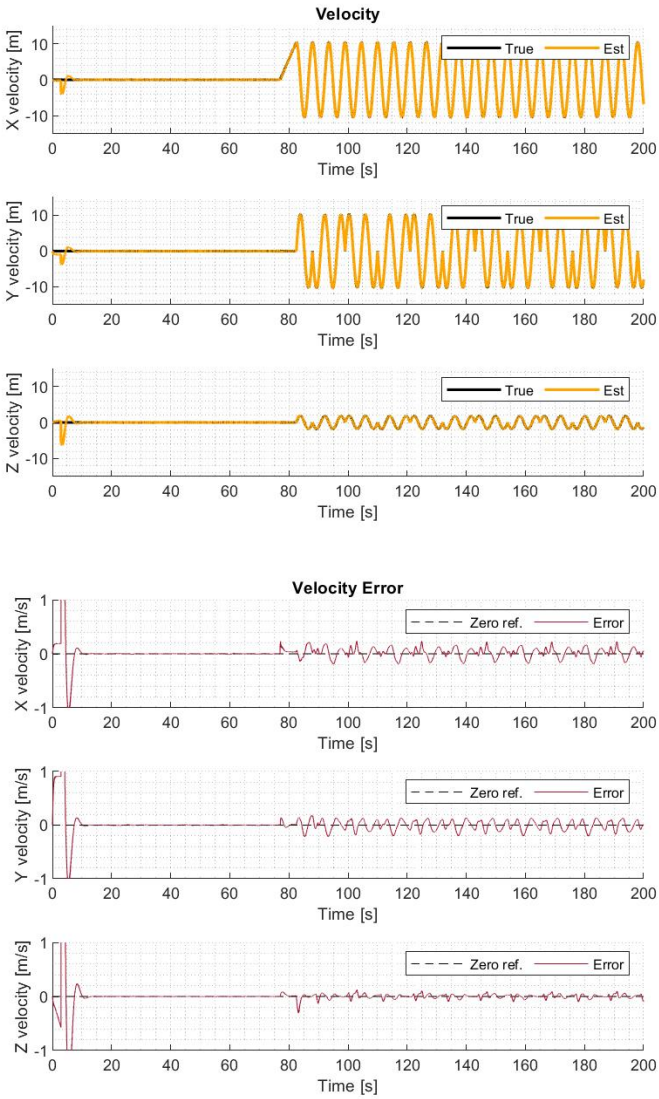


Figure I.2: Results of velocity estimation when the gravity estimation is omitted

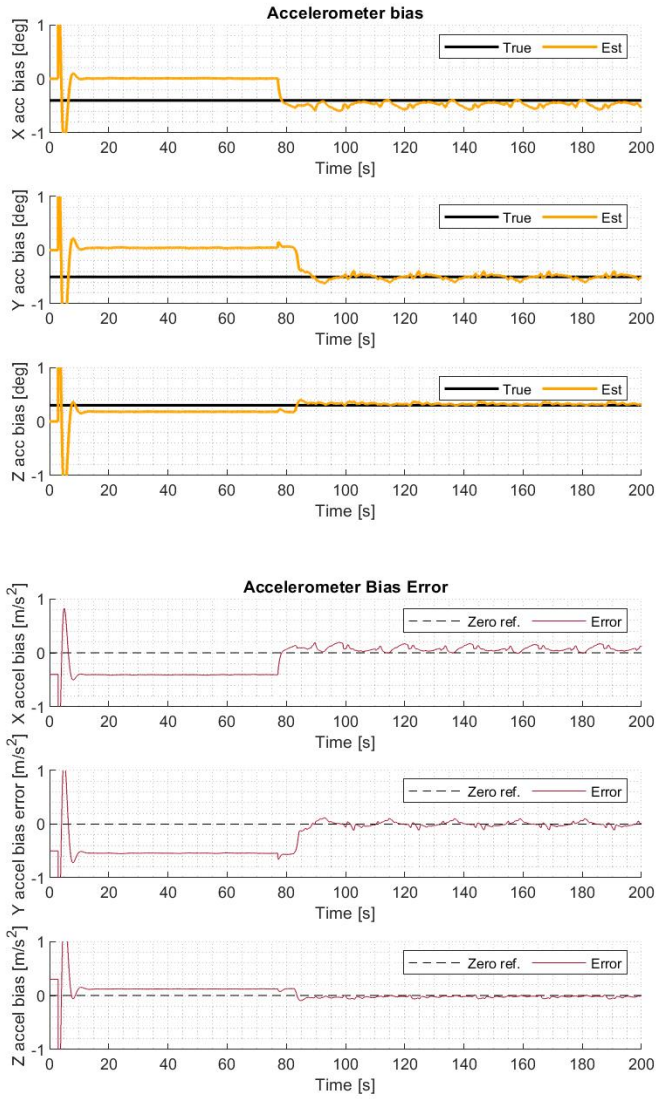


Figure I.3: Results of accelerometer bias estimation when the gravity estimation is omitted

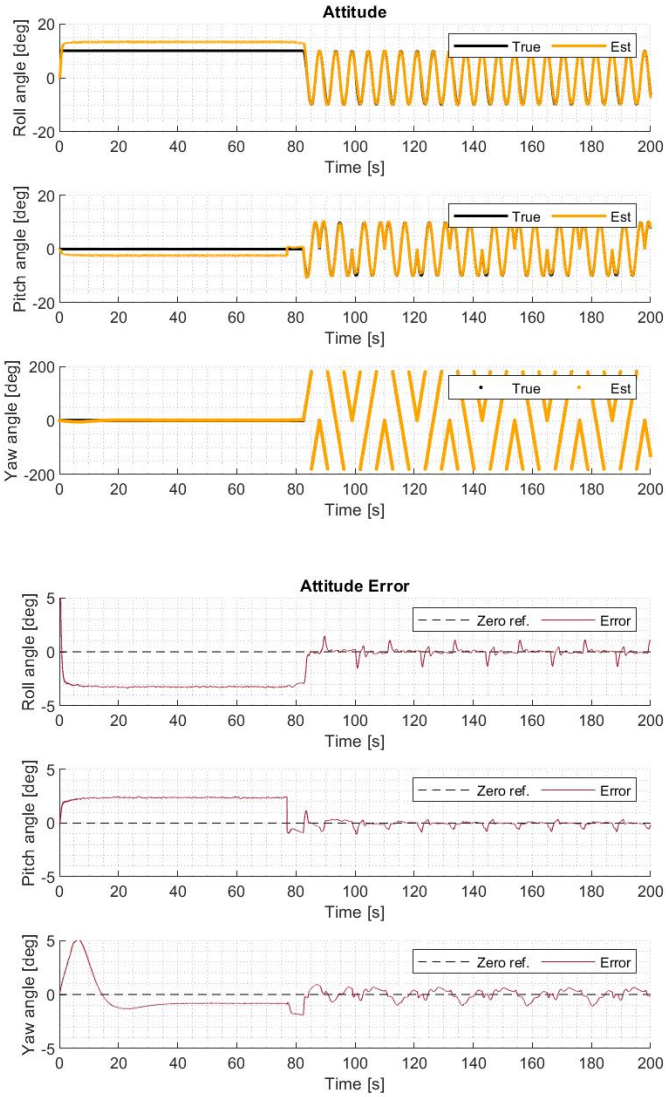


Figure I.4: Results of attitude estimation when the gravity estimation is omitted

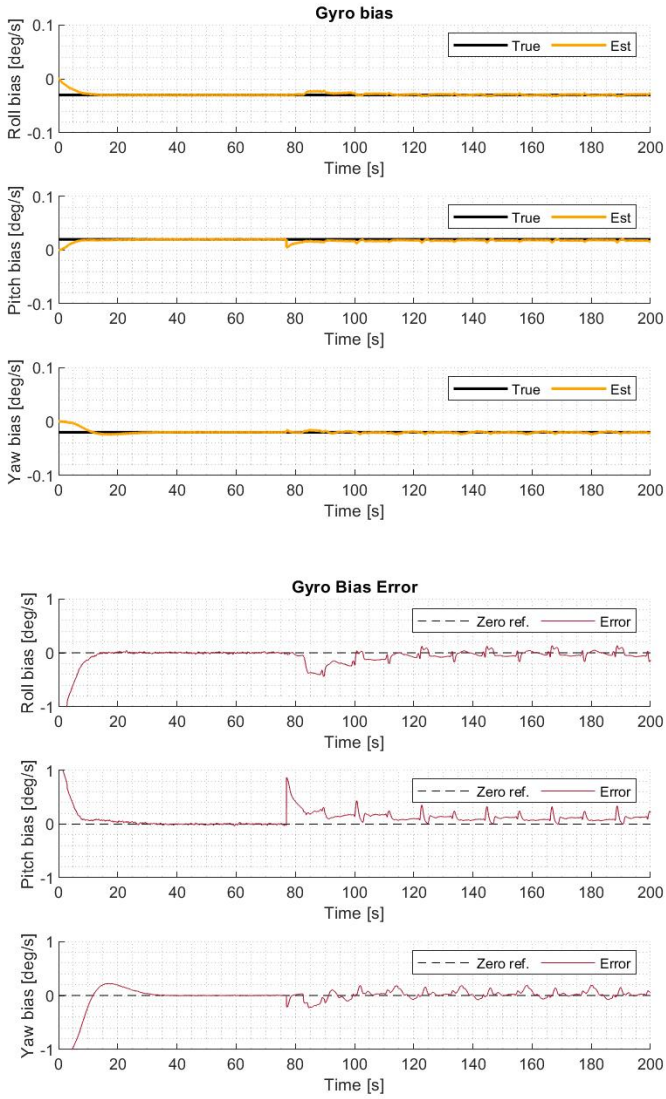


Figure I.5: Results of gyroscope bias estimation when the gravity estimation is omitted

Results with Single GNSS Aiding

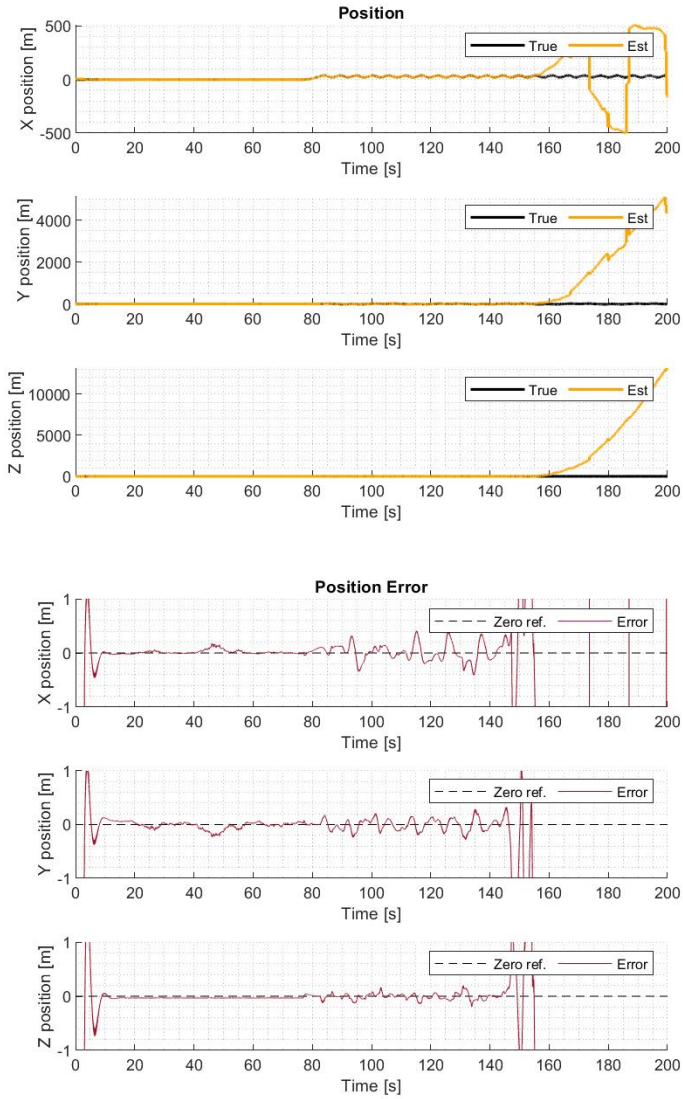


Figure J.1: Results of position estimation when the gravity estimation is omitted

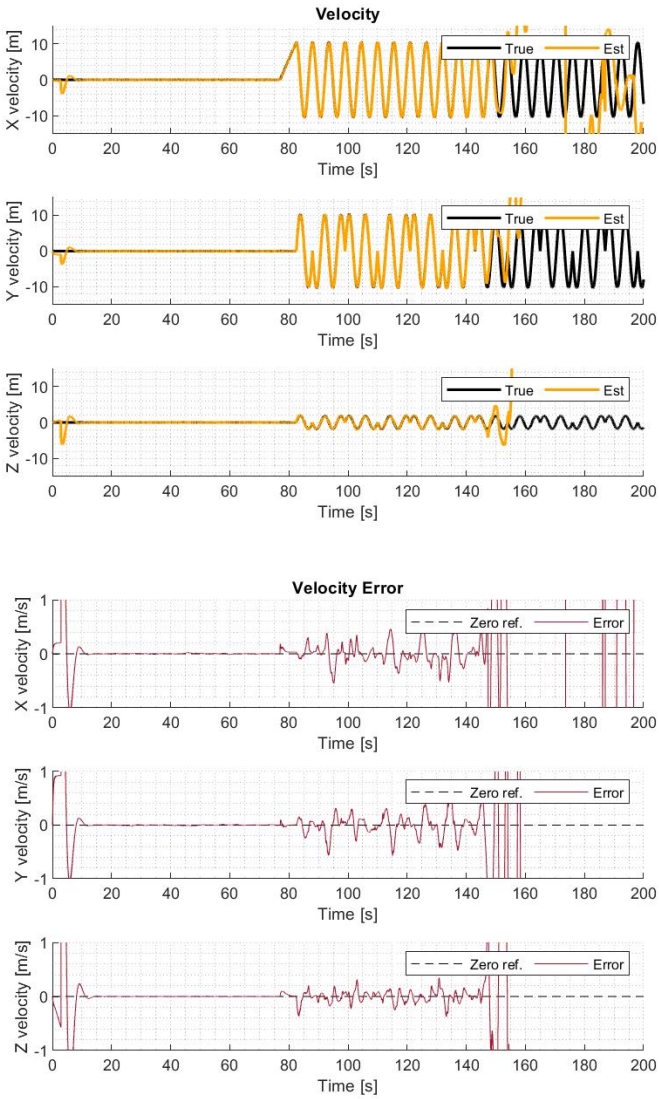


Figure J.2: Results of velocity estimation when the gravity estimation is omitted

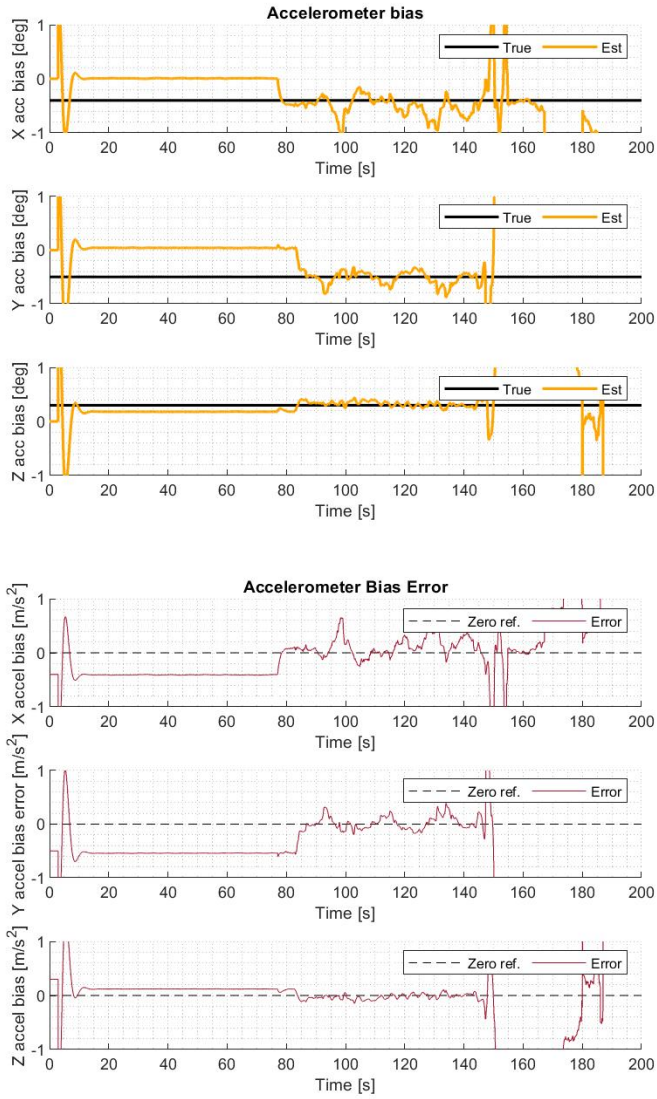


Figure J.3: Results of accelerometer bias estimation when the gravity estimation is omitted

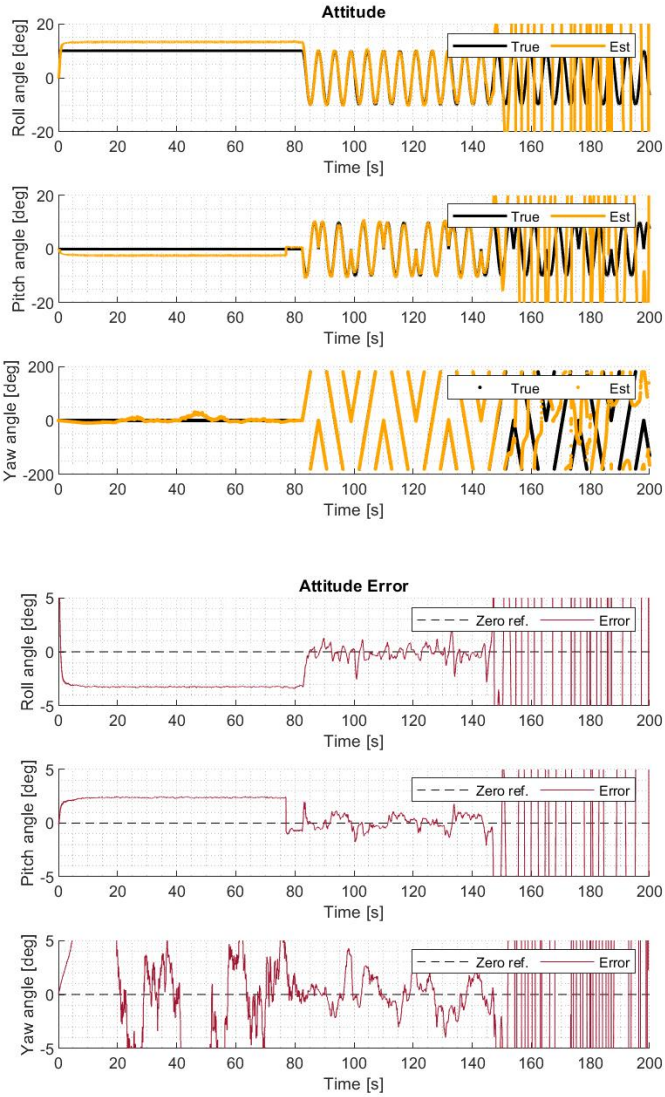


Figure J.4: Results of attitude estimation when the gravity estimation is omitted

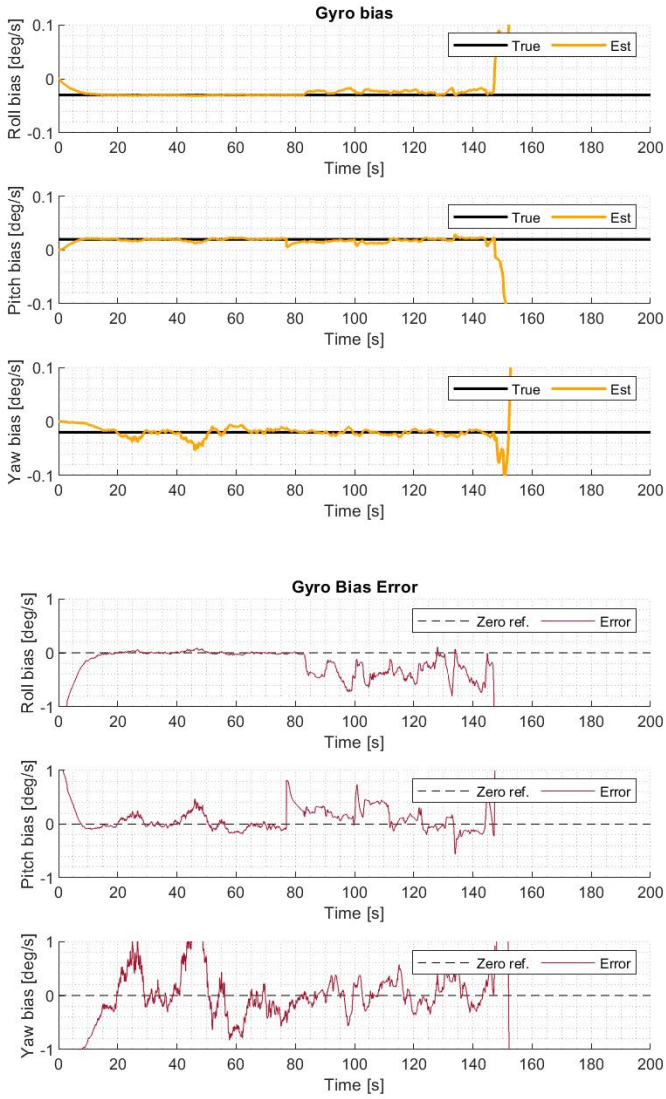


Figure J.5: Results of gyroscope bias estimation when the gravity estimation is omitted

References

Basler (2020). Knowledge base. https://www.baslerweb.com/en/sales-support/knowledge-base/(Accessed: 26.05.2020).

Brumback, B. and Srinath, M. (1987). A chi-square test for fault-detection in kalman filters. *IEEE Transactions on Automatic Control*, 32(6): 552–554.

Bryne, T. H. and Fossen, T. I. (2019). Lecture notes on aided inertial navigation systems. Technical report, Norwegian University of Science and Technology.

Consumer Reports (2020). Cars with advanced safety systems. https://www.consumerreports.org/car-safety/cars-with-advanced-safety-systems/(Accessed: 14.06.2020).

Farrell, J. A. (2008). *Aided Navigation*. The McGraw-Hill Companies.

Fossen, T. I. (2019). Lecture notes on error-state kalman filters for aided ins. Technical report, Norwegian University of Science and Technology.

FSG (2020). Formula student rules 2020. https://www.formulastudent.de/fileadmin/user_upload/all/2020/rules/FS-Rules_2020_V1.0.pdf (Accessed: 17.03.2020).

Gade, K. (1997). Integrering av treghetsnavigasjon i en autonom undervannsfarkost. Technical report, Forsvarets Forskningsintitutt.

Gringer, B. (2020). History of the autonomous car. https://www.titlemax.com/resources/history-of-the-autonomous-car/ (Accessed: 23.05.2020).

Hesai (2020a). *20-Channel Mechanical LiDAR User's Manual*. Building L2, Hongqiao World Center, Shanghai.

Hesai (2020b). *40-Channel Mechanical LiDAR User's Manual*. Building L2, Hongqiao World Center, Shanghai.

IMechE (2020). About formula student. https://www.imeche.org/events/formula-student/about-formula-student(Accessed: 17.03.2020).

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME — Journal of Basic Engineering*, 82(Series D): 35–45.

Larsen, T. D., Andersen, N. A., Ravn, O., and Poulsen, N. K. (1998). Incorporation of time delayed measurements in a discrete-time kalman filter. In *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, volume 4, pp. 3972–3977 vol.4.

Maybeck, P. S. (1982). *Stochastic Models, Estimation and Control*. Academic Press.

NIPH (2020). Facts about the virus and covid-19 disease. https://www.fhi.no/en/op/novel-coronavirus-facts-advice/facts-and-knowledge-about-covid-19/facts-about-novel-coronavirus/(Accessed: 20.05.2020).

Pegasem (2019). *Ground Speed Manual V1.44*. Egerlaenderstr. 1, 86720 Noerdlingen.

Perez, T. and Fossen, T. I. (2011). *Motion Control of Marine Craft*. The CRC Press, second edition. ISBN 1420073648.

Plett, G. L. (2004). Extended kalman filtering for battery management systems of lipb-based hev battery packs. Technical report, Department of Electrical and Computer Engineering at the University of Colorado.

Revolve (2020). Revolve NTNU - from student to engineer in a year. https://www.revolve.no/(Accessed: 19.03.2020).

Skibelid, A. B. (2019). Odometry, mapping and localisation of an autonomous race car for revolve ntnu. Master's thesis, Norwegian University of Science and Technology.

Solà, J. (2017). Quaternion kinematics for the error-state kalman filter. *CoRR*, abs/1711.02508.

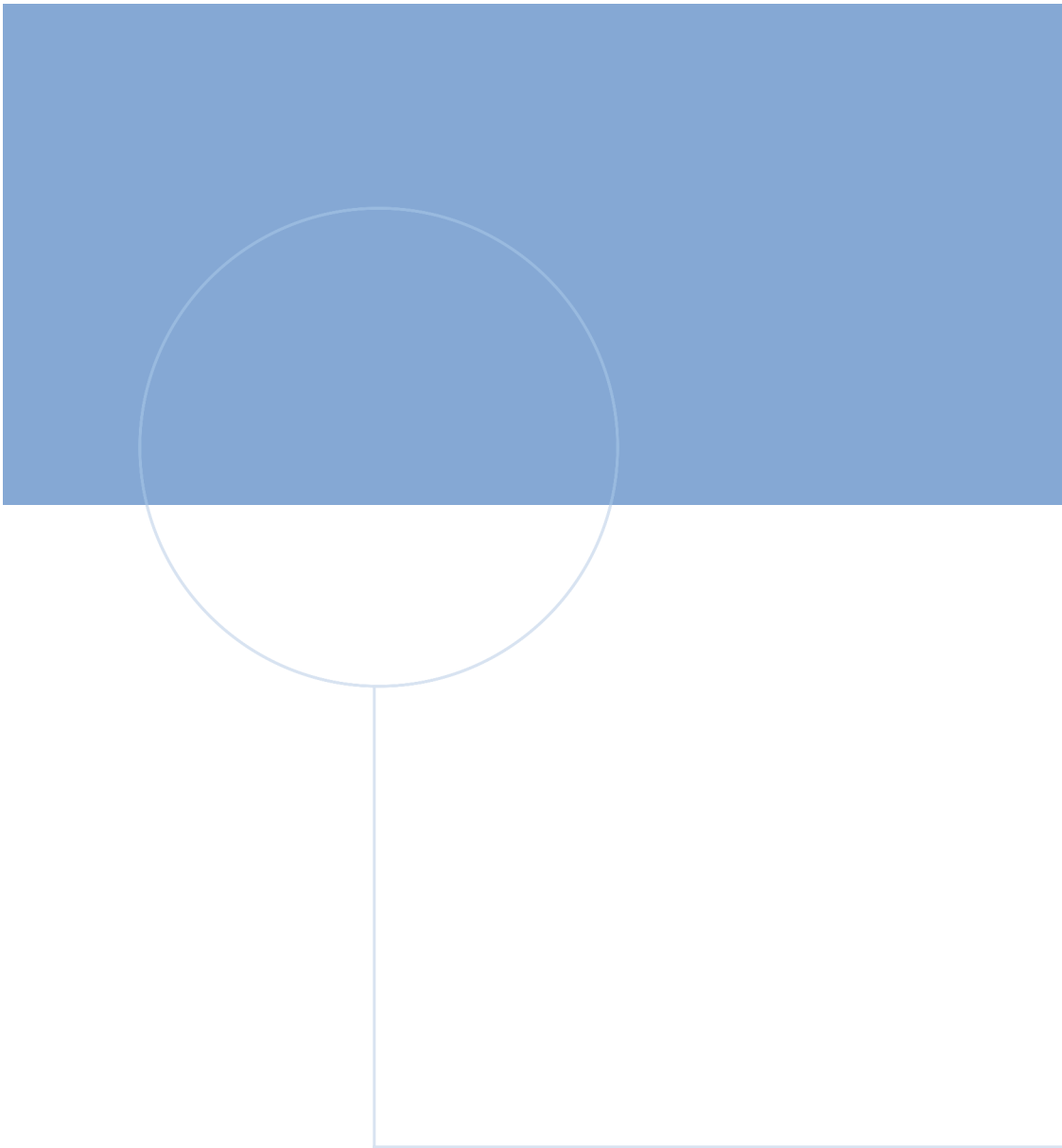
Synopsys, I. (2020). What is an autonomous car. https://www.synopsys.com/automotive/what-is-autonomous-car.html(Accessed: 14.06.2020).

Valls, M. I., Hendrikx, H. F. C., Reijgwart, V. J. F., Meier, F. V., Inkyu Sa, R. D., Gawel, A., Bürki, M., and Siegwart, R. (2018). Design of an autonomous racecar: Perception, state estimation and system integration. Technical report, The Autonomous Systems Lab, ETH Zurich.

VectorNav (2020). Vectornav technologies - MEMS-based inertial sensors. https://www.vectornav.com/(Accessed: 20.03.2020).

Wischnewski, A., Stahl, T., Betz, J., and Lohmann, B. (2019). Vehicle dynamics state estimation and localization for high performance race cars. Technical report, Department of Mechanical Engineering, Technical University of Munich.

Woollaston, V. (2020). How google's driverless cars work? https://www.alphr.com/cars/7038/how-do-googles-driverless-cars-work(Accessed: 23.05.2020).



NTNU

Norwegian University of
Science and Technology