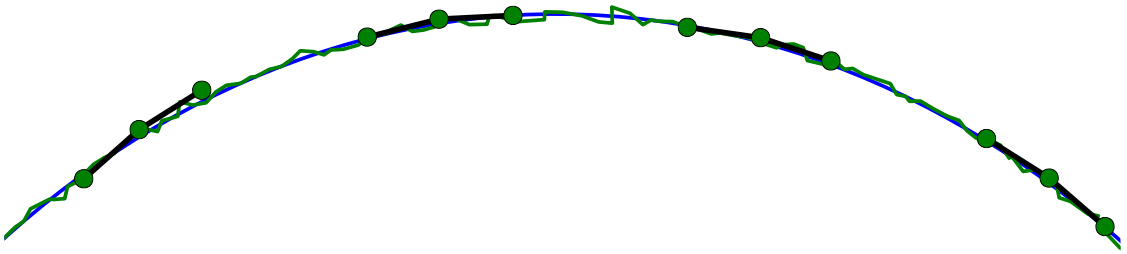
CHALMERS
UNIVERSITY OF TECHNOLOGY



Navigation of an Autonomous Vehicle & Urban Environment Demands Analysis

Development and evaluation of navigation methods & a quantitative study on the demands of an urban operational environment

Master's thesis in Systems, Control & Mechatronics

TOBIAS EKDAHL
KLARA PERSSON

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Master's thesis 2020

Navigation of an Autonomous Vehicle & Urban Environment Demands Analysis

Development and evaluation of navigation methods & a quantitative study on the demands of an urban operational environment

TOBIAS EKDAHL
KLARA PERSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Signal processing and Biomedical engineering
Signal processing
Chalmers University of Technology
Gothenburg, Sweden 2020

Navigation of an Autonomous Vehicle & Urban Environment Demands Analysis
Development and evaluation of navigation methods & a quantitative study on the demands of an urban operational environment
TOBIAS EKDAHL
KLARA PERSSON

Supervisor: Damir Dzebo, Semcon Sweden AB
Supervisor and examiner: Tomas McKelvey, Department of Electrical Engineering

Master's Thesis 2020
Department of Electrical Engineering
Signal processing and Biomedical engineering
Signal processing
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A simulation of the vehicle, where the blue trajectory represents the true position and the green is the estimate.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2020

Navigation of an Autonomous Vehicle & Urban Environment Demands Analysis
Development and evaluation of navigation methods & a quantitative study on the demands of an urban operational environment
TOBIAS EKDAHL
KLARA PERSSON
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This thesis focuses on investigating two parts related to navigation of a maintenance vehicle, used for snow plowing in urban environment. The first part is a comparative study, which evaluates the performance of four different navigation methods, that estimate position and heading of the vehicle. The evaluation was based on both simulated experiments and experiments using real sensor data. The results of the simulations showed that the method that estimates position most accurately was built upon extended Kalman filter (EKF) theory, utilizing measurements from one GNSS unit, and achieved an RMSE (root-mean-square error) of 0.027 m. Furthermore, the method that most accurately estimated heading was an EKF that fused information from a GNSS unit, an IMU and a magnetometer, achieving an RMSE of 0.59°. The experiments based on logged, real sensor data, showed that the EKF that utilizes one GNSS unit was superior to the other tested methods. Thus, when considering all results, this method was concluded to be the most reliable navigation method among the tested. The second part of the thesis was a quantitative study, where navigation in urban environment was simulated and analyzed. The simulation was built upon the assumption that GNSS measurements were not available due to disturbances caused by the surrounding environment. Therefore, other sensors setups, that could supplement the system in order to obtain sufficiently accurate navigation, were tested. The different sensor setups that were tested, were configured of combinations of radars, cameras and ultrasonic sensors. Another aspect of the simulations was to investigate the demands on the operational environment. More specifically, this was modelled by generating reference points along the path that the vehicle was driving, which could be perceived by the sensors. It was concluded that the most well-functioning sensor setup that was tested, used five radars and cameras, and six ultrasonic sensors, evenly spread to get a 360° view around the vehicle. Furthermore, it was concluded that the mentioned setup required a reference point every sixth meter along the road to achieve sufficiently accurate navigation.

Keywords: navigation, sensor fusion, Kalman filter, EKF, GNSS, IMU, magnetometer, maintenance vehicle, articulated vehicle, urban navigation.

Acknowledgements

First and foremost, we would like to thank Mats Larsson and Damir Dzebo at Semcon Sweden AB for giving us the opportunity to work with this project. Even though it sometimes has been very challenging, it has also been very developing. Furthermore, we would like to thank our supervisor and examiner, Tomas McKelvey, for interesting discussions and the time you spent on our project.

Finally, we would like to take the opportunity to thank our families for always being close, despite the physical distance between us. You have always been by our sides during all these years. Without your comfort and support, we would never have made it this far.

Tobias Ekdahl & Klara Persson, Gothenburg, May 2020

Contents

List of Figures

List of Tables

1

Introduction

In this first chapter of the report, the project will be described briefly. The chapter starts with a concise background that motivates why the study was carried out and puts the work into a broader context. Thereafter, an overview of the contributions of the project will be presented, followed by a section with related work. This is followed by three sections that present the aim of the project, the research questions that were stated and boundaries. The chapter ends with a section about the outline of the thesis, that provides a short summary of the remaining chapters and their content.

1.1 Background

The technology behind autonomous drive (AD) is constantly under development and is facing new challenges on every step of the way [1]. One of these challenges is related to navigation of the vehicle in a safe and reliable way [2]. Navigation is defined by two main tasks. The first is determining the *pose* (position and heading), and second is path planning [3]. Thus, navigation is a vital part and the key to a safe and accurate AD. Without a consistently accurate estimate of the pose of the vehicle, it is highly possible that the vehicle will cause danger, as it diverges from where it is expected to be located and where it actually is. Furthermore, a well-functioning path-planning system can not be constructed based on erroneous pose estimates [4].

This thesis has been executed in collaboration with Semcon Sweden AB. Today, Semcon develops and delivers a concept solution of an autonomous maintenance vehicle for snow plowing, as a partner in the research project, *Barmark*, led by Research Institutes of Sweden (RISE). The vehicle is supposed to operate in urban areas around for example pedestrians and cyclists. This requires high demands on the precision of the pose estimate. At the same time, it is a known problem that navigation in urban environments raises some difficulties. For example, position estimates based on information from GNSS (Global Navigation Satellite System) lacks in precision if high buildings, trees and other elements are blocking or disturbing the satellite signal [5].

1.2 Overview

As previously mentioned, it is of great importance that pose estimation is consistently accurate, and if an autonomous vehicle is to operate in an urban environment, other difficulties apply and need to be dealt with as well. One example of this is blocking of satellite signals. These challenges and difficulties have been solved by others in various ways, which will be discussed in more detail in the next section. A common approach is to use sensor fusion to merge observations from a GNSS unit and an inertial measurement unit (IMU) to have both global and local information about the vehicle. Another approach is to use all different kinds of sensors, e.g. camera, LiDAR (light detection and ranging) and radar, and utilize the provided information in different ways in order to get a more explicit overview than what is obtained from fusing data from GNSS and IMU only.

The list of different approaches can be made long. However, one aspect that many of the previous studies do not investigate, that this thesis focuses on, is the comparison of different navigation methods. Furthermore, as mentioned in Section 1.1, navigation also includes path planning, but as this is a large field of study on its own, this has not been included in the scope. Further on, when using the term navigation, only pose estimation is referred to. In this project, four different methods for two-dimensional pose estimation have been implemented and evaluated. Hence, this thesis contributes with insights about the performance of different navigation methods in comparison to each other.

Another contribution is an early development of a tool for simulating navigation, based on different sensor setups that can be used in urban environment, where measurements from GNSS are unavailable. The results of the simulation can be used in order to test different sensor setups, get insights about the demands on the operational environment and how different weather conditions complicate navigation. This is an economical and efficient way to test different setups before investing in hardware.

1.3 Related work

As mentioned in Section 1.2, there are many ways to solve and deal with the associated challenges and difficulties with navigation. What others have done before will be presented in more detail in this section.

In [6], the authors discuss the use of a radar sensor surveying preplaced radar beacons, with known positions, for autonomous land vehicle navigation. To estimate position, the information from the radar is fused via an *extended Kalman filter* (EKF) together with data from encoders that give steering angle and velocity. The final result is that the radar successfully observes the preplaced beacons, showing that a radar can perform well where optical sensors fail, e.g. in rainy or dusty weather conditions. However, one critical drawback with this approach is that the

system only gets data from one radar and if that radar fails, no other sensors are available to supplement the system.

To deal with the problem of a non-robust system, one approach is to fuse the local sensors with information from a global sensor. In [7], this is done by fusing GNSS and IMU data in order to increase position accuracy in a city. The common method of fusing the data with an EKF is used, and the conclusion is that the fusion produces a much better result than when exclusively using measurements from a GNSS unit. However, this is just as expected as the research has been carried out in a dense area where it is known that a GNSS solution lacks in accuracy because of satellite outage. One drawback of this approach, in comparison to this thesis, is that only the position is estimated and not the heading, which is an important component for accurate navigation. Another example of previous work about algorithms for GNSS and IMU integration can be found in [8]. One part of this thesis is built upon the same theory, which will be presented in more detail later in the report. They discuss position and orientation estimation using inertial sensors and provide a lot of useful theory and algorithms that show how the different estimates can be computed via fusion with EKF. A small shortcoming is that they do not provide an explicit algorithm for pose estimation where fusion is made using measurements from a GNSS unit and an IMU. Inspiration and background theory has been taken from their work, though it has been modified and developed in order to get the desired results for this work.

Another resource for this thesis is [9], where an algorithm for heading estimation with EKF, based on positioning data retrieved from two GPS receivers, is presented. How they model measurements from two GNSS units has been applied in this thesis as well, but the significant difference is that a comparison between different algorithms has been conducted in this thesis, which is something that their work does not focus on.

The common denominator for all approaches explained in this section is that they are all using EKF. However, another filter applicable for both IMU units and MARG (IMU and magnetometer) sensors is the *Madgwick filter* [10]. This is a novel orientation filter, which addresses the common issues related to Kalman-based approaches, namely computational load and parameter tuning. Even though this seems very promising and maybe could result in a very accurate result, it was decided to use EKF in this thesis. That is because the Kalman filter is very established and the theory is well documented and tested on numerous different applications.

1.4 Aim

The aim of the project was to implement four different navigation methods for a maintenance vehicle, such as the one at Semcon, and evaluate them relative to each other. Furthermore, the aim was also to investigate navigation in urban environment via a quantitative study. The quantitative study is based on a self-developed simulation tool where sensor setups, weather conditions and demands on

the operational environment can be studied.

1.5 Research questions

In this section, the specific questions related to the investigated areas of the project, are stated. These research questions are used to define the scope of the project.

To evaluate the four navigation methods, the following questions will be answered:

- Given a type of navigation method and its required equipment, what performance in terms of estimating the pose of the vehicle can be achieved?
- What method, among the tested, results in the most accurate navigation?

For the quantitative study of navigating in urban environment, the following questions will be answered:

- What is an acceptable navigation accuracy for a maintenance vehicle, when operating in urban environment?
- Given a sensor setup, what is demanded of the operational environment, in order to assure that the required navigation accuracy is reached?
- Given a sensor setup and an operational environment, what navigation accuracy can be achieved?
- How do the weather conditions; dark and precipitation, affect navigation accuracy?

1.6 Boundaries

The scope of the project was limited by stating some important boundaries. This is simply because the project needed to be restricted to the available time and resources. The considered boundaries are the following:

- Four different navigation methods were considered and evaluated. The ones that were chosen are considered to be well documented and established. However, these are not the only applicable methods and there are many other methods that could be evaluated.

- The results of the project are bounded by the specific dynamics of the maintenance vehicle that was used throughout the project. The results can therefore not be directly applicable to a vehicle with other dynamics. This is because the scope would be too comprehensive if other vehicle dynamics would be considered.

- The vehicle is only equipped with one GNSS unit and one antenna, thus the performance of adding an additional device will only be evaluated through simulations. This boundary was set due to budget limitations.

- The simulation of the urban environment is very simplified due to time limitations. Therefore, the simulation does not include a model of a realistic "real world" scenario due to complexity.

1.7 Thesis outline

First, in Chapter 2, *Theory*, the fundamental knowledge and ideas about concepts concerning the thesis are presented. This is to make it easier to follow and understand the rest of the report. Following are two consecutive method chapters, Chapter 3, *Method - Implementation and Evaluation of Navigation Methods* and Chapter 4, *Method - Urban Simulation*. It was decided to split the method chapter into two separate for simplicity for the reader. The results and related discussions, from applying the methodologies in Chapter 3 and 4, are separately presented, in Chapter 5, *Results & Discussion - Implementation and Evaluation of Navigation Methods*, and Chapter 6, *Results & Discussion - Urban Simulation*. The report ends with Chapter 7, *Conclusions & Future Work*, where a summary of the study is given and it is clearly stated what can be said about the outcome of the project, based on the research questions.

2
Theory

In this chapter, fundamental theory and concepts related to the project are explained. First, the maintenance vehicle and its sensors, that the study was based on, are presented and briefly explained. As all the evaluated navigation methods utilize data from one or multiple GNSS units, the general drawbacks of the sensor in urban environment, and the most common error sources for GNSS are explained. This is followed by a section that explains the concept of *real-time kinematic* (RTK), which is a technology that can compensate for many of the stated error sources. Also, since the two-dimensional pose of the vehicle is estimated in the first part of the project, GNSS data needs to be converted to a more comprehensible coordinate system. Therefore, the theory of how GPS data (which is included in GNSS) is converted from latitude and longitude to Cartesian coordinates in an ENU-frame (east-north-up). The last sections focus on explaining the basics of the nonlinear Kalman filter EKF and its properties.

2.1 The maintenance vehicle

The maintenance vehicle is meant to operate autonomously for snow plowing and can be seen in Fig. 2.1. However, throughout the project, the brushes that can be seen in front of the vehicle will not be considered. The vehicle goes under the category of an articulated vehicle, meaning that it can be modelled as two rigid bodies, connected via a joint. For the vehicle to turn, the relative angle between the two bodies is regulated via hydraulics. Meaning that the vehicle will go straight if there is no angle between the two bodies and move in a circular motion if the angle is fixed and nonzero. This type of steering allows the vehicle to make very sharp turns. Another characteristic is that the vehicle is moving at low speed and in experiments during this study it rarely exceeded 10 km/h.

2.2 System setup and hardware

Various sensors are mounted on the vehicle. Besides the GNSS for global positioning and a steering angle sensor, providing the steering angle and steering angle rate, an IMU (accelerometer and gyroscope), magnetometer, radar, ultrasonic sensor and a camera for local measurements are available. The different sensors will be explained in this section.

Figure 2.1: Picture of the maintenance vehicle. On the roof of the cab, the mounted GNSS unit can be seen as a white disk.

2.2.1 GNSS unit

The Global Navigation Satellite System is a general term, for all satellite navigation systems, which provide signals for position and timing measurements to a GNSS receiver [11]. The different navigation systems that goes under the term is: Europe's *Galileo*, the American *NAVSTAR Global Positioning System* (GPS), Russia's *Global'naya Navigatsionnaya Sputnikovaya Sistema* (GLONASS) and China's *BeiDou Navigation Satellite System*.

If the conditions are right, GNSS can deliver positioning data with very high accuracy. In this project, the GNSS data is also corrected with RTK (which will be explained in more detail in Section 2.4), which improves the accuracy even more and compensates for different error sources [12]. The specific accuracy for the GNSS unit is summarized in Table 2.1. The accuracy is taken from the datasheet and is specific for the unit mounted on the vehicle [13]. The larger uncertainty in the vertical

Table 2.1: RTK accuracy of the position measurements from the GNSS unit that is mounted on the vehicle.

Direction	**Standard deviation** [m]
horizontal	0.010
vertical	0.015

component, seen in Table 2.1, is because the receiver can not observe satellites below the horizon. Which results in weaker vertical constraints and hence, larger uncertainties [12].

2.2.2 Inertial measurement unit and magnetometer

Inertial sensors are commonly used for position and orientation estimation [8]. The considered sensors are accelerometer, gyroscope and magnetometer. Via double integration and after subtraction of the Earth's gravity of the accelerometer data, an estimate of the sensor's local position is obtained. For the gyroscope, that measures angular velocity, the data can be integrated, which gives the sensor's orientation. The magnetometer data is used to estimate orientation with respect to the magnetic North [14]. In an ideal world, where the initial position and orientation are known and the mathematical models perfectly transform the measurement data, this would provide exact information about the pose of the sensor, in reference to the starting position. However, this is not the case as the data usually is both noisy and biased.

2.2.2.1 Mathematical model of accelerometer measurements

A 3D accelerometer measurement, $\mathbf{y}_a$ can be modelled as

$$\mathbf{y}_a = \mathbf{f}^s + \boldsymbol{\delta}_a^s + \mathbf{e}_a^s, \tag{2.1}$$

where $\mathbf{f}^s$ is the external specific force, $\boldsymbol{\delta}_a^s$ is the sensor bias and $\mathbf{e}_a^s$ is Gaussian distributed sensor noise [8]. The superscript s indicates that the measurement components are with respect to the sensor frame, that is the fixed coordinate system relative to the sensor unit. The specific force is defined as

$$\mathbf{f}^s = \mathbf{R}^{s/w}(\mathbf{a} - \mathbf{g}), \tag{2.2}$$

where $\mathbf{a}$ denotes the linear acceleration of the sensor and $\mathbf{g}$ the acceleration due to Earth's gravity. Both terms are given with respect to the world frame, w, which is a fixed coordinate system relative to the movement of the sensor. $\mathbf{R}^{s/w}$ is the rotation matrix from the world frame, to the sensor frame, which is indicated by the superscript. By inserting Eq. (2.2) in Eq. (2.1) the final mathematical model of accelerometer measurements are given by

$$\mathbf{y}_a = \mathbf{R}^{s/w}(\mathbf{a} - \mathbf{g}) + \boldsymbol{\delta}_a^s + \mathbf{e}_a^s. \tag{2.3}$$

2.2.2.2 Mathematical model of gyroscope measurements

A 3D gyroscope measurement, $\mathbf{y}_\omega$, can be modelled as

$$\mathbf{y}_\omega = \boldsymbol{\omega}^s + \boldsymbol{\delta}_\omega^s + \mathbf{e}_\omega^s, \tag{2.4}$$

where $\boldsymbol{\omega}^s$ is the angular velocity of the sensor, $\boldsymbol{\delta}_\omega^s$ is the sensor bias and $\mathbf{e}_\omega^s$ is Gaussian sensor noise [8].

2.2.2.3 Mathematical model of magnetometer measurements

A 3D magnetometer measurement, $\mathbf{y}_m$, can be modelled as

$$\mathbf{y}_m = \mathbf{R}^{s/w}\mathbf{m} + \mathbf{e}_m^s, \tag{2.5}$$

where $\mathbf{m}$ is the earth magnetic field and $\mathbf{e}_m^s$ is Gaussian noise [8].

2.2.3 Additional sensors

The additional sensors that can be used for localization are radar, ultrasonic and camera. A short summary of how they work will be explained in this section. The explanations are very brief and general because no in-depth knowledge about how they work is of importance for this project.

2.2.3.1 Radar

In this project, an FMCW (Frequency-Modulated Continuous-Wave) radar has been considered. This is a special type of radar that radiates continuous transmission power and it can change its operating frequency during the measurement [15]. The advantage with frequency modulation is that the target range can be determined because of a timing mark. The timing mark comes from that a transmitted signal increases or decreases in frequency periodically and when the echo signal is received, that change of frequency gets a delay. Meaning, that the distance is measured by comparing the frequency of the received to a reference that usually is the transmitted signal. The distance, D, to the reflected object can be calculated by

$$D = \frac{c_0|\Delta t|}{2} = \frac{c_0|\Delta f|}{2\left(\frac{d(f)}{d(t)}\right)}, \tag{2.6}$$

where c_0 is speed of light, Δt is delay time in seconds, Δf is the measured difference in frequency and df/dt is the frequency shift per unit of time.

2.2.3.2 Ultrasonic

An ultrasonic sensor can be used to measure distance to a target [16]. This is done by an ultrasound pulse that is transmitted and then reflected black, when the signal hits an object, to the transducer. The distance, d, to the target can be calculated by

$$d = \frac{vt}{2}, \tag{2.7}$$

where v is the velocity of sound and t is the propagation time. The expression is divided by two as the propagation time is the time for the signal to go back and forth.

2.2.3.3 Camera

Pose estimation with a vision-based system is often used as a supplement to a GNSS based navigation method [17]. The system can give valuable information in environments where the GNSS is not working properly, e.g. in a city. There exist a lot of different techniques and algorithms for pose estimation with a vision-based system. However, this will not be further investigated because the useful information, for this project, is how well a vision-based system can estimate position and heading. Therefore, the process of going from the output of the camera to a pose estimate will not be explained.

2.3 Error sources for GNSS data

Even though GNSS, under favorable conditions, can deliver very precise and accurate position measurements some error sources can cause errors up to several meters [18]. Most of the errors can be taken care of if RTK is used, more on this in the next section. Here, the most common error sources will be explained.

2.3.1 Multipath

Multipath is a problem that arises when the satellite signal is reflected on objects in the surrounding environment before the GNSS antenna receives the signal [19]. Surrounding buildings can for example cause these reflections as the signal can hit a wall and then the antenna. The phenomenon is depicted in Fig. 2.2. The reflections entail that the signal travels farther than a true signal, known as the line of sight (LOS) signal. That is because a LOS signal travels in a straight path between the satellite and the antenna. The delay of the signal will result in an error on position estimates, that is based on this flawed data. As multipath arises from obstacles in the local environment, such as buildings in an urban environment, it is very hard, or even impossible, to model. One recommendation is to set the elevation mask to 13-15 degrees to minimize multipath [20]. The elevation mask is a degree limit to the horizon. Meaning that not all visible satellites are used, only the ones above the limit [18].

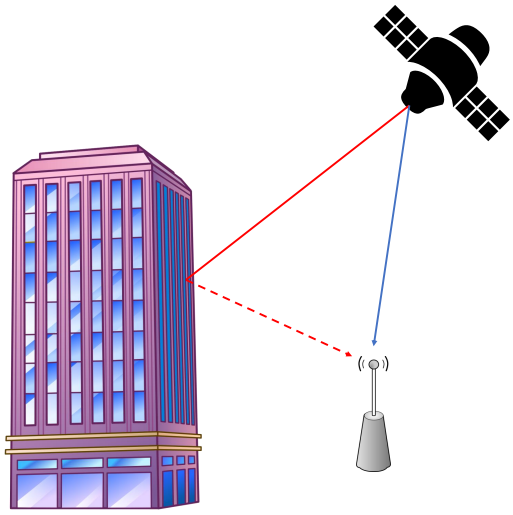


Figure 2.2: Illustration of the problem with multipath. In the occurrence of multipath, the satellite signal first hits a wall on a building and then the receiver, which in turn makes the path longer than the LOS signal.

2.3.2 Satellite clocks and orbit errors

Every GNSS satellite is equipped with a very accurate atomic clock but they tend to drift [19]. Even a very small inaccuracy will result in a significant error on the position measurements. Therefore, clock errors must be considered.

Another error that is related to the satellites is their orbit. The orbit a satellite is traveling in is very accurate and well known, but small deviations exist [19]. This small inaccuracy will, just like the satellite clock, arise a large error in the position measurements. The error is estimated to be ±2.5 m.

2.3.3 Ionospheric and tropospheric delay

The ionosphere is a layer of the atmosphere and is around 60-800 km above the ground [19]. The layer contains a large number of ionized atoms, called ions, which are electrically charged particles [21]. The atoms have been ionized due to interaction with solar radiation, and these ions delay the satellite signal. The ionospheric delay varies a lot depending on for example solar activity, time of the year, time of the day and location, which makes it hard to predict how much the delay will affect the resulting position estimate.

Closest to the surface of the earth is the layer of the atmosphere that is called the troposphere. How much the troposphere will interrupt the satellite signal and cause a delay is dependent on, for example, humidity, temperature and atmospheric pressure in the troposphere.

2.4 Real-time kinematic

Real-time kinematic (RTK) is a method to improve the position measurements from GNSS [12]. The method is based on received GNSS-signals at a stationary reference point with known coordinates, and these known coordinates are used to correct the positioning of a roving receiver at another location, e.g. on a vehicle. Because of perturbations caused by the ionosphere, troposphere and clocks, the received GNSS-signals at the reference point differ from the true coordinates and as the coordinates of the stationary reference point are known, this information can be used to mathematically correct the position. By applying similar corrections to the distant rover, its position can be determined with high accuracy. However, because of the distance between the receiver and the rover, the GNSS-signals have been perturbed differently, and the mathematical corrections introduce some uncertainties in the positioning of the roving receiver. To deal with the latter uncertainties, Network RTK can be used, which utilizes correction data from many reference stations. Because of the use of several reference points, it is also possible to increase the distance between the reference station and rover without any loss of precision in positioning.

Site specific characteristics that differ between the receiver's and the rover's environment cannot be corrected with Network RTK and will, therefore, propagate to the resulting positioning of the rover [12]. Site specific error sources for the receiver are for example electromagnetic disturbances, stability of antenna with respect to movement of the ground and signal delay because of moist, frost or snow. Other error sources for the rover can for example be lengths, orientation and tilt of

the vehicle, on which the rover is mounted. Hence, rover calibration is important to decrease errors of this kind. Despite the existence of all different error sources, the resulting error according to [12] sum up to a total RMSE (root-mean-square error), in a nominal situation, of 27.3 mm vertically and 12.0 mm horizontally. However, if multipath is the biggest source of error, RTK can not deal with this problem.

2.5 Convert GPS latitude and longitude to ENU coordinates

GPS is a satellite navigation system owned by the American government [22]. The position coordinates are given in the unit of latitude and longitude. Since latitude and longitude provide a position point on the elliptical surface of the Earth, these coordinates need to be projected on a geographical map coordinate system. One established map projection is the *Universal Transverse Mercator* (UTM), which consists of a series of sixty map projection zones, each covering six-degree sections of longitude [23]. The UTM coordinates are given by two parameters, namely northing and easting, both in the direction of the respective cardinal direction (north and east). The projection zones are further divided into subzones, which creates a grid over the globe. Each subzone is denoted by a number corresponding to the longitude section, followed by a letter corresponding to the zone. For example, Gothenburg is located in the subzone 32V.

Although UTM is a more appropriate coordinate system to represent the 2D pose of the vehicle, rather than latitude and longitude, the zones cover very large areas of the surface of the Earth. Therefore, the UTM coordinates were converted into a local ENU-coordinate system (east-north-up), where the origin was set as the first measured UTM-coordinate. This results in that the initial position of the vehicle is zero and therefore first measured UTM-coordinate was subtracted from all the following measurements. However, ENU is a three-dimensional coordinate frame [24], and thus the U-coordinate was always considered to be zero since this was not of interest in this study.

2.6 Dynamical model of an articulated vehicle

As mentioned in Section 2.1, an articulated vehicle can be modelled as a front and a rear body connected by a joint, as illustrated in Fig. 2.3. The lengths of the front and rear body are denoted ℓ_1 and ℓ_2 respectively. The heading of the front and the rear body is given by θ_1 and θ_2 respectively (the angles between the respective bodies and the positive x-axis). The steering angle is denoted ϕ (the angle of the front body relative to the rear body). The notation for the speed of the vehicle is v. Lastly, the position of the front body is denoted (x_1, y_1) and (x_2, y_2) is the position of the rear body. By using an approach inspired by P. I. Corke and P. Ridley [25], a dynamical model of the vehicle could be derived.

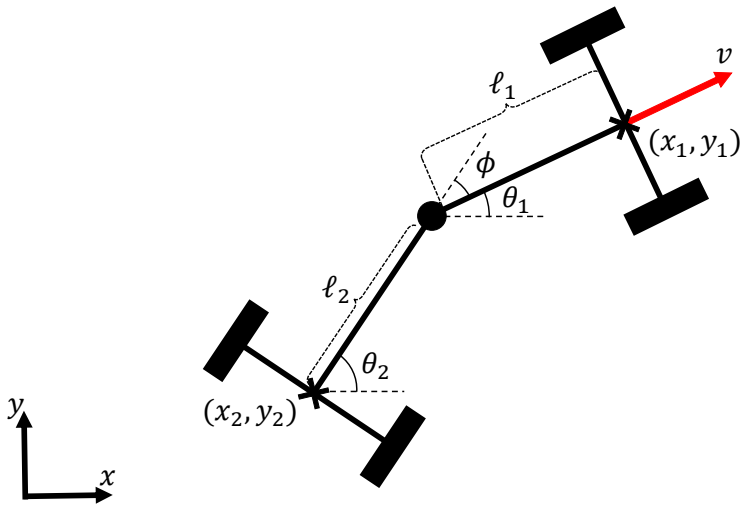


Figure 2.3: A theoretical model of the dynamics of an articulated vehicle.

The velocity of the front body can be divided into components in the x and y direction of the global coordinate system

$$\dot{x}_1(t) = v(t)\cos(\theta_1(t)), \tag{2.8}$$
$$\dot{y}_1(t) = v(t)\sin(\theta_1(t)). \tag{2.9}$$

Furthermore, the relationship between the position of the front body and the rear body can be described as

$$x_1(t) = x_2(t) + \ell_2\cos(\theta_2(t)) + \ell_1\cos(\theta_1(t)), \tag{2.10}$$
$$y_1(t) = y_2(t) + \ell_2\sin(\theta_2(t)) + \ell_1\sin(\theta_1(t)). \tag{2.11}$$

Differentiating Eq. (2.10) and Eq. (2.11) gives the relationship between the velocity components of the front body and the rear body

$$\dot{x}_1(t) = \dot{x}_2(t) - \ell_2\dot{\theta}_2(t)\sin(\theta_2(t)) - \ell_1\dot{\theta}_1(t)\sin(\theta_1(t)), \tag{2.12}$$
$$\dot{y}_1(t) = \dot{y}_2(t) + \ell_2\dot{\theta}_2(t)\cos(\theta_2(t)) + \ell_1\dot{\theta}_1(t)\cos(\theta_1(t)). \tag{2.13}$$

Since the vehicle is heavy and moving at low speed it is assumed that there is no slippage between the wheels and the ground. Thus, there is a nonholonomic constraint for each body of the vehicle,

$$\dot{x}_1(t)\sin(\theta_1(t)) - \dot{y}_1(t)\cos(\theta_1(t)) = 0, \tag{2.14}$$
$$\dot{x}_2(t)\sin(\theta_2(t)) - \dot{y}_2(t)\cos(\theta_2(t)) = 0. \tag{2.15}$$

Furthermore, it can be concluded by trigonometry that

$$\phi(t) = \theta_2(t) - \theta_1(t), \tag{2.16}$$

and if differentiated, the steering angle rate is described as

$$\dot{\phi}(t) = \dot{\theta}_2(t) - \dot{\theta}_1(t), \tag{2.17}$$

In order to obtain a dynamical expression of the heading of the front body, as a function of the steering angle and velocity, the relations above can be used. This is done by first substituting Eq. (2.8) into Eq. (2.12), followed by substitutions of Eq. (2.9) and Eq. (2.13) - Eq. (2.17), applying trigonometric identities and reducing the expression. This finally results in

$$\dot{\theta}_1(t) = -\frac{v(t)\sin(\phi(t)) + \ell_2\dot{\phi}(t)}{\ell_2 + \ell_1\cos(\phi(t))}. \tag{2.18}$$

2.7 State estimation with Kalman filter

The theory explained in this section and its related subsections are based on the book by Simo Särkkä [26]. The Kalman filter is a recursive algorithm that includes two steps; a prediction and an update. In the prediction step, prior knowledge about the states of interest are used to make a prediction about the states at the current time step. The prior is represented by the expected value of the prior, $\hat{\mathbf{x}}_{k-1|k-1}$ and its corresponding uncertainties in the form of a covariance matrix, $\mathbf{P}_{k-1|k-1}$. From the prediction step the expected values of the states, $\hat{\mathbf{x}}_{k|k-1}$, as well as the uncertainty, $\mathbf{P}_{k|k-1}$, is obtained. The predicted states are then updated by using observations obtained from sensors, which gives the final expected values (state estimates) for the current time step, $\hat{\mathbf{x}}_{k|k}$, and their uncertainties, $\mathbf{P}_{k|k}$. The final output from the filter is called the posterior. The filter is recursive because it then uses the posterior as the prior knowledge in the prediction step of the next time instant and so forth. The process that was just described is illustrated as a flow chart in Fig. 2.4.

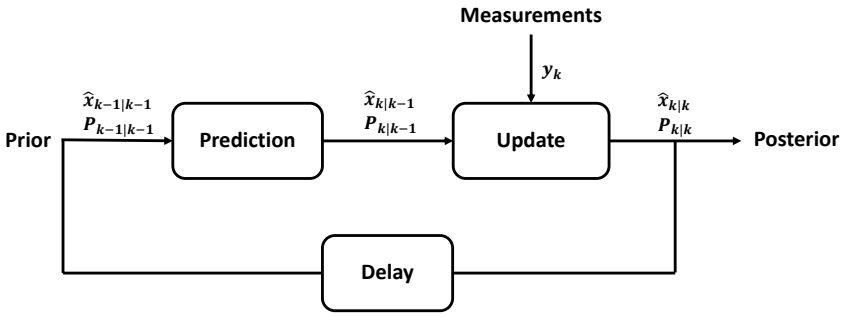


Figure 2.4: Flow chart of the computations in the EKF algorithm.

2.7.1 Extended Kalman Filter

The extended Kalman filter (EKF) is a non-linear version of the Kalman filter. There are other types of non-linear Kalman filter, such as the *cubature Kalman filter* (CKF)

and the *unscented Kalman filter* (UKF). However, according to Särkkä, the strength of the EKF is that the theory is relatively simple compared to its performance.

For the prediction step of the EKF, the following equations apply,

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}), \tag{2.19}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}(\hat{\mathbf{x}}_{k-1|k-1})\mathbf{P}_{k-1|k-1}\mathbf{F}(\hat{\mathbf{x}}_{k-1|k-1})^T + \mathbf{Q}_{k-1}, \tag{2.20}$$

where $\mathbf{f}(.)$ represents a motion model, which is a mathematical model of the dynamics of the agent being observed. $\mathbf{F}(.)$ is the Jacobian of the motion model and $\mathbf{Q}_{k-1}$ is a covariance matrix corresponding to the uncertainties of the motion model.

For the update step of the EKF, the following equations apply,

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})), \tag{2.21}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}^T, \tag{2.22}$$

$$\mathbf{S}_k = \mathbf{H}(\hat{\mathbf{x}}_{k|k-1})\mathbf{P}_{k|k-1}\mathbf{H}(\hat{\mathbf{x}}_{k|k-1})^T + \mathbf{R}_k, \tag{2.23}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}(\hat{\mathbf{x}}_{k|k-1})^T\mathbf{S}_k^{-1}, \tag{2.24}$$

where $\mathbf{K}_k$ is the Kalman gain, $\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})$ is called the innovation and $\mathbf{S}_k$ is the innovation covariance. The innovation consists of two terms. The first, $\mathbf{y}_k$, is a measurement vector containing observations from sensors. The second term is computed by using a measurement model, $\mathbf{h}(.)$, which relates the measurements to the predicted states. Thus, the innovation can be viewed as something that captures the new information, between the prediction and the observations. This means that if the innovation is big, the update of the predicted states will also be big. Furthermore, the remaining components of the equations are $\mathbf{H}(.)$, which is the Jacobian of the measurement model, and $\mathbf{R}_k$ which is the covariance matrix corresponding to the uncertainties of the sensors.

The Kalman gain decides how much each predicted state should be affected by the innovation. This also implies that even though one cannot directly measure a certain state, the state prediction will be updated. For example, if it is desired to estimate the x- and y-position, and the heading of a vehicle, but only noisy measurements of the x- and y-position are accessible, the prediction of the heading can still be updated because of the Kalman gain. The logic behind this is that if the difference between the observations and the output of the measurement model is large, then the states that are not possible to measure probably also needs a large update.

2.7.2 Extended Kalman filter with input variables

A slightly different representation of the EKF equations presented in Section 2.7.1, can be made if some of the measurements are viewed as input variables [27]. The process of the EKF with input variables is visualized in Fig. 2.5. This can be

compared to Fig. 2.4 in order to see the differences between the two versions of the Kalman filter algorithm. The second illustration includes input variables, while the first does not. This version can be applied to reduce the number of states in the state vector [8]. Another beneficial aspect is that the inputs will have a slightly faster impact on the states.

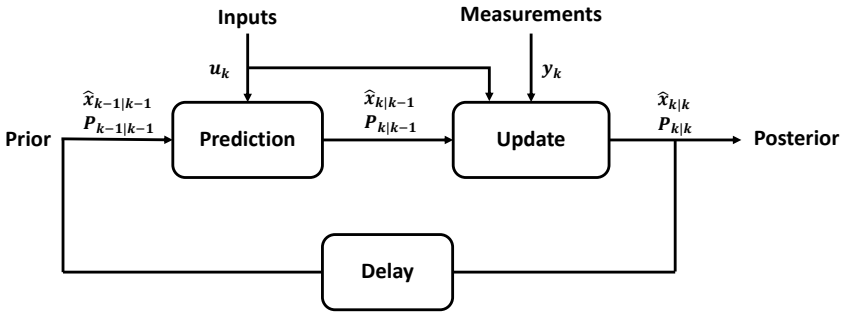


Figure 2.5: Flow chart of the computations in the EKF algorithm with input variables.

For the prediction step of the EKF with input measurements, the following equations apply,

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k, \mathbf{w}_k), \tag{2.25}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_x \mathbf{P}_{k-1|k-1} \mathbf{F}_x^T + \mathbf{F}_w \mathbf{Q}_{k-1} \mathbf{F}_w^T, \tag{2.26}$$

where $\mathbf{u}_k$ is the input variable vector, $\mathbf{w}_k$ is the Gaussian noise related to the input variables and,

$$\mathbf{F}_x = \left. \frac{\partial \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)}{\partial \mathbf{x}_{k-1}} \right|_{\substack{\mathbf{w}_k = 0 \\ \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}}},$$

$$\mathbf{F}_w = \left. \frac{\partial \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)}{\partial \mathbf{w}_k} \right|_{\substack{\mathbf{w}_k = 0 \\ \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}}}.$$

For the update step, the following equations apply,

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_k, \mathbf{v}_k)), \tag{2.27}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}^T, \tag{2.28}$$

$$\mathbf{S}_k = \mathbf{H}_x \mathbf{P}_{k|k-1} \mathbf{H}_x^T + \mathbf{H}_v \mathbf{R}_k \mathbf{H}_v^T, \tag{2.29}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_x^T \mathbf{S}_k^{-1}, \tag{2.30}$$

where $\mathbf{v}_k$ is the Gaussian noise related to the measurements and,

$$\mathbf{H}_x = \left.\frac{\partial \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k)}{\partial \mathbf{x}_k}\right|_{\substack{\mathbf{v}_k=0 \\ \mathbf{x}_k=\hat{\mathbf{x}}_{k|k-1}}},$$

$$\mathbf{H}_v = \left.\frac{\partial \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k)}{\partial \mathbf{v}_k}\right|_{\substack{\mathbf{v}_k=0 \\ \mathbf{x}_k=\hat{\mathbf{x}}_{k|k-1}}}.$$

This concludes the theory of state estimation with the extended Kalman filter.

3

Method - Implementation and Evaluation of Navigation Methods

The approach of the first part of the project is presented in this chapter. First, the development of the different navigation methods is described and summarized in pseudo code algorithms for basic understanding. Afterwards, the datasets that were created and used is presented. The chapter ends with a section about how the evaluation of the methods has been done.

3.1 Navigation methods

One of the main parts of the project was to evaluate four different navigation methods. In this study, the focus was on estimating the position and heading of the front of the vehicle. For all navigation methods, the position is given in ENU coordinates (for the relation between GPS and ENU coordinates, see Section 2.5) and the heading is zero degrees when the vehicle is pointing in east direction. Furthermore, a rotation counterclockwise from east results in a positive heading angle. In this section, the development of the different methods is presented.

3.1.1 Navigation method 1 - GNSS Previous point

To estimate the position of the front of the vehicle, at time k, the displacement of the mounting point of the GNSS unit, $(x_{gnss,k}, y_{gnss,k})$, relative to the position of the front, $(x_{1,k}, y_{1,k})$, needs to be considered. The GNSS unit is mounted at an Euclidean distance of 1.01 m from the front. Thus, the estimated position of the front of the vehicle can be obtained via

$$x_{1,k} = x_{gnss,k} + 1.01\cos(\theta_k), \tag{3.1}$$

$$y_{1,k} = y_{gnss,k} + 1.01\sin(\theta_k), \tag{3.2}$$

where θ_k is the heading.

A simple way of estimating the heading, θ_k, is to utilize position measurements from the previous and current time instant. Given knowledge about the position in both the previous time step, $k-1$, and current time step, k, the heading, θ_k, can be estimated through trigonometry based on the two points,

$$\theta_k = \text{atan2}((y_{gnss,k} - y_{gnss,k-1}),\ (x_{gnss,k} - x_{gnss,k-1})). \tag{3.3}$$

This approach of estimating the heading will from here on be referred to as *Previous point*. An illustration of the method is given in Fig. 3.1.

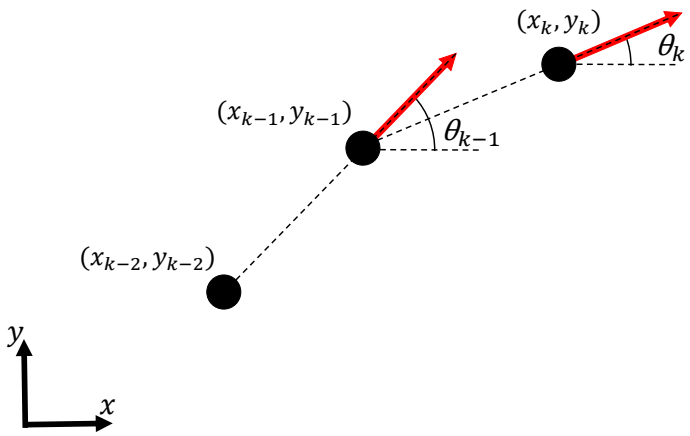


Figure 3.1: To estimate the current heading angle, θ_k, using the Previous point method, information about the position at current time, (x_k, y_k), and the previous, (x_{k-1}, y_{k-1}), is required.

If the route that the vehicle drives is almost constant and without sharp turns, this method could give an acceptable result. However, by applying this simple method, some critical drawbacks need to be considered. As seen in Fig. 3.2, when entering the curve and estimating the heading at time $k-1$, the heading estimate, θ_{k-1}, is delayed in relation to the true heading, $\theta_{t,k-1}$. This raises an estimation error, $\theta_{e,k-1}$. Further on, when the vehicle has proceeded into the curve at time k, the steering angle has increased, causing a larger estimation error, $\theta_{e,k}$. The estimated heading will diverge a lot from the true heading as the method, by itself, does not consider the vehicle dynamics. Consequently, this also affects the position estimate, as the heading angle is used to calculate the position of the front relative to the GNSS mounting point. Therefore, a compensation angle is added to the estimated heading. The heading compensation angle is derived using the vehicle dynamics in Eq. (2.18), but formulated with the infinitesimal increment of the heading

$$d\theta = -T\frac{v_k \sin(\phi_k) + \ell_2 \dot{\phi}_k}{\ell_2 + \ell_1 \cos(\phi_k)}. \tag{3.4}$$

The speed, v_k, is calculated as the Euclidean distance between the current and previous position divided by the sampling time T. The steering angle rate, $\dot{\phi}_k$, was calculated as the difference between two consecutive steering angle measurements, divided by T. Since the sharpness of a turn is correlated to the steering angle of the vehicle, the error of the estimated pose increases the more the steering angle changes between two time steps. Summing Eq. (3.3) and Eq. (3.4) results in the final expression for the heading estimate, given by

$$\theta_k = \text{atan2}((y_{gnss,k} - y_{gnss,k-1}),\ (x_{gnss,k} - x_{gnss,k-1})) - T\frac{v_k \sin(\phi_k) + \ell_2 \dot{\phi}_k}{\ell_2 + \ell_1 \cos(\phi_k)}. \tag{3.5}$$

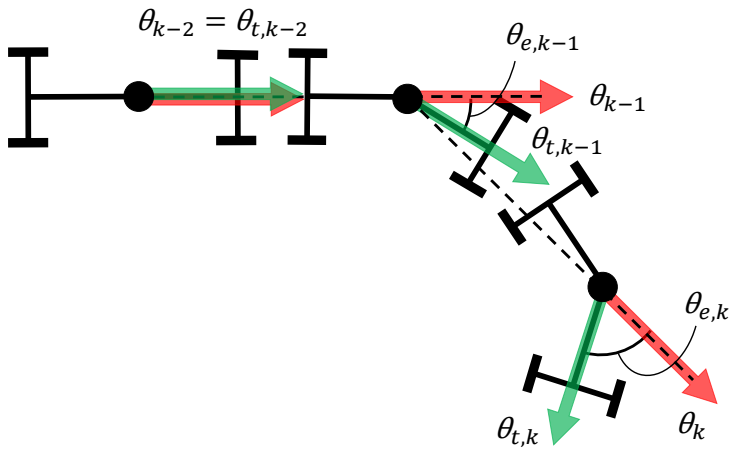


Figure 3.2: An exaggerated example visualizing the odometry of the vehicle and how the heading estimate error arises during sharp turns.

In summary, the GNSS Previous point navigation method is explained in pseudo code in Algorithm 1.

Algorithm 1: Navigation method 1 - GNSS Previous point

```
Result: Pose estimation using the GNSS Previous point navigation method
initialize front position;
while GNSS, steering angle and steering angle rate data are accessible do
    get current GPS coordinates;
    calculate current sensor position in ENU coordinates;
    get current steering angle;
    get current steering angle rate;
    calculate current heading using Previous point;
    if abs(steering angle) > 0.0 then
        calculate the compensation angle;
        adjust current heading with compensation angle;
    end
    calculate current front position using sensor position and heading;
    update the previous position with the current;
end
```

3.1.2 Navigation method 2 - EKF 1 GNSS

One commonly applied method to fuse multiple sensor data in a nonlinear system (which the articulated vehicle is) is to use an EKF [28]. For this method, measurements from one GNSS unit and a steering angle sensor was used. The theory behind the EKF is explained in Section 2.7.1. As was mentioned in that section, the algorithm requires a state vector, motion model and a measurement

model to compute the posterior, which in this case is the final pose estimate. The fundamentals of the navigation method are explained in pseudo code in Algorithm 2 and the state vector, as well as how the motion and measurement models were derived, will be presented in the two following sections.

Algorithm 2: Navigation method 2 - EKF 1 GNSS

Result: Pose estimation with EKF, given data from one GNSS and a steering angle sensor.

initialize a prior of the state vector and its covariance matrix;

while *GNSS, steering angle and steering angle rate data are accessible* **do**

- get current GPS coordinates;
- calculate current sensor position in ENU coordinates;
- get current steering angle;
- get current steering angle rate;
- For the EKF prediction step do:
  1. predict state vector, given the prior and the motion model;
  2. predict state covariance matrix;
- For the EKF update step do:
  1. update predicted state vector, given the measurement model and measurements
  2. update predicted state covariance matrix
- get the current front position and heading from the posterior state vector;
- update the prior with the posterior;

end

3.1.2.1 State vector and motion model

In Section 2.6, the system dynamics of the vehicle, i.e. its motion, were modelled. The motion model of the system is equivalent to the vehicle dynamics equations represented in matrix form. In this case, the differential equations of the vehicle dynamics are nonlinear, which implies that the motion model is required to be expressed on a nonlinear form, according to

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{q}(t), \tag{3.6}$$

where $\mathbf{f}(\mathbf{x}(t))$ is a nonlinear function and $\mathbf{q}(t)$ is motion noise, which is assumed to be a white Gaussian noise (WGN) process, $\mathcal{N}(\mathbf{0}, \mathbf{Q})$. The matrix $\mathbf{Q}$ is the covariance matrix that corresponds to the distribution of the process noise. The state vector $\mathbf{x}(t)$ was chosen as

$$\mathbf{x}(t) = [x_1(t),\ y_1(t),\ v(t),\ \theta_1(t),\ \phi(t),\ \dot{\phi}(t)]^T,$$

where $(x_1(t), y_1(t))$ is the position of the front, $v(t)$ is the speed, $\theta_1(t)$ is the heading of the front and $\phi(t), \dot{\phi}(t)$ is the steering angle and steering angle rate respectively.

Another way to choose the state vector is to consider steering angle and steering angle rate as inputs to the system instead, which would reduce the state vector. However, the complexity of having them in the state vector is not large and it was believed that the pose estimate would be more accurate if steering angle and steering angle rate first was predicted and then updated with measurements as that will smooth out the estimate and not be that reactive to the sensor measurements that fluctuated.

Four of the state derivatives of $\dot{x}_i(t)$ $(i = 1, ..., 6)$, namely the derivatives of position in x and y, heading and steering angle are defined in Section 2.6 by Eq. (2.8), Eq. (2.9), Eq. (2.17) and Eq. (2.18). The remaining state derivatives are the linear acceleration of the front body and the acceleration of the steering angle. Since the vehicle in general is moving at low and relatively constant speed, it is assumed that the acceleration is zero. Thus, the linear acceleration and steering angle acceleration can be described by a so called *constant velocity model* [29]. This means that all accelerations are assumed to be zero and to relax the constraint, process noise is added to those states. Consequently, the continuous-time motion model can be formulated as

$$\underbrace{\begin{bmatrix} \dot{x}_1(t) \\ \dot{y}_1(t) \\ \dot{v}(t) \\ \dot{\theta}_1(t) \\ \dot{\phi}(t) \\ \ddot{\phi}(t) \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} v(t)\cos(\theta_1(t)) \\ v(t)\sin(\theta_1(t)) \\ 0 \\ -\frac{v(t)\sin(\phi(t))+\ell_2\dot{\phi}(t)}{\ell_2+\ell_1\cos(\phi(t))} \\ \dot{\phi}(t) \\ 0 \end{bmatrix}}_{\mathbf{f}(\mathbf{x}(t))} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{q}(t)} \underbrace{\begin{bmatrix} q_c^v(t) \\ q_c^\phi(t) \end{bmatrix}}_{\mathbf{q}_c(t)}, \tag{3.7}$$

where $\mathbf{q}_c(t)$ is the added noise due to the assumptions that all accelerations are zero. The superscripts of the elements indicate which state the noise concern.

As the sensor data is obtained in discrete time, all filter components also need to be modelled in discrete time. Therefore, the continuous-time motion model Eq. (3.7) was discretized. One commonly applied method for discretization is the *Euler discretization method* [30]. The discrete-time motion model obtained by applying Euler descretization is given by

$$\underbrace{\begin{bmatrix} x_{1,k} \\ y_{1,k} \\ v_k \\ \theta_{1,k} \\ \phi_k \\ \Phi_k \end{bmatrix}}_{\mathbf{x}_k} = \underbrace{\begin{bmatrix} x_{1,k-1} + Tv_{k-1}\cos(\theta_{1,k-1}) \\ y_{1,k-1} + Tv_{k-1}\sin(\theta_{1,k-1}) \\ v_{k-1} \\ \theta_{1,k-1} - T\frac{v_{k-1}\sin(\phi_{k-1})+\ell_2\Phi_{k-1}}{\ell_2+\ell_1\cos(\phi_{k-1})} \\ \phi_{k-1} + T\Phi_{k-1} \\ \Phi_{k-1} \end{bmatrix}}_{\mathbf{f}(\mathbf{x}_{k-1})} + \mathbf{q}_{k-1}, \tag{3.8}$$

where Φ_k is the discrete form of $\dot{\phi}(t)$, T is the sampling time and $\mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ is the discrete-time process noise, where $\mathbf{Q}$ is the covariance matrix that represents the distribution of the process noise. To be able to compute the predicted covariance

matrix in the EKF equations, the Jacobian matrix, $\mathbf{F}(\mathbf{x}_{k-1})$, was derived. This was done by differentiating each equation in $\mathbf{f}(\mathbf{x}_{k-1})$ with respect to each state in $\mathbf{x}_{k-1}$, which results in

$$\mathbf{F}(\mathbf{x}_{k-1}) = \begin{bmatrix} 1 & 0 & \frac{\delta f_1}{\delta v_{k-1}} & \frac{\delta f_1}{\delta \theta_{1,k-1}} & 0 & 0 \\ 0 & 1 & \frac{\delta f_2}{\delta v_{k-1}} & \frac{\delta f_2}{\delta \theta_{1,k-1}} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{\delta f_4}{\delta v_{k-1}} & 1 & \frac{\delta f_4}{\delta \phi_{k-1}} & \frac{\delta f_4}{\delta \Phi_{k-1}} \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.9}$$

where

$$\frac{\delta f_1}{\delta v_{k-1}} = T\cos(\theta_{1,k-1}), \quad \frac{\delta f_1}{\delta \theta_{1,k-1}} = -Tv_{k-1}\sin(\theta_{1,k-1}),$$

$$\frac{\delta f_2}{\delta v_{k-1}} = T\sin(\theta_{1,k-1}), \quad \frac{\delta f_2}{\delta \theta_{1,k-1}} = Tv_{k-1}\cos(\theta_{1,k-1}),$$

$$\frac{\delta f_4}{\delta v_{k-1}} = -\frac{T\sin(\phi_{k-1})}{\ell_2 + \ell_1\cos(\phi_{k-1})},$$

$$\frac{\delta f_4}{\delta \phi_{k-1}} = -\frac{Tv_{k-1}\cos(\phi_{k-1})}{\ell_2 + \ell_1\cos(\phi_{k-1})} - \frac{T\ell_1\sin(\phi_{k-1})(\ell_2\Phi_{k-1} + v_{k-1}\sin(\phi_{k-1}))}{(\ell_2 + \ell_1\cos(\phi_{k-1}))^2},$$

$$\frac{\delta f_4}{\delta \Phi_{k-1}} = -\frac{T\ell_2}{\ell_2 + \ell_1\cos(\phi_{k-1})}.$$

This concludes the derivation of the motion model and everything is now known for the prediction step of the filter.

3.1.2.2 Measurement model

A measurement model describes how measurements, $\mathbf{y}_k$, are related to the state vector $\mathbf{x}_k$. Thus, the measurement model can be expressed as

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k, \tag{3.10}$$

where $\mathbf{h}(\mathbf{x}_k)$ is a nonlinear function and $\mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ is measurement noise, where $\mathbf{R}$ is the covariance matrix that represents the distribution of the measurement noise. In this navigation method, the measurements were the position of the GNSS unit, $(x_{gnss,k}, y_{gnss,k})$, as well as the steering angle, $\phi_{m,k}$, and the steering angle rate, $\Phi_{m,k}$, which gives the measurement vector

$$\mathbf{y}_k = [x_{gnss,k},\ y_{gnss,k},\ \phi_{m,k},\ \Phi_{m,k}]^T.$$

Since the states $x_{1,k}$ and $y_{1,k}$ corresponds to the position of the front body and not the position of the GNSS unit, which is located $\ell_{gnss} = 1.01$ m from the front body towards the joint, the measurement model gets the following expression

$$\underbrace{\begin{bmatrix} x_{gnss,k} \\ y_{gnss,k} \\ \phi_{m,k} \\ \Phi_{m,k} \end{bmatrix}}_{\mathbf{y}_k} = \underbrace{\begin{bmatrix} x_{1,k} - \ell_{gnss}\cos(\theta_{1,k}) \\ y_{1,k} - \ell_{gnss}\sin(\theta_{1,k}) \\ \phi_k \\ \Phi_k \end{bmatrix}}_{\mathbf{h}(\mathbf{x}_k)} + \mathbf{r}_k. \tag{3.11}$$

To compute the updated state vector and covariance matrix in the EKF equations, the Jacobian matrix, $\mathbf{H}(\mathbf{x}_k)$, was derived. This was done by differentiating each equation in $\mathbf{h}(\mathbf{x}_k)$ with respect to each state in $\mathbf{x}_k$, which gives

$$\mathbf{H}(\mathbf{x}_k) = \begin{bmatrix} 1 & 0 & 0 & \ell_{gnss}\sin(\theta_{1,k}) & 0 & 0 \\ 0 & 1 & 0 & -\ell_{gnss}\cos(\theta_{1,k}) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.12}$$

With the derivations of the state vector, motion model and now also the measurement model, everything is in place in order to perform all the steps explained in Algorithm 2.

3.1.3 Navigation method 3 - EKF 2 GNSS

The third navigation method that was implemented and evaluated was to fuse information from two GNSS units and the steering angle sensor via an EKF. It is very similar to the method presented in Section 3.1.2, and therefore this method incorporates the structure of Algorithm 2, with the exception that it utilizes information from two GNSS units. Another similarity is that the same state vector and motion model is applicable. However, because of the additional GNSS unit, a different measurement model is required for the update step of the filter. The motivation behind evaluating this method is to investigate if the accuracy increases when utilizing information from more than one GNSS units. The state vector and motion model that was used is explained in Section 3.1.2.1, and in the following section, the measurement model is derived.

3.1.3.1 Measurement model

Since this method utilizes measurements from two GNSS units and the steering angle sensor, the measurement vector becomes

$$\mathbf{y}_k = [x_{gnss\,1,k},\ y_{gnss\,1,k},\ x_{gnss\,2,k},\ y_{gnss\,2,k},\ \phi_{m,k},\ \Phi_{m,k}]^T.$$

To obtain a realistic scenario, the two GNSS units were modelled to be mounted 1.01 m down from the front body and 0.6 m between each other, along the axis that is normal to the front body. A conceptual illustration can be seen in Fig. 3.3.

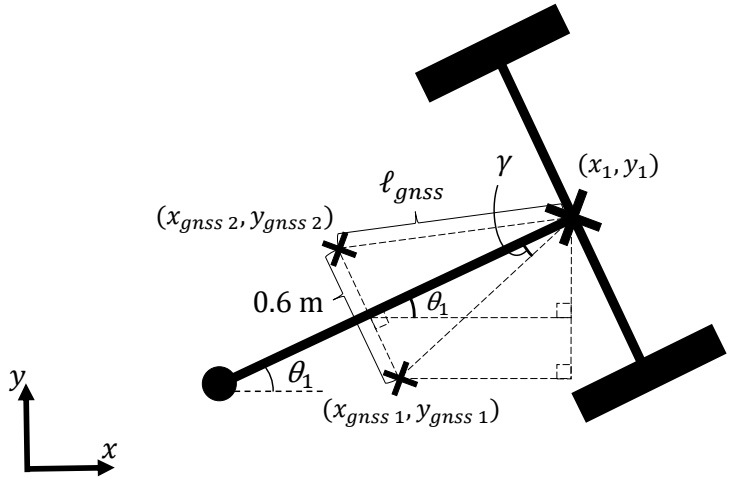


Figure 3.3: Derivation of the position of two GNSS devices relative to the front of the vehicle.

With the use of trigonometry, the Euclidean distance from the front position to each GNSS unit, ℓ_{gnss}, can be calculated as

$$\ell_{gnss} = \sqrt{1.01^2 + (0.6/2)^2}. \tag{3.13}$$

Furthermore, the angle γ shown in Fig. 3.3 is given by

$$\gamma = \arctan\left(\frac{0.6/2}{1.01}\right). \tag{3.14}$$

With these constants known, the measurement model, that relates the measurements to the states, is formulated as

$$\underbrace{\begin{bmatrix} x_{gnss\,1,k} \\ y_{gnss\,1,k} \\ x_{gnss\,2,k} \\ y_{gnss\,2,k} \\ \phi_k \\ \Phi_k \end{bmatrix}}_{\mathbf{y}_k} = \underbrace{\begin{bmatrix} x_{1,k} - \ell_{gnss}\sin(\frac{\pi}{2} - \gamma - \theta_{1,k}) \\ y_{1,k} - \ell_{gnss}\cos(\frac{\pi}{2} - \gamma - \theta_{1,k}) \\ x_{1,k} - \ell_{gnss}\sin(\frac{\pi}{2} - \gamma - \theta_{1,k}) - 0.6\sin(\theta_{1,k}) \\ y_{1,k} - \ell_{gnss}\cos(\frac{\pi}{2} - \gamma - \theta_{1,k}) + 0.6\cos(\theta_{1,k}) \\ \phi_k \\ \Phi_k \end{bmatrix}}_{\mathbf{h}(\mathbf{x}_k)} + \mathbf{r}_k, \tag{3.15}$$

where $\mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ and $\mathbf{R}$ is the covariance matrix that represents the distribution of the measurement noise.

Like in the previous method, $\mathbf{H}(\mathbf{x}_k)$, which is the Jacobian of $\mathbf{h}(\mathbf{x}_k)$ was derived and is defined as

$$\mathbf{H}(\mathbf{x}_k) = \begin{bmatrix} 1 & 0 & 0 & \ell_{gnss}\cos(\gamma + \theta_{1,k} - \frac{\pi}{2}) & 0 & 0 \\ 0 & 1 & 0 & \ell_{gnss}\sin(\gamma + \theta_{1,k} - \frac{\pi}{2}) & 0 & 0 \\ 1 & 0 & 0 & \ell_{gnss}\cos(\gamma + \theta_{1,k} - \frac{\pi}{2}) - 0.6\cos(\theta_{1,k}) & 0 & 0 \\ 0 & 1 & 0 & \ell_{gnss}\sin(\gamma + \theta_{1,k} - \frac{\pi}{2}) - 0.6\sin(\theta_{1,k}) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.16}$$

This concludes the derivations of the components used in the third method.

3.1.4 Navigation method 4 - EKF GNSS-IMU-mag

This method is based on the approach suggested by M. Kok, J. D. Hol, and T. B. Schön [8], to fuse data from inertial measurements (accelerometer and gyroscope) with additional sensors, such as GPS and magnetometer, in order to estimate the 3D pose of a vehicle. The information provided by the sensors were fused using an EKF with input variables (see the theory in Section 2.7.2). The chosen input variables were the measurements from the accelerometer and gyroscope. One reason for using input variables is because it enables that the state vector can be reduced. The state vector, motion model and measurement model, are all derived differently from previous EKF based methods. That is partly because of the input variables, but also because the heading estimate from IMU and magnetometer data is calculated with a quaternion representation. When utilizing IMU and magnetometer data it is convenient to use unit quaternions, since they are given in a compact representation and are not ambiguous [31, 32]. Also, because the usage of quaternions, the pose estimate is calculated in 3D but as the other two methods are in 2D, the final results from this method are also presented in 2D.

The navigation method is explained at a conceptual level in Algorithm 3 and the detailed derivations of the state vector, motion model and measurement model are explained in the two following sections.

Algorithm 3: Navigation method 4 - EKF GNSS-IMU-mag

Result: Pose estimation with EKF GNSS-IMU-mag
initialize a prior of the state vector and its covariance matrix;
while *GNSS, IMU and magnetometer data are accessible* **do**
  get current GPS coordinates;
  calculate current sensor position in ENU coordinates;
  get current IMU data and set as input;
  get current magnetometer data;
  For the EKF prediction step do:
    1. predict state vector, given the prior, the input and the motion model;
    2. predict state covariance matrix;
  For the EKF update step do:
    1. update predicted state vector, given the measurement model and measurements;
    2. update predicted state covariance matrix;
    3. normalize the state vector and its covariance matrix;
  calculate the current heading by transforming the posterior quaternion to Euler angles and extract the yaw angle;
  calculate the current front position using the posterior position and the current heading;
  update the prior with the posterior;
end

3.1.4.1 State vector and motion model

For this method, the state vector was chosen as

$$\mathbf{x}(t) = [\mathbf{p}(t),\ \mathbf{v}(t),\ \tilde{\mathbf{q}}^T(t)]^T,$$

where $\mathbf{p}(t)$ is the 3D position of the GNSS unit, $\mathbf{v}(t)$ is the 3D velocity and $\tilde{\mathbf{q}}(t)$ is the unit quaternion that represents the orientation. The reason that the estimated position is the GNSS position and not the front (as in the other two EKF methods), is to reduce complexity in the measurement model, which will be explained more explicitly in the next subsection.

The continuous-time relation between position, velocity and acceleration was used to model the predicted position and is given by

$$\mathbf{v}(t) = \frac{d\mathbf{p}(t)}{dt}, \quad \mathbf{a}(t) = \frac{d\mathbf{v}(t)}{dt}. \tag{3.17}$$

Furthermore, the relation between the orientation and the angular velocity, $\boldsymbol{\omega}^s(t)$ (the superscript s indicates that it is with respect to the sensor frame), is used for the heading estimate and is given by

$$\frac{d\tilde{\mathbf{q}}(t)}{dt} = \tilde{\mathbf{q}}(t) \odot \frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega}^s(t) \end{bmatrix}. \tag{3.18}$$

The $\odot$ symbol denotes the quaternion product operator (see definition in Appendix A.2).

Using Euler discretization on Eq. (3.17) and Eq. (3.18), while assuming that the acceleration is constant between each sample, results in

$$\mathbf{p}_k = \mathbf{p}_{k-1} + T\mathbf{v}_{k-1} + \frac{T^2}{2}\mathbf{a}_{k-1}, \tag{3.19}$$

$$\mathbf{v}_k = \mathbf{v}_{k-1} + T\mathbf{a}_{k-1}, \tag{3.20}$$

$$\tilde{\mathbf{q}}_k = \tilde{\mathbf{q}}_{k-1} \odot \exp_q\left(\frac{T}{2}\boldsymbol{\omega}^s_{k-1}\right), \tag{3.21}$$

where $\exp_q(.)$ is the quaternion exponential operator (see definition in Appendix A.3). The quaternion exponential is used to convert the orientation of the gyroscope relative to its starting point into quaternion representation. Using the equation for accelerometer measurements, Eq. (2.3), the linear acceleration, $\mathbf{a}_{k-1}$, can be substituted. Thus, Eq. (3.19) and Eq. (3.20) can be formulated as

$$\mathbf{p}_k = \mathbf{p}_{k-1} + T\mathbf{v}_{k-1} + \frac{T^2}{2}(\mathbf{R}^{w/s}_{k-1}(\mathbf{u}_{a,k-1} - \boldsymbol{\delta}_{a,k-1}) + \mathbf{g} + \mathbf{e}_{a,k-1}), \tag{3.22}$$

$$\mathbf{v}_k = \mathbf{v}_{k-1} + T(\mathbf{R}^{w/s}_{k-1}(\mathbf{u}_{a,k-1} - \boldsymbol{\delta}_{a,k-1}) + \mathbf{g} + \mathbf{e}_{a,k-1}). \tag{3.23}$$

The rotation matrix $\mathbf{R}^{w/s}_{k-1}$ rotates the sensor frame, s, to the world frame, w (where it is of interest to know the pose), which is indicated by the superscript. Since the orientation state is represented by quaternions, $\mathbf{R}^{w/s}_{k-1}$ is computed by conversion of the quaternions at each time step (see definition in Appendix A.4). Furthermore, $\mathbf{g}$ is the acceleration due to Earth's gravity and $\mathbf{u}_{a,k-1}$ is the accelerometer measurement, which is an input variable. The sensor bias of the accelerometer is given by $\boldsymbol{\delta}_{a,k-1}$ and $\mathbf{e}_{a,k-1}$ is the sensor noise.

Like the substitution of the linear acceleration in Eq. (3.22) and Eq. (3.23), the equation for gyroscope measurements, Eq. (2.4), can be used to substitute the angular velocity in Eq. (3.21). Therefore, the orientation state can be formulated as

$$\tilde{\mathbf{q}}_k = \tilde{\mathbf{q}}_{k-1} \odot \exp_q\left(\frac{T}{2}(\mathbf{u}_{\omega,k-1} - \boldsymbol{\delta}_{\omega,k-1} - \mathbf{e}_{\omega,k-1})\right), \tag{3.24}$$

where $\mathbf{u}_{\omega,k-1}$ is the gyroscope measurement, which is an input variable. The sensor bias of the gyroscope is given by $\boldsymbol{\delta}_{\omega,k-1}$ and $\mathbf{e}_{\omega,k-1}$ denotes the sensor noise.

Finally, the motion model is composed of Eq. (3.22) - Eq. (3.24). Also, the accelerometer and gyroscope are assumed to be calibrated, which implies that $\boldsymbol{\delta}_{a,k-1}$ and $\boldsymbol{\delta}_{\omega,k-1}$ are zero. Therefore, the motion model results in

$$\underbrace{\begin{bmatrix}\mathbf{p}_k\\ \mathbf{v}_k\\ \tilde{\mathbf{q}}_k\end{bmatrix}}_{\mathbf{x}_k} = \underbrace{\begin{bmatrix}\mathbf{p}_{k-1} + T\mathbf{v}_{k-1} + \frac{T^2}{2}(\mathbf{R}^{w/s}_{k-1}\mathbf{u}_{a,k-1} + \mathbf{g} + \mathbf{e}_{p,a,k-1})\\ \mathbf{v}_{k-1} + T(\mathbf{R}^{w/s}_{k-1}\mathbf{u}_{a,k-1} + \mathbf{g} + \mathbf{e}_{v,a,k-1})\\ \tilde{\mathbf{q}}_{k-1} \odot \exp_q\left(\frac{T}{2}(\mathbf{u}_{\omega,k-1} - \mathbf{e}_{\omega,k-1})\right)\end{bmatrix}}_{\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})}. \tag{3.25}$$

The first subscript of the accelerometer noise indicates which state that the noise concerns. The two noises are unique, however they correspond to the same Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{Q}_a)$. Likewise, the noise of gyroscope is proportional to $\mathcal{N}(\mathbf{0}, \mathbf{Q}_\omega)$. The matrices $\mathbf{Q}_a$ and $\mathbf{Q}_\omega$ represents the covariances related to the process noise.

Furthermore, the Jacobian of $\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})$ with respect to $\mathbf{x}_{k-1}$ is

$$\mathbf{F}_x = \begin{bmatrix} \mathbf{I}_{3x3} & T\mathbf{I}_{3x3} & \frac{T^2}{2}\frac{\partial \mathbf{R}_{k-1}^{w/s}}{\partial \tilde{\mathbf{q}}_{k-1}}\mathbf{u}_{a,k-1} \\ \mathbf{0}_{3x3} & \mathbf{I}_{3x3} & T\frac{\partial \mathbf{R}_{k-1}^{w/s}}{\partial \tilde{\mathbf{q}}_{k-1}}\mathbf{u}_{a,k-1} \\ \mathbf{0}_{4x3} & \mathbf{0}_{4x3} & \left(\exp_q\left(\frac{T}{2}\mathbf{u}_{\omega,k-1}\right)\right)^R \end{bmatrix}. \tag{3.26}$$

The derivative of the rotation matrix, $\mathbf{R}_{k-1}^{w/s}$ with respect to the quaternion, is defined in Appendix A.5. The operator, $(.)^R$, is the right multiplication matrix operator (see definition in Appendix A.6).

Furthermore, the Jacobian of $\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})$ with respect to $\mathbf{w}_{k-1} = [\mathbf{e}_{p,a,k-1},\ \mathbf{e}_{v,a,k-1},\ \mathbf{e}_{\omega,k-1}]^T$ is

$$\mathbf{F}_w = \begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{I}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{4x3} & \mathbf{0}_{4x3} & -\frac{T}{2}(\tilde{\mathbf{q}}_{k-1})^L \frac{\partial \exp_q(\mathbf{e}_{\omega,k-1})}{\partial \mathbf{e}_{\omega,k-1}} \end{bmatrix}. \tag{3.27}$$

The operator $(.)^L$ is the left multiplication matrix operator (see definition in Appendix A.7).

3.1.4.2 Measurement model

The measurement vector contains measurements of the position of the GNSS unit, $\mathbf{p}_{gnss,k}$, and magnetometer, $\mathbf{y}_{m,k}$, and therefore results in

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{p}_{gnss,k} \\ \mathbf{y}_{m,k} \end{bmatrix}. \tag{3.28}$$

Using the equation for magnetometer measurements, Eq. (2.5), the measurement model that relates the measurements to the states can be formulated as

$$\underbrace{\begin{bmatrix} \mathbf{p}_{gnss,k} \\ \mathbf{y}_{m,k} \end{bmatrix}}_{\mathbf{y}_k} = \underbrace{\begin{bmatrix} \mathbf{p}_k + \mathbf{e}_{p,k} \\ \mathbf{R}_k^{s/w}\mathbf{m} + \mathbf{e}_{m,k} \end{bmatrix}}_{\mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)}, \tag{3.29}$$

where $\mathbf{m}$ in this case is the local, horizontal magnetic field. The measurement noises $\mathbf{e}_{p,k}$ and $\mathbf{e}_{m,k}$ are proportional to $\mathcal{N}(\mathbf{0}, \mathbf{R}_p)$, respectively $\mathcal{N}(\mathbf{0}, \mathbf{R}_m)$, where $\mathbf{R}_p$ and $\mathbf{R}_m$ are the covariance matrices that represent the distribution of the measurement noise.

The Jacobian of the measurement model, $\mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)$ with respect to $\mathbf{x}_k$, is consequently

$$\mathbf{H}_x = \begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x4} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \frac{\partial \mathbf{R}_{k-1}^{w/s}}{\partial \tilde{\mathbf{q}}_{k-1}} \mathbf{m} \end{bmatrix}. \tag{3.30}$$

Finally, the Jacobian of the measurement model with respect to $\mathbf{v}_k = [\mathbf{e}_{p,k}\ \mathbf{e}_{m,k}]$, is computed as

$$\mathbf{H}_v = \mathbf{I}_{6x6}. \tag{3.31}$$

The reason why the geometrical relation between the GNSS unit and the front position is not included in the measurement model is because the orientation is in unit quaternions. The quaternions can be converted to Euler angles (where the yaw angle is the heading, θ_k, of the vehicle) and then the front position can be obtain making the same comutation as in the measurement models of the two previous methods. However, calculating the Jacobian of this conversion function is very complex. Therefore the quaternions were converted into Euler angles outside the filter and after the conversion, the front position, $(x_{1,k}, y_{1,k})$, could be calculated using the following equations:

$$x_{1,k} = p_{x,k} + 1.01\cos(\theta_k), \tag{3.32}$$
$$y_{1,k} = p_{y,k} + 1.01\sin(\theta_k). \tag{3.33}$$

3.2 Datasets

To evaluate the performance of the navigation methods relative to each other, it is of great importance that the analysis is done under equal conditions for each method. Throughout the project, logged data collected from the vehicle's sensors have been accessible. Among the data were GNSS, IMU, magnetometer, steering angle and steering angle rate readings from previous test drives. However, when evaluating the navigation methods, a simulated dataset has been also been used, because of two underlying motivations. Firstly, the logged data is noisy. Therefore, it is very challenging to attempt to formulate a ground truth dataset for the pose estimation based only on these measurements. Secondly, although GNSS data is provided from one unit, one of the methods requires readings from two units. Thus, this method needed to be analyzed based on simulated data regardless. Consequently, if the remaining methods were not based on simulated data, all of the navigation methods could not be evaluated relative to each other. However, to get an intuition of how the navigation methods behave with real data, all but the third method were afterward tested using the logged data.

3.2.1 Simulated data

Generating the simulated data consisted of two parts. The first was to generate a ground truth dataset of the states that were to be estimated. This can be viewed as

the actual behaviour of the vehicle. Furthermore, this dataset can be used to analyze how accurate the estimates are, by measuring how much the estimated states diverge from the ground truth states.

A ground truth trajectory of the position of the front of the vehicle was generated with consideration to the dynamics of an articulated vehicle (motion model). The resulting path is seen in Fig. 3.4, as well as the corresponding heading at each time step. The graphs of the remaining ground truth states for the same process; speed, steering angle and steering angle rate, can be seen in Appendix C. However, these are not directly connected to pose estimation and will therefore not be considered further on.

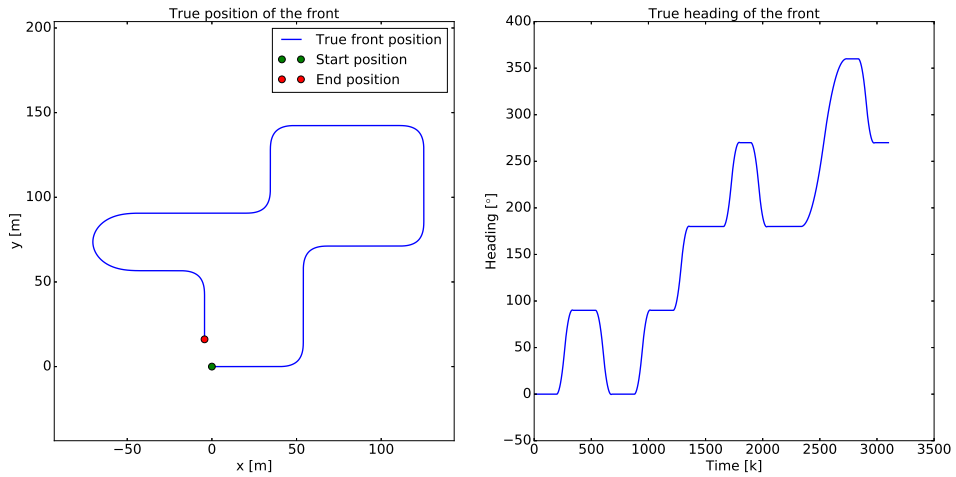


Figure 3.4: A simulated ground truth position trajectory for the front of the vehicle (left) and the corresponding ground truth heading (right).

From the ground truth states, perfect measurements (no noise) could then be generated by using the produced measurement models. To obtain more realistic observations, the perfect measurements were distorted with an additive Gaussian noise, corresponding to the uncertainties of the existing sensors of the vehicle. It is assumed that this is a correct way of modelling the noise under the assumptions that no outer obstacles occur (i.e. multipath) and the path is not very long and hence, not revealing any significant errors correlated over time. In Fig. 3.5, a zoomed in segment of the generated measurements from one, respectively two GNSS units, registered along the path, can be seen. The blue trajectories represent the true positions of the respective GNSS units. Although the simulated measurements are very accurate (near the blue trajectories) it is possible to distinguish a small noise on the position observations, by the distortion of the red markers.

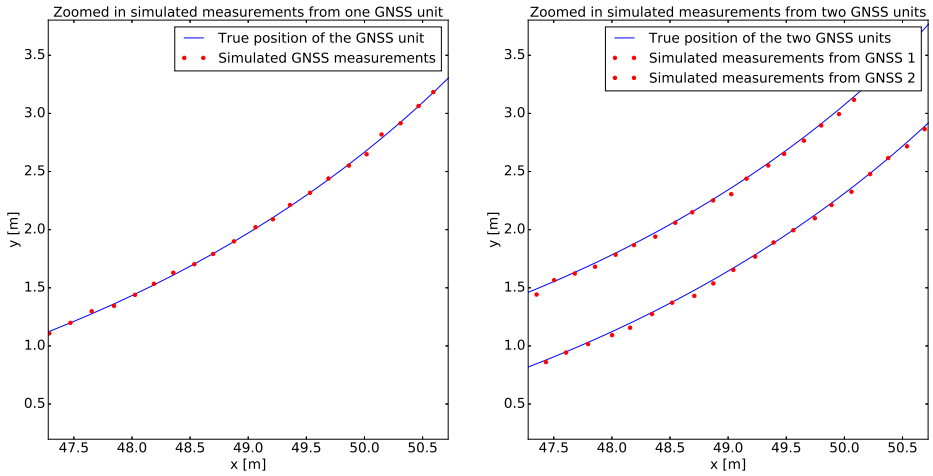


Figure 3.5: Simulated noisy measurements from one (left), respectively two GNSS units (right), sampled from the ground truth trajectory. The blue trajectories represent the true positions of the respective GNSS units.

To simulate measurements from an IMU and a magnetometer along the same path as in Fig. 3.4, the MATLAB toolbox, *imuSensor*, was used [33]. In Fig. 3.6 the simulated measurements from the IMU and the magnetometer can be seen. These measurements have also been modelled to be corrupted by noise corresponding to the uncertainties of the sensors installed on the vehicle. The uncertainties of these sensors were calculated from a recorded sequence of a test drive when the vehicle was not moving. However, the simulation assumes that the vehicle drives under perfect circumstances, i.e. no irregularities appear along the path. Because of this assumption, the measurements are much less exposed to noise than the sensors in a real scenario would experience when the vehicle is moving.

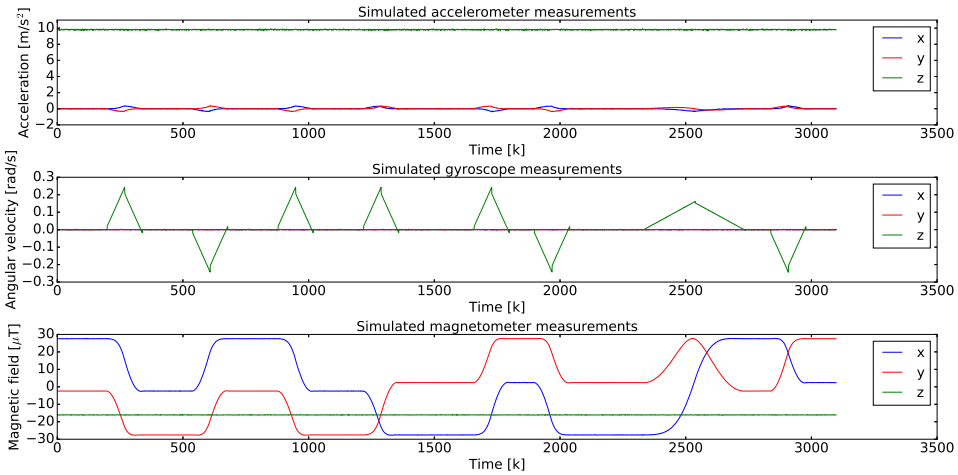


Figure 3.6: Simulated noisy measurements from an accelerometer (upper), a gyroscope (center) and a magnetometer (lower).

3.2.2 Logged data from physical sensors

From a previous test drive, a sequence was recorded. The sequence contains many different quantities measured by the different sensors on the vehicle. The accessible measurements that are relevant to the project are:

- GNSS data (i.e. GPS latitude and longitude)
- Steering angle
- Steering angle rate
- IMU data
  - Accelerometer
  - Gyroscope
- Magnetometer data

To use the GNSS data in the developed navigation method algorithms it first needs to be preprocessed. The reason for this is because the algorithms need the position measurements to be given in a Cartesian coordinate system. The position coordinates that were extracted from the logged dataset were the GPS latitude and longitude. These measurements were then transformed into ENU coordinates, which are seen in Fig. 3.7.

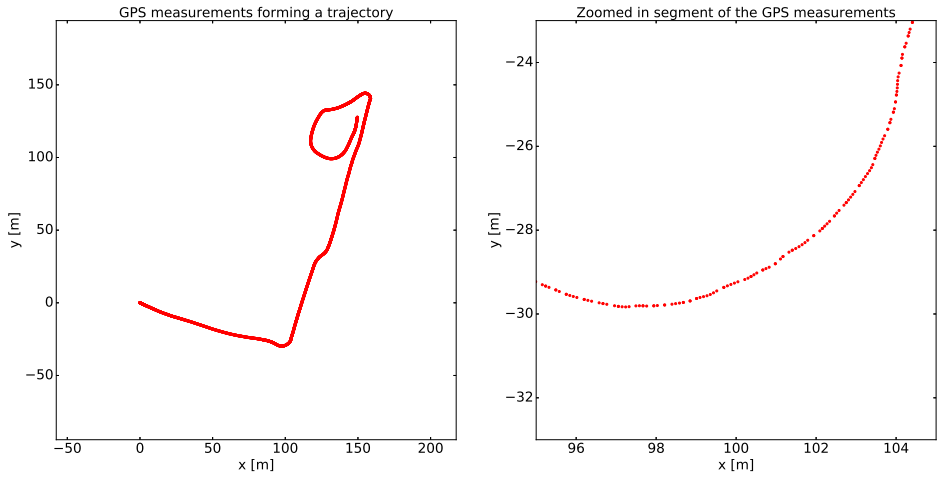


Figure 3.7: ENU coordinates that were transformed from the logged GPS data (left) and a zoomed in segment of the same measurements (right).

3.3 Evaluation of navigation methods

First, all the methods were tested, using the simulated dataset. The resulting position and heading estimates could then be plotted together with the ground truth data. Based on these plots a visual analysis could be made by observing how the pose estimates behave in relation to the ground truth data. Then the absolute estimation error was calculated at each time step, as the absolute value

of the estimate minus the ground truth. Furthermore, an RMSE value for position and heading was calculated for each method. The calculated value is used to check how much the estimate deviates from the ground truth. Hence, the lower value, the better. The RMSE value for position is given by

$$RMSE_{pos} = \sqrt{\frac{\sum_{k=0}^{K}(\hat{x}_k - x_k)^2 + (\hat{y}_k - y_k)^2}{K}}, \tag{3.34}$$

where $\hat{x}_k$ and $\hat{y}_k$ are the estimated position components, while x_k and y_k are the corresponding ground truth values and K is the total number of samples [34]. When calculating the RMSE value for heading,

$$RMSE_{head} = \sqrt{\frac{\sum_{k=0}^{K}(\hat{\theta}_k - \theta_k)^2}{K}} \tag{3.35}$$

applies. The estimated heading is denoted $\hat{\theta}_k$ and θ_k is the ground truth heading.

Finally, the logged dataset was used to check whether the methods worked in a real scenario or not. This was done by plotting the results of all methods, except the one with two GNSS devices as all required data was not included in the dataset, in the same figure and checked if they were following the same trend. As mentioned in Section 3.2, it is a very complex procedure to derive reliable ground truth values from the logged data as it consists of noisy measurements. Hence, the same aspects can not be analyzed as for the simulated test and no RMSE value can be calculated. One common way to evaluate the performance of a Kalman filter, when no ground truth of the state variables is available, is to check the filter innovation [35]. A correctly functional filter should have an innovation that is uncorrelated and zero mean over time. This was visually checked by plotting the innovation.

In summary, to compare and evaluate the four different navigation methods, the following methodology was applied:

1. Simulations:
   - Run each navigation method, using the simulated dataset, and compare the resulting position and heading estimates with the ground truth data.
   - For each method, calculate the absolute estimation error for position and heading at each time step, and calculate their respective RMSE value.
2. Real scenario experiments:
   - Run all navigation methods, except method 3, using the logged dataset. Compare the resulting trends of each navigation method to each other.
   - Analyze the behaviour of the filter properties of navigation method 2 and 4.
3. Evaluate the navigation methods relative to each other based on the simulations and the real case experiments.

4

Method - Urban Simulation

As mentioned in Section 1.4, the urban simulation is a tool for early testing and evaluation of different sensor setups used in an urban environment under different weather conditions. The simulation is also used for investigating the demands of the operational environment. These demands are related to the so-called *reference points* that local sensors are observing. The local sensors which have been investigated are radar, camera and ultrasonic sensor. The simulation is in a conceptual stage of the development. This means that it delivers the desired results and works as intended, but because of simplifications and assumptions, further development would provide more reliable results.

This chapter is initiated with a section overviewing the simulation and it is focused on the whole picture for a more comprehensible introduction. Thereafter, other sections will describe some of the features of the simulation in more detail. When the different parts of the simulation have been described, a section concerning how the different sensor setups were chosen is followed. The chapter ends with a section about how the sensor setups, demands on the operational environment and weather impact are evaluated.

4.1 Overview of developed simulation for navigation in urban environment

As mentioned earlier, in Section 1.6, the simulated urban environment is very simplified due to time limitation and the complexity of simulating an "real world" scenario. Hence, the simulated urban environment is based on the same dataset that was used to evaluate the navigation methods (see Section 3.2.1). The path is constructed to symbolize a realistic route that a vehicle is driving in a city, with a lot of 90° turns, making it look like the vehicle is driving in a block. Furthermore, to illustrate the city environment even more, the reference points, that the local sensors are observing, are placed at a distance of 25 m between each other, evenly spread along the road. This is the same distance as the Swedish Road Administration is recommending for distances between lampposts in a city environment [36]. This distance can also be changed to put the reference points closer to each other. The simulated environment, together with the reference points can be seen in Fig. 4.1.

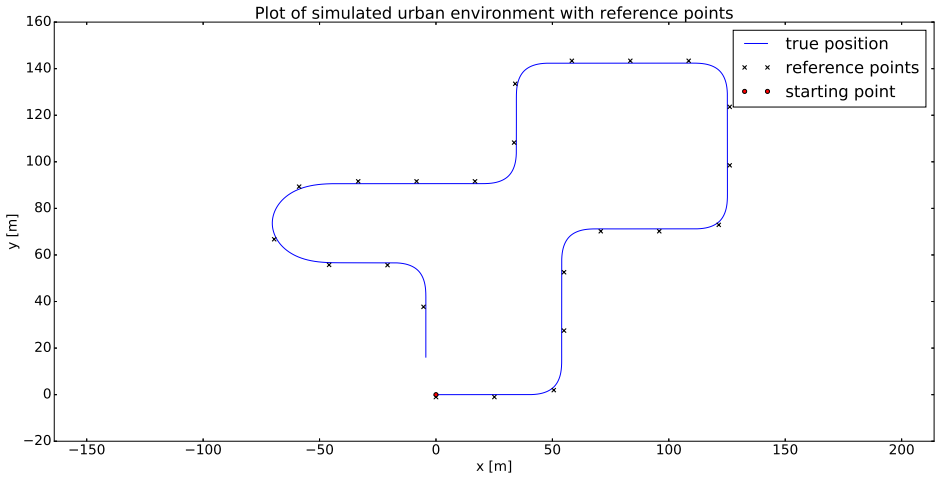


Figure 4.1: Figure of the simulated urban environment together with the reference points that are located at a distance of 25 m between each other, evenly spread along the road. The reference points are used to localize the vehicle in the environment.

The just mentioned reference points are a vital part of the simulation. As the position measurements from the GNSS were considered to be unreliable in an urban environment, it is further on assumed that no GNSS data is available. Therefore, local sensors are used instead. These local sensors are observing reference points in the operational environment. The reference points can for example be viewed as perfect sensor reflectors for a radar. This means that, when a sensor is observing a reference point, a position measurement is generated. This is a simplification as the raw sensor data is not being considered. In other words, the preprocessing of going from raw sensor data to a position measurement is not considered. Instead, when a sensor is observing a reference point, a position measurement is obtained. These measurements are assumed to be corrupted by white Gaussian noise and the accuracy of the measurements are proportional to the sensor specific position accuracy found in the literature [37, 38, 39]. How the sensors are observing the reference points can in short be described by that every sensor has a field of view (FoV), if the reference point is inside the field, the sensor is observing it.

As the goal of the simulation is to investigate navigation in an urban environment, the final result is a pose estimate. To generate this pose estimate, a modified version of the EKF 1 GNSS, explained in Section 3.1.2, was developed. In the filter used for this simulation, the same state vector and motion model was used as for EKF 1 GNSS. However, as no GNSS data is available, the position measurements are generated by utilizing other sensors. Furthermore, the difference in the implementation lies in the update step of the filter. As a measurement is obtained only if a sensor is observing a reference point, this means that the number of measurements will vary between the time instances. Hence, this was dealt with by having an adaptive measurement model. Meaning that, instead of having a fixed dimension of the measurement model, it was changed in every iteration in order to deal with the constantly changing amount of measurements. In summary, the

simulation is presented in pseudo code in Algorithm 4.

Algorithm 4: Pose estimation with EKF in urban environment

Result: Pose estimation with EKF in urban environment
Initialization of inputs:
1. generate ground truth states
2. define weather condition
3. define sensor types and amount
4. define mounting angle for every sensor
5. define distance between reference points
6. given the distance, generate reference points
7. initialize prior for state vector, **x**, and covariance matrices P and Q

for *every time step, k* **do**
  for *every sensor and reference point* **do**
    if *reference point inside FoV* **then**
      generate measurement;
    end
  end
  For the EKF prediction step do:
    1. predict state vector, given prior and motion model;
    2. predict state covariance matrix;
  For the EKF update step do:
    1. generate measurement model based on number of generated measurements;
    2. update predicted state vector, given measurements and generated measurement model;
    3. update predicted state covariance matrix;
  update the prior with the posterior;
end

4.2 Inputs to the system

To make the simulation more user-friendly a couple of settings are given to the system via input. Making it possible for the user to choose a couple of variables of interest. The changeable variables are summarized in the list below:

- number of sensors (unlimited)
- type of sensors (radar, camera or ultrasonic sensor)
- mounting angle for the sensors, making it possible to have a 360° view of the surroundings.
- weather condition (clear, precipitation or dark)
- distance between reference points

As the simulation has this amount of changeable variables, it gives an enormous amount of different test cases. Which in turn brings value to the simulation as a lot of different tests can be made.

4.3 Modelling of sensors and their impact of weather

It is assumed that all sensors are mounted on the same place in the front of the vehicle. This is not possible in a real world application but the simplification was done to minimize geometrical computations. Therefore, how the sensors are mounted on the vehicle is only modelled by an angle and the sensors can point in any direction. Furthermore, the different sensor types were modelled by defining how wide and far away they can observe an object. This was done by constructing a specific field of view for every sensor type, that is defined by a radius and an angle, creating a circle segment. In Fig. 4.2 is an illustrative example of what the constructed field of views look like for the different sensors.

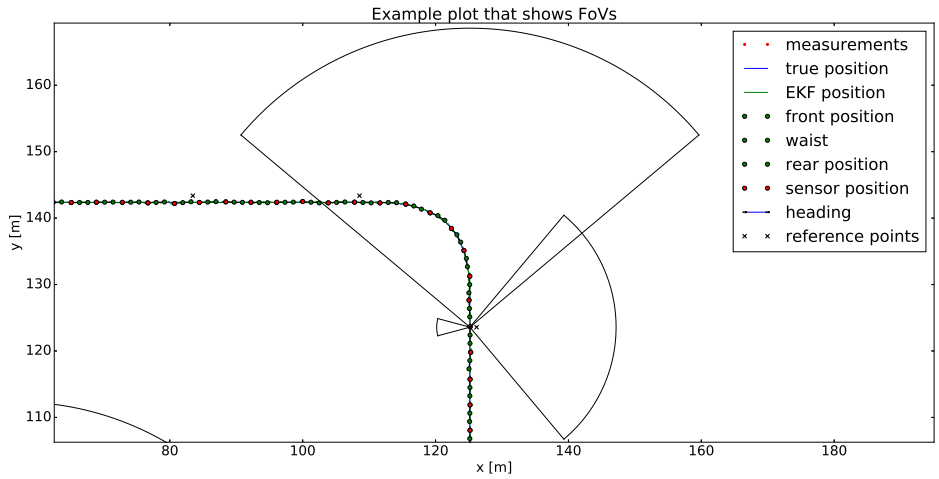


Figure 4.2: Illustrative example of the specific field of views for the different sensors. The camera is pointing to the right, radar is pointing upwards and to the left is the ultrasonic sensor.

Reasonable parameters for range and angle are summarized in Table 4.1 and was found in the literature [38, 40, 41] and from data sheets [42, 43].

Table 4.1: Table of the two parameters, range and angle, that defines the size of the specific FoV for every sensor.

sensor	**range** [m]	**angle** [°]
radar	45	±50
camera	22	±50
ultrasonic	5	±15

Depending on the weather condition, the sensor types are affected differently. What is significantly affected is how far away the sensors can observe an object [44]. This effect was modelled by reducing the range according to the sensor type and weather

condition. The radar is only slightly affected by precipitation and not at all by the dark. For the camera, the range is reduced to a third of the range in beneficial (i.e. clear) weather when it is dark and in the case of precipitation, the range is reduced to a sixth of the actual range [44]. The range of the ultrasonic sensor is not affected at all by the two other weather conditions. The final ranges of the sensors in the different weather conditions are summarized in Table 4.2.

Table 4.2: Table with range of the field of view for the different sensor types when operating in the weather conditions; dark and precipitation.

Sensor	**Dark** [m]	**Precipitation** [m]
radar	45	39
camera	7	4
ultrasonic	5	5

4.4 How reference points are detected by the sensors

For a reference point to be detected by the sensor, it must be inside of the sensor's field of view. This can be checked by three conditions that all need to be fulfilled.

A FoV of a sensor can be illustrated by a circle segment and the reference point as a point defined by x and y coordinates. The three conditions that needs to be fulfilled are:

1. The point must be located counter-clockwise to the right end of the circle segment.
2. The point must be located clockwise to the left end of the circle segment.
3. The distance between the point and the center of the circle segment must be less than the radius of the circle segment.

The first two conditions in the list above can be checked with simple equations and for simplicity, the method will be explained with two general vectors, $\mathbf{v}_1 = [x_1, y_1]$ and $\mathbf{v}_2 = [x_2, y_2]$. To test if $\mathbf{v}_2$ is counter-clockwise to $\mathbf{v}_1$, the size of the projection of $\mathbf{v}_2$ on the counter-clockwise normal, $\mathbf{n}_1 = [n_{1,x}, n_{1,y}] = [-y_1, x_1]$, to $\mathbf{v}_1$ is checked. If the projection is positive, this implies that $\mathbf{v}_2$ is counter-clockwise to $\mathbf{v}_1$. The equations are as follows:

$$x_2 n_{1,x} + y_2 n_{1,y} > 0 \rightarrow \mathbf{v}_2 \text{ counter-clockwise to } \mathbf{v}_1 \tag{4.1}$$

where Eq. (4.1) is the dot product between $\mathbf{v}_2$ and $\mathbf{n}_1$. Because of the definition of $\mathbf{n}_1$, the projection of $\mathbf{v}_2$ on $\mathbf{n}_1$ can directly be defined as a function of $\mathbf{v}_2$ and $\mathbf{v}_1$:

$$\begin{aligned} -x_2 y_1 + y_2 x_1 > 0 &\rightarrow \mathbf{v}_2 \text{ counter-clockwise to } \mathbf{v}_1 \\ x_2 y_1 - y_2 x_1 > 0 &\rightarrow \mathbf{v}_2 \text{ clockwise to } \mathbf{v}_1 \end{aligned} \tag{4.2}$$

The third condition is true if the magnitude of a vector, $\mathbf{u} = (u_1, u_2)$, is less than a given radius, r. This can be calculated using Pythagoras theorem:

$$\sqrt{u_1^2 + u_2^2} < r \tag{4.3}$$

This concludes the last of the most important elements of the urban simulation.

4.5 Investigation of different sensor setups and demands on the infrastructure

To investigate how well different sensor setups are estimating the pose of the vehicle, a couple of different setups were chosen and are summarized in Table 4.3. In order to safely having an autonomous vehicle operating in an urban environment, the sensors must cover 360° around the vehicle [45]. Therefore, the first setup was chosen to make sure that criteria was fulfilled. The other setups are simply small modifications of the first one. However, the last setup in Table 4.3, setup 4, does not fulfill the criteria. That is because it was of interest to test how well pose estimation can be done without 360° coverage.

Table 4.3: Table of the different sensor setups and how many of each sensor type that was chosen.

Setup	**radars**	**cameras**	**ultrasonics**	**mounting angle**
1	4	4	4	evenly spread to get 360°
2	3	3	2	evenly spread to get 360° but gaps will appear
3	5	5	6	evenly spread to get 360° with more available data
4	1	1	2	radar and camera facing forward and ultrasonic to the sides

4.6 Evaluation of sensor setups, demands on the operational environment and weather impact

To get a result of how well the sensor setups performed, an RMSE value was calculated for every method with the distance between the reference point fixed at 25 m. The distance was chosen as it is the same as the recommended distance between lampposts in a city environment in Sweden. The evaluation of the demands of the operational environment, given the different setups, was based on an iterative process where the distance between the reference points was decreased until an acceptable RMSE was achieved. For this iterative process, the weather condition was fixed as clear weather and the initial distance between the reference points was 25 m. Finally, to evaluate how the different weather conditions complicate the pose estimation, the same iterative process was performed for the different weathers.

5

Results & Discussion - Implementation and Evaluation of Navigation Methods

This chapter includes the results on which the evaluation of the four navigation methods described in Section 3.1, was made. After briefly explaining the implementation, the results of the simulation experiments are presented. This is done by showing the position and heading estimates from each method compared to ground truth data, followed by the absolute value of the estimation error of the methods and their corresponding RMSE value. Thereafter, the results of the real scenario experiments, when using the logged dataset, is presented. These results include pose estimates of the three methods that were tested. Lastly, plots of the innovation of the EKF based methods are shown. These can be used to overview how well the filters are working when no ground truth data is available.

The following sections are meant to highlight the parts of the results that were considered most significant to the study. To more easily compare the performance of the different methods, they have all been plotted in the same figures. For completeness, the individual results of the simulations can be seen in Appendix D.

5.1 Implementation details

To initiate the filters of the EKF based methods, a prior that was known to be relatively close to the actual start pose of the vehicle was chosen. Furthermore, a corresponding uncertainty that was relatively large was selected.

The covariance matrices representing the distributions of the measurement and process noises, that were used in the experiments (both simulations and using real sensor data), can be seen in Appendix B. The measurement covariances related to the position were given in the datasheet of the GNSS unit. The measurement covariance related to the steering angle sensor was calculated based 200 data points, during a sequence when the vehicle stood still. Lastly, the measurement covariance matrices concerning the IMU and magnetometer data were also calculated using 200 data points from the simulated dataset and the logged dataset. The 200 data points that were utilized for the covariances used in the simulations were taken from a sequence when the vehicle was moving straight forward at constant velocity, and

hence not affecting the accelerometer and gyroscope. Furthermore, the 200 data points used from the logged files were taken from a sequence when the vehicle stood still.

Concerning the process noise matrices, they were chosen by visually observing how different configurations of the matrices affected the behaviour of the navigation methods in the simulation experiments. First, the matrices were set to the identity matrix. Then, the elements of the diagonal were tuned by either being increased or decreased, with the goal that the position and heading estimates would be as close to the ground truth as possible. The final matrices that were chosen were the ones, among the tested, when the pose estimates achieved the best performance, relative to the ground truth. Due to the time limit of the project, a more extensive tuning of the process noise parameters was not made. The matrices that were ultimately chosen in the simulated case were then also used in the real scenario experiments.

5.2 Simulation experiments

The first experiment was to test all the navigation methods, using the simulated dataset. In Fig. 5.1 a zoomed in part of the ground truth position and heading (explained in Section 3.2.1), along with the estimates of each method, is visualized. It can be seen how well the different navigation methods estimate position and heading, by how well the ground truth graphs are replicated. The closer a graph of an estimate is to the ground truth, the better the method is considered to work. From the figure, it can be seen that the navigation methods that are based on EKF are close to the ground truth. In contrast, the GNSS Previous point method deviates a lot from ground truth (especially in curves) and appears to be more exposed to noise. For example, in the heading plot (right), by visual inspection, the estimate is fluctuating about $\pm 10°$ when the ground truth heading is constant.

Since the EKF based methods are closer to ground truth than GNSS Previous point, they also have an estimation error of smaller magnitude. This can be seen in Fig. 5.2, which shows the absolute value of the estimation error at each time step for the first half of the sequence. Again, it is evident that the magnitude of the estimation error is relatively small for the EKF based methods compared to GNSS Previous point. The EKF based methods rarely have a positional error that exceeds 0.1 m and a heading error that is larger than 5°. Regarding GNSS Previous point, the absolute positional estimation error sometimes is as large as about 0.5 m, while the heading estimate can be up to around 25° erroneous. It can also be seen that the magnitude of the estimation error of the GNSS Previous point method increases when the vehicle is turning, that is where the spikes occur.

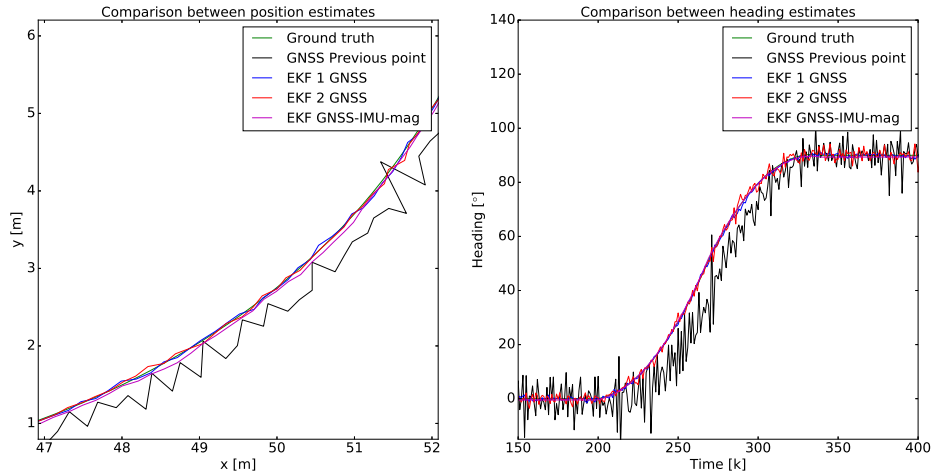


Figure 5.1: Zoomed in parts of the simulated ground truth pose, along with the estimates of the position trajectory (left) and heading (right). The EKF based methods are closer to the ground truth than the GNSS Previous point method.

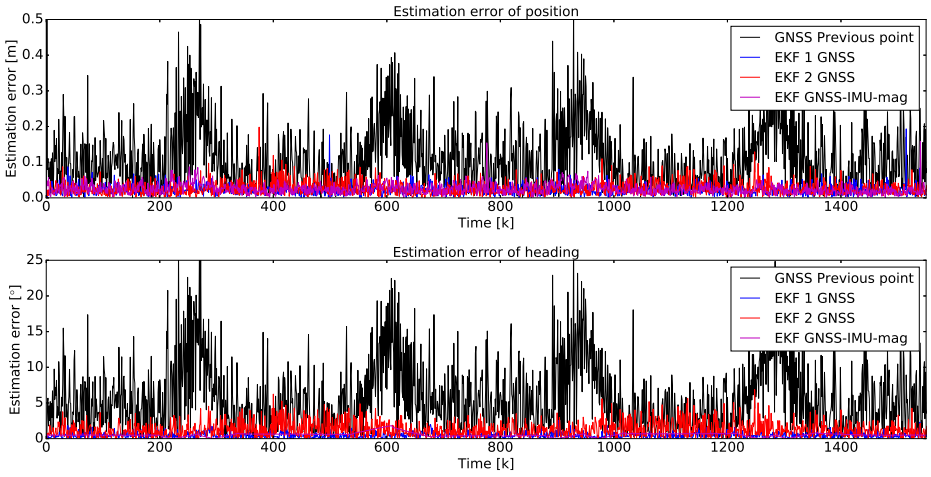


Figure 5.2: The absolute value of the estimation error, during the first half of the sequence, for the different navigation methods. The upper plot is the estimation error for position and the lower plot is the absolute value for heading. The GNSS Previous point method suffers from the largest error.

The calculated RMSE values of each method are presented in Table 5.1. Because of the constantly high estimation error, that was obtained using the GNSS Previous point method, it also has the highest RMSE for both position and heading, at about 0.15 m, respectively 7.79°. The other three methods were close to each other, but the EKF GNSS-IMU-mag method had the lowest heading RMSE value, which was approximately 0.59°. However, the navigation method that only uses information from one GNSS unit is slightly better at estimating the position, with an RMSE around 0.027 m.

Table 5.1: RMSE values for the different navigation methods. The lowest position RMSE was achieved by EKF 1 GNSS, while EKF GNSS-IMU-mag had the lowest heading RMSE.

Method	**RMSE position [m]**	**RMSE heading [°]**
GNSS Previous point	0.150	7.794
EKF 1 GNSS	0.027	0.645
EKF 2 GNSS	0.032	1.629
EKF GNSS-IMU-mag	0.028	0.590

5.3 Real scenario experiments

After the navigation methods were tested on simulated data, three of them were also tested on the logged dataset. As motivated in Section 3.2, the EKF 2 GNSS could not be tested since all the required data was not in the dataset.

In Fig. 5.3 is a plot of the resulting position estimate and a zoomed in area of the same plot. It can be seen that the GNSS Previous point and EKF 1 GNSS, lie close to each other. Concerning the EKF GNSS-IMU-mag method, it follows the other two methods closely in the beginning of the sequence. However, as time passes, a slight offset to the other methods can be distinguished.

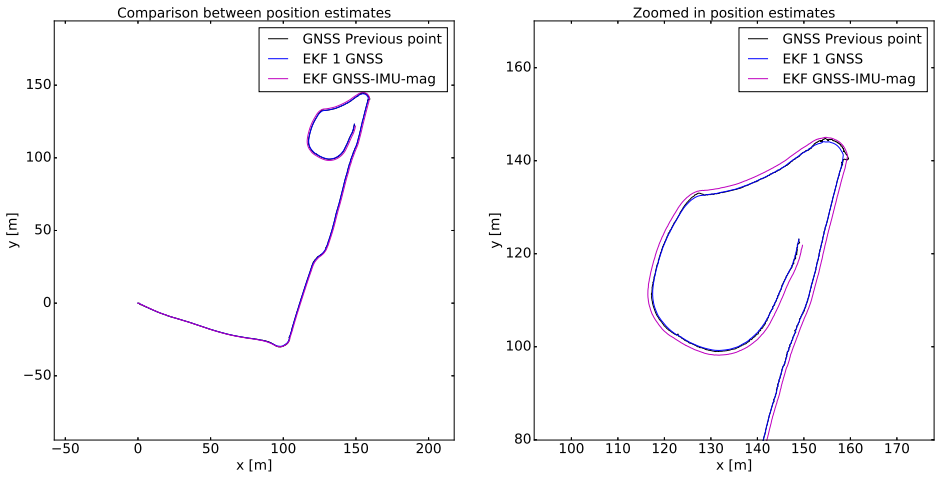


Figure 5.3: The position estimate for each method (left) and a zoomed in area on the same plot (right). As time passes, a small offset can be seen between the EKF GNSS-IMU-mag and the other two methods.

The same trend can also be seen for the heading estimates, which are visualized in Fig. 5.4. The left plot is the heading estimates seen over the whole sequence and the right plot illustrates the first 1400 time steps of the same plot. Compared to Fig. 5.3 the growing offset between the EKF GNSS-IMU-mag and the other methods is more clearly distinguished. Furthermore, the other two methods seem to follow the same trend, with the exception that GNSS Previous point has larger fluctuations.

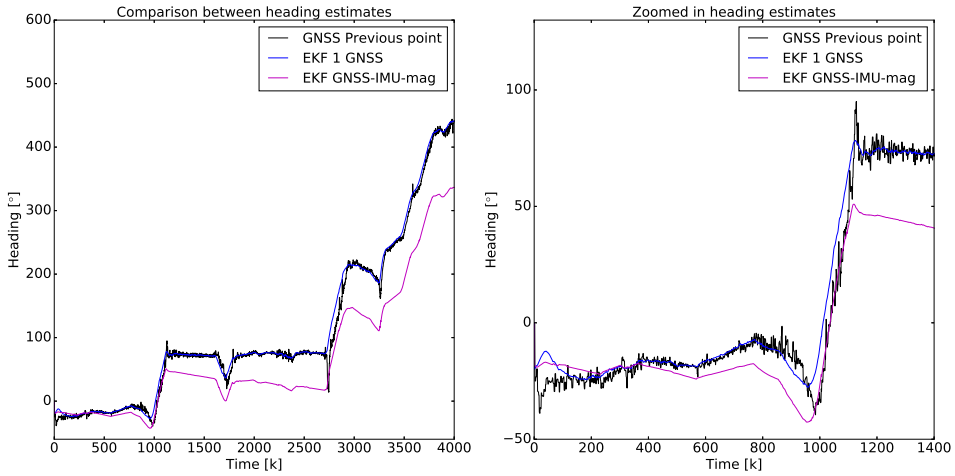


Figure 5.4: The heading estimate for each method (left) and the estimated heading during the first 1400 time steps (right). A growing offset can be seen between the EKF GNSS-IMU-mag and the other methods.

Since there is no ground truth data available for the logged data, it is not possible to calculate the RMSE. However, as explained in Section 3.3 the performance of the EKFs can be analyzed based on the computed innovation at each recursion of the filter. In Fig. 5.5 is the innovation of the x- and y-position for the EKF 1 GNSS method during the whole sequence.

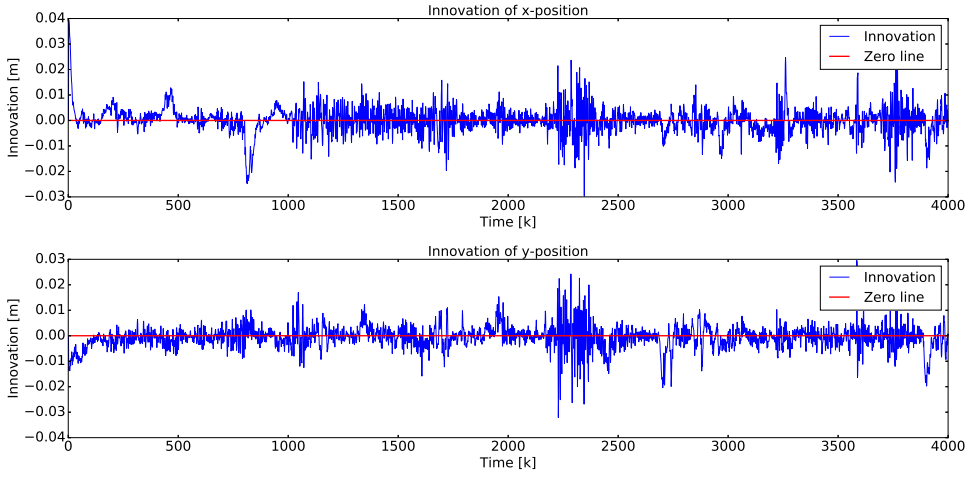


Figure 5.5: The innovation at each filter recursion for the x- (upper) and y-position (lower) for the EKF 1 GNSS method. The innovation has a mean close to zero, but is correlated during parts of the sequence.

It can be seen that the mean of the innovation of both components is close to zero. Calculating the mean value of the innovation showed that x-position fluctuates around $1.46 \cdot 10^{-6}$ m and the y-position around $-6.39 \cdot 10^{-4}$ m. However, the process is not consistently uncorrelated throughout the whole sequence. This is especially

evident in the initial 1000 time steps of the innovation of the x-position, where multiple consecutive values appear on the same side of the mean. Meaning that this is not representative of the behavior of WGN.

In Fig. 5.6, the resulting innovation for the x- and y-component of the position for the EKF GNSS-IMU-mag method, can be seen. In comparison to the innovation in Fig. 5.5 it has a mean that is not consistently zero. Furthermore, the innovation is also clearly correlated, which is as most obvious at about time step 2800 and forward.

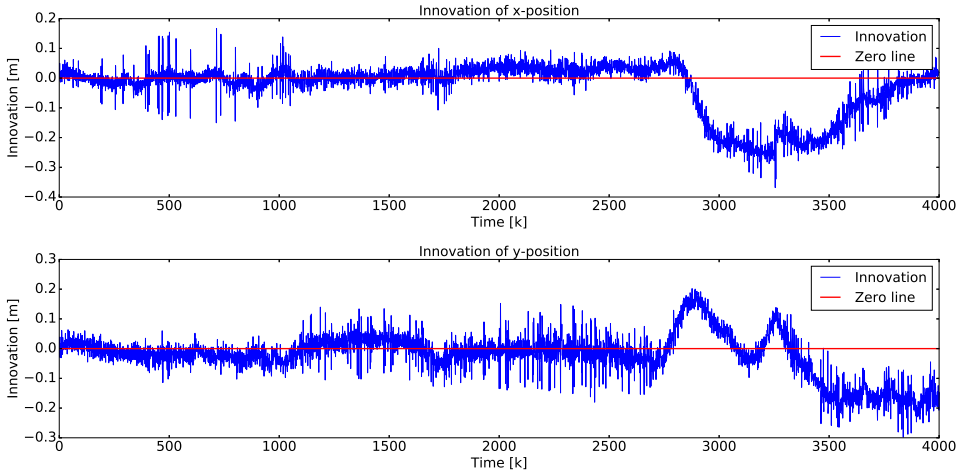


Figure 5.6: The innovation at each filter recursion for the x- (upper) and y-position (lower) for the EKF GNSS-IMU-mag method. The innovation is not zero-mean and is not uncorrelated during the sequence.

5.4 Discussion

The results of the evaluation of the navigation methods can be analyzed based on several aspects. If the first experiments are considered, where simulated data was used, it showed that the method EKF GNSS-IMU-mag performed the best in terms of estimating heading. This is based on that its calculated RMSE had the lowest value, which was approximately 0.59°. When it comes to estimating the position, the EKF 1 GNSS method achieved the highest accuracy, having an RMSE of about 2.7 cm. However, this is insignificantly better than the EKF GNSS-IMU-mag method, which had an RMSE value that was only 1 mm higher. Thus, based on the simulation experiments, the EKF GNSS-IMU-mag method was considered as the overall best performing navigation method.

The outcome of real scenario experiments, based on the logged dataset, provided a different result. However, since there was no ground truth data to be used, the analysis of the results is not as straightforward as in the simulation case. The experiments showed that the position estimates from EKF 1 GNSS and GNSS

Previous point were roughly in the same region during the whole sequence, whereas the EKF GNSS-IMU-mag method has a slight offset compared to the other methods in the latter part of the sequence. This is because the simplification of estimating the position of the GNSS unit within the EKF and then post process the position using the heading estimate was made. Since the estimated heading tends to drift over time, this also affects the final position estimate. However, since measurements from the GNSS unit are utilized, the offset of the position estimate will not grow boundlessly. As can be seen in Eq. (3.32) and Eq. (3.33), the impact of the heading on the position estimate depends on the Euclidean distance between the front position and mounting point of the GNSS unit, which is 1.01 m. Therefore, the magnitude of the offset of the position estimate will never exceed 1.01 m.

The drift of the EKF GNSS-IMU-mag method can have several explanations. One likely explanation is that the real IMU and magnetometer data is biased. If this is the case, one solution that could make the pose estimate more accurate is to include a bias estimate, for both the gyroscope and accelerometer, in the state vector. It is also very likely that the magnetometer data is corrupted by local magnetic fields and this can be compensated for by a better calibration of the sensor before collecting data. In contrast, the simulated data of the IMU and magnetometer was made by generating perfect measurements, that later were corrupted by WGN. However, these measurements were always fluctuating around this non-biased mean, resulting in that the filter behaves in a desirable way.

Furthermore, the innovation of the two EKF based methods especially showed that the filter of EKF GNSS-IMU-mag does not behave as desired. This is evident, as its innovation consistently was neither zero-mean nor uncorrelated. With these two arguments, it is therefore strongly believed that the EKF GNSS-IMU-mag estimates deviate the most from the actual pose of the vehicle.

Focusing on the EKF 2 GNSS method, one could on beforehand expect that it would outperform the EKF 1 GNSS, since merging information from two GNSS units intuitively would result in a more accurate pose estimate, than when only using one. However, even though more measurements are available, the geometrical configuration of the mounting points of the two units are believed to arise larger uncertainties, compared to the EKF 1 GNSS method, where the GNSS unit is mounted at a different location. The reason for this is explained as follows. The measurement model that was used in this method, defined by Eq. (3.15), shows that the relation between the front position and the position of the GNSS unit depends on the heading. Since the heading is estimated, it will always have uncertainties, and if it is erroneous compared to the true heading, this will result in that calculated position of the GNSS unit will also be erroneous. As can be seen in Eq. (3.15), the impact of the heading depends on the Euclidean distance, ℓ_{gnss}, between the positions of the sensors and the front position. Comparing the choices of mounting points for the single GNSS unit, and when using two units, as well as applying Pythagora's theorem, it is evident that ℓ_{gnss} will always be larger for the EKF 2 GNSS method. The consequence is that the pose estimate becomes more uncertain

for this method, compared to the EKF 1 GNSS. Thus, it can be concluded that the choice of where the GNSS units were modelled to be mounted affected the accuracy of the pose estimate. Therefore, it is important to mount GNSS units near the position of interest and unnecessary displacements should be avoided.

Finally, when considering the GNSS Previous point method, it is expected to be performing the worst as no post processing of the estimate is done. However, the true position and heading are likely to be located somewhere within the area between the rippled estimates as the method builds on measurements from GNSS, which has high accuracy. Therefore, it can be argued that this method is more reliable over time than EKF GNSS-IMU-mag. The estimates from the GNSS Previous point will at least be in the same region as the actual heading.

In summary, when taking both the results of the simulations and the real scenario experiments into account, the EKF 1 GNSS is considered to be the most reliable navigation method, among the tested. The EKF GNSS-IMU-mag showed great potential in the simulations but lacked in terms of robustness when tested on real sensor data. If this method is further developed to be less sensitive to erroneous measurements, it might become a very accurate navigation method.

6

Results & Discussion - Urban Simulation

In this chapter, the results of the simulation of navigating in urban environment are presented. First, implementation details are presented and that is followed by sections for results of sensor setups, weather conditions and demands on the operational environment. The chapter ends with a discussion section.

6.1 Implementation details

As mentioned in Section 4.1, position measurements are generated when a reference point is inside a sensors field of view and these measurements are corrupted by white Gaussian noise, proportional to position accuracy for every sensor type. The noise is modelled by a measurement covariance matrix, $\mathbf{R}$, used in the update step of the Kalman filter.

$$\mathbf{R} = \begin{bmatrix} \Sigma_{rad} & 0 & 0 \\ 0 & \Sigma_{cam} & 0 \\ 0 & 0 & \Sigma_{ult} \end{bmatrix} = \begin{bmatrix} 0.1^2 & 0 & 0 \\ 0 & 0.037^2 & 0 \\ 0 & 0 & 0.15^2 \end{bmatrix},$$

where $\Sigma_{rad}, \Sigma_{cam}$ and Σ_{ult} are the variances for the radar, camera and ultrasonic sensor respectively. However, note that the size of the matrix will adapt in every iteration in order to match the number of available measurements. The matrix above is only an example to illustrate the numerical values for the different sensors. Finally, after tuning, a process noise covariance matrix $\mathbf{Q}$ was generated.

$$\mathbf{Q} = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10.0 \end{bmatrix}$$

6.2 Sensor setups

In Table 6.1, the RMSE values for the different sensor setups are presented. The parameters weather and distance between reference points are fixed at clear, respectively 25 m. In the table, it can be seen that setup 3 achieved the best

results, in terms of both position and heading. The corresponding RMSE values were approximately 0.030 m, respectively 1.13°.

Table 6.1: Table with RMSE values for position and heading for the different sensor setups with a fixed distance of 25 m between the reference points and weather condition set to clear.

Setup	**RMSE position** [m]	**RMSE heading** [°]
1	0.033	1.286
2	0.043	1.563
3	0.030	1.134
4	0.142	2.156

6.3 Weather conditions

The next aspect that was studied was the demands on the operational environment, via an iterative process. In Fig. 6.1, the RMSE for position and heading, in relation to distance between the reference points in clear weather is depicted. The dashed line is a threshold value that represents acceptable error. The threshold was chosen to be the best achieved RMSE values for position and heading in the evaluation of different navigation methods (see Section 5.2), which were 0.027 m and 0.59°. These values were chosen as it is desired to reach the same accuracy as when navigating with GNSS under favorable conditions (i.e. no major disturbances). From Fig. 6.1 it can be concluded that setup 3 reaches the threshold with the largest distance between reference points, for both position and heading. Also, noteworthy in the figure is that the curves not linearly decrease. That is due to the randomness associated with the measurements. And the same is true about the following plots showing the results of weather impact.

In Fig. 6.2 the RMSE, for position and heading, in relation to the distance between the reference points in dark weather conditions is depicted. It can be seen that the threshold is reached for position accuracy by all setups and for all besides setup 4 for heading accuracy. Furthermore, it can also be seen that the distance between the reference points is shorter than for clear weather when the setups reach the accuracy threshold.

The last weather condition that was considered was precipitation. In Fig. 6.3 the RMSE, for position and heading, in relation to the distance between the reference points, in this weather condition, is depicted. It can be concluded that the different setups are following the same trend and that the distance between reference points is mush shorter than for clear weather. It can also be seen that the threshold for position is reached by all setups whereas only setup 3 and setup 1 reach the heading accuracy.

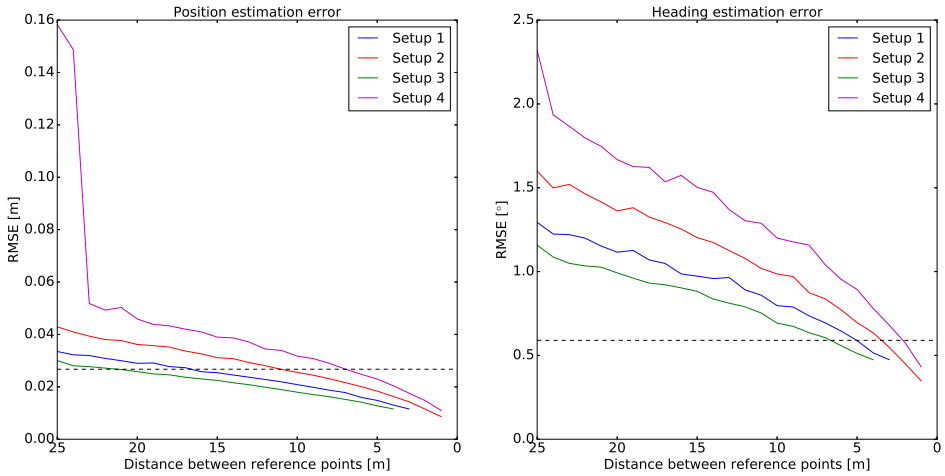


Figure 6.1: Plot with RMSE values for position (left) and heading (right) for the different sensor setups when the weather condition is clear.

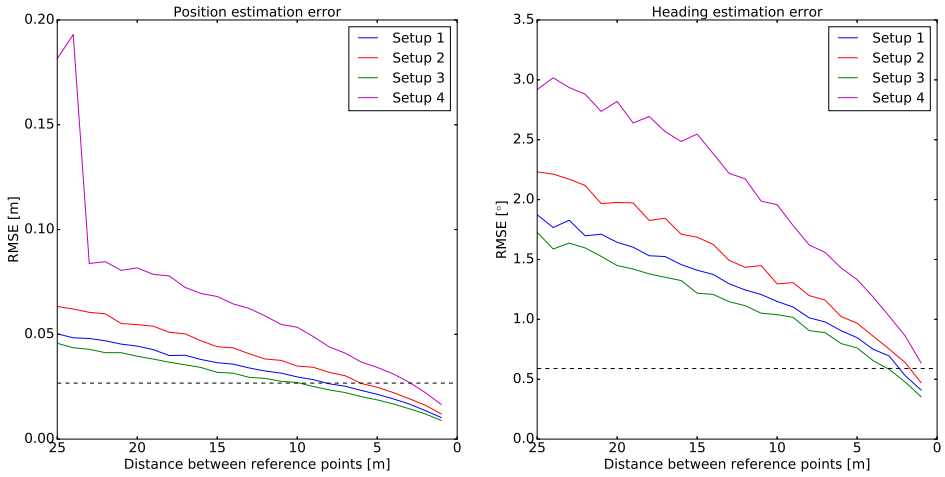


Figure 6.2: Plot with RMSE values for position (left) and heading (right) for the different sensor setups when the weather condition is dark.

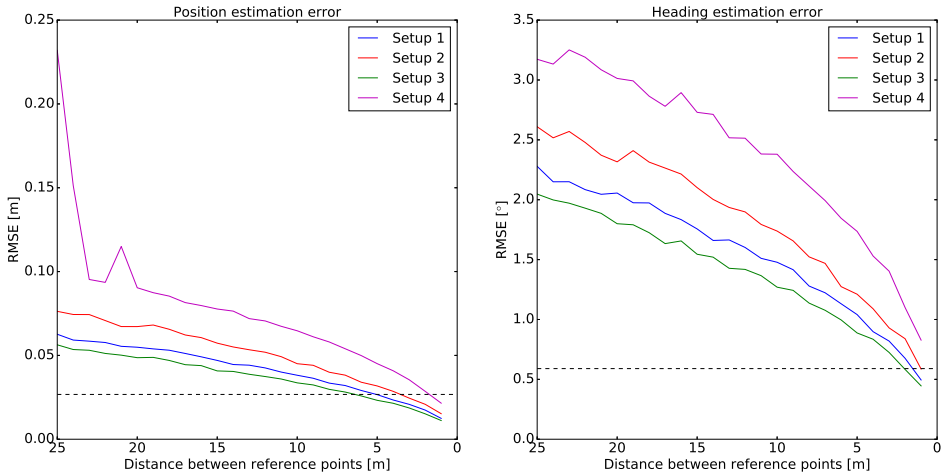


Figure 6.3: Plot with RMSE values for position (left) and heading (right) for the different sensor setups when the weather condition is set to precipitation.

6.4 Demands on operational environment

The results from the investigation of the demands on the operational environment are summarized in Table 6.2. The presented values are the distance between the reference points that ensures the desired accuracy for heading, for the different weather conditions. As the accuracy for position was reached before the heading condition, the presented distance in Table 6.2 ensures desired accuracy for both position and heading. From the table, it can be seen that 6 m is the longest distance that still assures sufficiently accurate navigation when the weather condition is clear and using the best performing sensor setup. Furthermore, it can also be seen that sensor setup 4, with no 360° coverage around the vehicle, only met the required accuracy for clear weather.

Table 6.2: Table with the distance between the reference points, for the different weather conditions, that ensures the desired accuracy.

Sensor setup	**Clear** [m]	**Dark** [m]	**Precipitation** [m]
1	4	2	1
2	3	2	1
3	6	3	2
4	2	-	-

6.5 Discussion

When discussing the different sensor setups and demands on the operational environment, it is more interesting to discuss the relations rather than the numerical values of the results. For example, it is trivial that sensor setup 3 delivers the best

result (the more sensors, the more measurements and hence, better estimations), but what is interesting is that all setups reach the predefined accuracy at a distance, not far from each other. This can mean that the setups are very similar to each other and not differ enough to see any significant variations. Another explanation to the result can be that it is not that important which setup that is chosen as it does not seem to make any bigger difference. Another interesting result is that the distance where the different setups reach the required accuracy is a lot shorter than the initial distance of 25 m. That could indicate that the demands on the operational environment are high and reference points need to be placed at a short distance between each other.

The simulation is, as mentioned earlier, simplified in different ways and hence, the results are considered as pointers and not as strict facts. It is obvious that the simulation could deliver more reliable results if further work and time was invested. One aspect that needs to be further developed is how raw measurements from the local sensors are converted into pose estimates, as this has not been included. Also if reference points are temporarily shadowed by other reference points, the sensors would not be able to see them. This is another aspect that at the moment has not been considered. A third potential improvement would be to simulate a GNSS unit that during some sequences provide reliable measurements, and sometimes very inaccurate data, which is a likely scenario in a city. It would be interesting to see how robust the navigation is if suddenly some measurements are corrupted.

Another interesting thing about the simulation is that a lot of different aspects can be tested and investigated. The presented results in this report are only a selection of what could be investigated. For example, different sensors can be tested in relation to each other and not only different setups.

7
Conclusions & Future Work

From the evaluation of the different navigation methods it was concluded, based on simulated experiments, that the lowest RMSE for position and heading that was achieved was 0.027 m by using the EKF 1 GNSS method, respectively 0.59° by using the EKF GNSS-IMU-mag. However, the EKF GNSS-IMU-mag proved to be sensitive to biased sensor measurements in the experiments, based on logged data. Therefore, it can be concluded that the GNSS 1 EKF method performed the best, among the tested, when considering both the simulated and logged datasets.

Since the EKF GNSS-IMU-mag showed potential under simulated conditions, further development would be to make it more robust in "real world" conditions. One improvement could be to estimate the sensor biases, which hopefully will reduce the present drifts of the pose estimate.

Based on the investigation of the urban simulation, it was concluded that 6 m is the shortest distance between the reference points that can assure accurate navigation in clear weather conditions for the setup that performs the best among the tested. This setup was configured of five radars and cameras, and six ultrasonic sensors, evenly spread to get 360° coverage around the vehicle. Furthermore, it was concluded that precipitation is the worst weather condition for the selected sensor setups. Finally, it was concluded that it seems like it is not enough to have 25 m between reference points, which is the case if they were mounted on lampposts in a generic Swedish city. Hence, other elements in the operational environment need to be used as mounting points for sensor reflectors.

At the moment a lot of simplifications and assumptions have been made in the urban simulation, making it less representative for a real scenario. Therefore, potential future work would be to make it more complex, partly in how the estimation of the pose of the vehicle is derived, but also when it comes to modelling phenomenons that are likely to occur in urban environment.

Bibliography

[1] J. Hecht, "Self-driving vehicles: Many challenges remain for autonomous navigation," *LaserFocusWorld*, 2020.

[2] MIT Technology Review Insights. (2019, mar) Autonomous driving: Safety first. [Online]. Available: https://www.technologyreview.com/2019/03/25/65933/autonomous-driving-safety-first/

[3] Nationalencyklopedin. navigation. [Online]. Available: http://www.ne.se/uppslagsverk/encyklopedi/lång/navigation

[4] W. Rahiman and Z. Zainal, "An overview of development GPS navigation for autonomous car," in *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, 2013, pp. 1112–1118.

[5] L. Hsu, H. Tokura, N. Kubo, Y. Gu, and S. Kamijo, "Multiple Faulty GNSS Measurement Exclusion Based on Consistency Check in Urban Canyons," *IEEE Sensors Journal*, vol. 17, no. 6, pp. 1909–1917, March 2017.

[6] S. Clark and H. Durrant-Whyte, "Autonomous land vehicle navigation using millimeter wave radar," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 4, 1998, pp. 3697–3702 vol.4.

[7] M. N. Cahyadi and I. Rwabudandi, "Integration of GNSS-IMU for increasing the observation accuracy in Condensed Areas (Infrastructure and Forest Canopies)," *E3S Web of Conference*, vol. 94, no. 03015, may 2019.

[8] M. Kok, J. D. Hol, and T. B. Schön, "Using Inertial Sensors for Position and Orientation Estimation," *Foundation and Trends in Signal Processing*, vol. 11, no. 1-2, pp. 1–153, Jun 2017.

[9] G. Trigo, D. Donas-Boto, C. Silva, and J. E. Sanguino, "Vehicle Heading Estimation Using a Two Low-Cost GPS Receiver Configuration," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, May 2011, pp. 1–5.

[10] S. Madgwick. (2010) "An efficient orientation filter for inertial and inertial/magnetic sensor arrays". [Online]. Available: https://www.x-io.co.uk/res/doc/madgwick_internal_report.pdf

[11] European GNSS Agency. (2017) “What is GNSS?”. [Online]. Available: https://www.gsa.europa.eu/european-gnss/what-gnss

[12] R. Emardson, P. Jarlemark, S. Bergstrand, T. Nilsson, and J. Johansson, “Measurement accuracy in Network-RTK,” SP Technical Research Institute of Sweden, SP Report, 2009.

[13] Swift Navigation, *Duro User Manual*, Swift Navigation.

[14] V. Renaudin, M. H. Afzal, and G. Lachapelle, “New method for magnetometers based orientation estimation,” in *IEEE/ION Position, Location and Navigation Symposium*, 2010, pp. 348–356.

[15] C. Wolff. (n.d) Frequency-Modulated Continuous-Wave Radar (FMCW Radar). [Online]. Available: https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html

[16] P. Hauptmann, R. Lucklum and B. Henning, “Ultrasonic Sensors for Process Control,” *Wiley Online Library*, vol. 3, no. 1, pp. 163–207, 2001.

[17] N. Kothari, M. Gupta, L. Vachhani, and H. Arya, “Pose estimation for an autonomous vehicle using monocular vision,” in *2017 Indian Control Conference (ICC)*, 2017, pp. 424–431.

[18] Trimble. (n.d.) Critical factors affecting RTK accuracy. [Online]. Available: https://www.trimble.com/OEM_ReceiverHelp/V4.44/en/PositionModes_CriticalFactorsRTK.html

[19] NovAtel. (n.d.) An Introduction to GNSS. [Online]. Available: https://www.novatel.com/an-introduction-to-gnss/chapter-4-gnss-error-sources/error-sources/

[20] R. Odolinski, “Swedish User Guidelines for Network RTK,” GNSS Research Centre, Tech. Rep., 2010.

[21] A. P. Andrews, M. S. Grewal, and L. R. Weill, *Global Positioning Systems, Inertial Navigation and Integration.* New York, NY: John Wiley & Sons, Inc., 2001.

[22] National Coordination Office for Space-Based Positioning, Navigation, and Timing. (2017) The Global Positioning System. [Online]. Available: https://www.gps.gov/systems/gps/

[23] Lantmäteriet. (n.d.) UTM. [Online]. Available: https://www.lantmateriet.se/sv/Kartor-och-geografisk-information/gps-geodesi-och-swepos/Om-geodesi/Kartprojektioner/UTM/

[24] M. Ebrahimi, X. Chang, M. Daneshtalab, J. Plosila and P. Liljeberg, “DyXYZ: Fully Adaptive Routing Algorithm for 3D NoCs,” Department of Information Technology, University of Turku, Finland and Department of Electronic and Microelectronics, University of Mons, Belgium, Tech. Rep., 2013.

[25] P. I. Corke and P. Ridley, "Steering kinematics for a center-articulated mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, pp. 215–218, April 2001.

[26] S. Särrkä, *Bayesian Filtering and Smoothing*, ser. IMS Textbooks. Cambridge University Press, 2013, vol. 3.

[27] L. Hammarstrand, "Filtering, smoothing and predictions," 2019, lecture notes from the course: Sensor fusion & nonlinear filtering, held at Chalmers University of Technology.

[28] J. Z. Sasiadek and P. Hartana, "Sensor data fusion using Kalman filter," in *Proceedings of the Third International Conference on Information Fusion*, vol. 2, 2000, pp. 19–25.

[29] L. Hong, "Discrete Constant-Velocity-Equivalent Multirate Models for Target Tracking," *Mathl. Comput. Modelling*, vol. 28, no. 11, pp. 7–18, may 1998.

[30] N. Banihashemi and C. Yalçın Kaya, "Inexact Restoration for Euler Discretization of Box-Constrained Optimal Control Problems," *J Optim Theory Appl*, no. 156, p. 726–760, 2013, https://doi.org/10.1007/s10957-012-0140-4.

[31] C. Jahanchahi and D. P. Mandic, "A Class of Quaternion Kalman Filters," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 533–544, 2014.

[32] E. B. Dam, M. Koch and M. Lillholm, "Quaternions, Interpolation and Animation," Department of Computer Science, University of Copenhagen, Tech. Rep., jul 1998.

[33] MathWorks. (n.d.) imuSensor. [Online]. Available: https://se.mathworks.com/help/nav/ref/imusensor-system-object.html

[34] navipedia. (2013, aug) Positioning Error. [Online]. Available: https://gssc.esa.int/navipedia/index.php/Positioning_Error

[35] L. Hammarstrand, "The kalman filter and its properties," 2019, lecture notes from the course: Sensor fusion & nonlinear filtering, held at Chalmers University of Technology.

[36] Trafikverket. (2014) Vägbelysningshandboken. [Online]. Available: https://www.trafikverket.se/contentassets/799732cdb35e439fbb1329106fecfac8/handbok_vagbelysning_ver_14_140625.pdf

[37] MathWorks. Model Radar Sensor Detections. [Online]. Available: https://se.mathworks.com/help/driving/examples/model-radar-sensor-detections.html

[38] P. Irmisch, "Camera-based distance estimation for autonomous vehicles," Master's thesis, Technical University Berlin, n.d.

[39] Pepperl+Fuchs. (2015, jan) Ultrasonic Sensors Knowledge (Part 4): Influences on Measurement Accuracy. [Online]. Available: https://www.pepperl-fuchs.com/global/en/25518.htm

[40] Point Grey Research. (2008, dec) Wide-angle lens offers 100° horizontal FOV. [Online]. Available: https://www.vision-systems.com/cameras-accessories/article/16745376/wideangle-lens-offers-100-horizontal-fov?fbclid=IwAR16IbAJRG4jBAdJmnn7BcRICO1q_JFJK88h24d82cHykNp4RNXZnYaSzfY

[41] J. Alcantar. (2016, nov) Ultrasonic Sensors – Field of View. [Online]. Available: https://www.arxterra.com/ultrasonic-sensors-field-of-view/

[42] SICK. (n.d.) Radar sensors RMS3xx. [Online]. Available: https://www.sick.com/ag/en/detection-and-ranging-solutions/radar-sensors/rms3xx/c/g448551

[43] Nex Robotics. (2016, nov) Ultrasonic Car Detection Sensor MB8450. [Online]. Available: http://www.nex-robotics.com/products/sensors/ultrasonic-range-sensors/ultrasonic-car-detection-sensor-mb8450.html

[44] M. Barnard. (2016, jul) Tesla & Google Disagree About LIDAR — Which Is Right? [Online]. Available: https://cleantechnica.com/2016/07/29/tesla-google-disagree-lidar-right/

[45] Felix. (2019, jan) Sensor Set Design Patterns for Autonomous Vehicles. [Online]. Available: https://autonomous-driving.org/2019/01/25/positioning-sensors-for-autonomous-vehicles/

[46] Y.-B. Jia. (2013, sep) Quaternions and Rotations*. [Online]. Available: http://graphics.stanford.edu/courses/cs348a-17-winter/Papers/quaternion.pdf

[47] Automatic Control Linköping. (2019, mar) Orientation Estimation using Smartphone Sensors. [Online]. Available: https://www.control.isy.liu.se/student/tsrt14/file/orientation.pdf

A

Mathematical Definitions of Quaternions and Related Operators

A.1 The unit quaternion

A quaternion is a tuple of four elements that is a more concise representation than a rotation matrix [46]. A unit quaternion, $\mathbf{q}$ is defined as

$$\mathbf{q} = \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{1}{2}\alpha(t)\right) \\ \sin\left(\frac{1}{2}\alpha(t)\right) \begin{bmatrix} \hat{v}_x \\ \hat{v}_y \\ \hat{v}_z \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{1}{2}\alpha(t)\right) \\ \sin\left(\frac{1}{2}\alpha(t)\right)\hat{\mathbf{v}} \end{bmatrix}, \tag{A.1}$$

where $\alpha(t)$ is a rotation angle about a unit vector $\hat{\mathbf{v}}$.

A.2 Quaternion multiplication

Given two quaternions $\mathbf{p} = [p_0,\ \mathbf{p}_v]^T$ and $\mathbf{q} = [q_0,\ \mathbf{q}_v]^T$, the following apply for the product of the two quaternions,

$$\mathbf{p} \odot \mathbf{q} = \begin{bmatrix} p_0 q_0 - \mathbf{p}_v \mathbf{q}_v \\ p_0 \mathbf{q}_v + q_0 \mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v \end{bmatrix}, \tag{A.2}$$

where $\odot$ is the quaternion product operator and $\times$ is the cross product operator [8].

A.3 The quaternion exponential

A given rotation vector, $\boldsymbol{\eta}$, can be expressed as a unit quaternion, $\mathbf{q}$, using the quaternion exponential operator $\exp_q(.)$ [8]. The quaternion exponential is defined as

$$\mathbf{q} = \exp_q(\boldsymbol{\eta}) = \begin{bmatrix} \cos(||\boldsymbol{\eta}||) \\ \frac{\boldsymbol{\eta}}{||\boldsymbol{\eta}||}\sin(||\boldsymbol{\eta}||) \end{bmatrix}. \tag{A.3}$$

A.4 Converting quaternions to rotation matrix

A quaternion, $\mathbf{q} = [q_0,\ q_1,\ q_2,\ q_3]^T$, can be converted into a rotation matrix $\mathbf{R}$ [8]. The conversion is defined as

$$\mathbf{R} = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}. \tag{A.4}$$

A.5 Differentiating a rotation matrix with respect to quaternions

Given a rotation matrix, $\mathbf{R}$, defined as Eq. (A.4), differentiated with respect to each component of the quaternion, results in a concatenation of four 3D tensors [47]. The resulting matrix is defined as

$$\frac{\partial \mathbf{R}}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial \mathbf{R}}{\partial q_0} & \frac{\partial \mathbf{R}}{\partial q_1} & \frac{\partial \mathbf{R}}{\partial q_2} & \frac{\partial \mathbf{R}}{\partial q_3} \end{bmatrix}, \tag{A.5}$$

where,

$$\frac{\partial \mathbf{R}}{\partial q_0} = 2\begin{bmatrix} 2q_0 & -q_3 & q_2 \\ q_3 & 2q_0 & -q_1 \\ -q_2 & q_1 & 2q_0 \end{bmatrix}, \tag{A.6}$$

$$\frac{\partial \mathbf{R}}{\partial q_1} = 2\begin{bmatrix} 2q_1 & q_2 & q_3 \\ q_2 & 0 & -q_0 \\ q_3 & q_0 & 0 \end{bmatrix}, \tag{A.7}$$

$$\frac{\partial \mathbf{R}}{\partial q_2} = 2\begin{bmatrix} 0 & q_1 & q_0 \\ q_1 & 2q_2 & q_3 \\ -q_0 & q_3 & 0 \end{bmatrix}, \tag{A.8}$$

$$\frac{\partial \mathbf{R}}{\partial q_3} = 2\begin{bmatrix} 0 & -q_0 & q_1 \\ q_0 & 0 & q_2 \\ q_1 & q_2 & 2q_3 \end{bmatrix}. \tag{A.9}$$

A.6 The right multiplication matrix operator

Given a quaternion, $\mathbf{q} = [q_0,\ q_1,\ q_2,\ q_3]^T$, the operator $(.)^R$ yields a so called right multiplication matrix [8]. The right multiplication matrix of the quaternion is given by

$$\mathbf{q}^R = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix}. \tag{A.10}$$

A.7 The left multiplication matrix operator

Given a quaternion, $\mathbf{q} = [q_0,\ q_1,\ q_2,\ q_3]^T$, the operator $(.)^L$ yields a so called left multiplication matrix [8]. The left multiplication matrix of the quaternion is given by

$$\mathbf{q}^L = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}. \tag{A.11}$$

B

Filter Parameters

Presented here are the final covariance matrices that were used in the three different navigation methods that are based on EKF theory.

B.1 Covariance matrices for EKF 1 GNSS

Process noise:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix} \tag{B.1}$$

Measurement noise:

$$\mathbf{R} = \begin{bmatrix} 0.01^2 & 0 & 0 & 0 \\ 0 & 0.015^2 & 0 & 0 \\ 0 & 0 & \left(0.1\frac{\pi}{180}\right)^2 & 0 \\ 0 & 0 & 0 & \left(\frac{\pi}{180}\right)^2 \end{bmatrix} \tag{B.2}$$

B.2 Covariance matrices for EKF 2 GNSS

Process noise:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix} \tag{B.3}$$

Measurement noise:

$$R = \begin{bmatrix} 0.01^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.015^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.015^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \left(0.1\frac{\pi}{180}\right)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \left(\frac{\pi}{180}\right)^2 \end{bmatrix} \tag{B.4}$$

B.3 Covariance matrices for EKF GNSS-IMU-mag

Since the measurements from the accelerometer and gyroscope were seen as input variables and included in the motion model, their corresponding measurement noise covariance matrices were viewed as process noise. Therefore, they are denoted by $\mathbf{Q}$ instead of $\mathbf{R}$, like other covariance matrices, coming from measurements, have been.

Process noise in simulations:

$$\mathbf{Q}_a = \begin{bmatrix} 0.000161208018411 & 0.000002785021867 & -0.000035370975502 \\ 0.000002785021867 & 0.000122832638566 & 0.000040004219164 \\ -0.000035370975502 & 0.000040004219164 & 0.001725727557812 \end{bmatrix} \tag{B.5}$$

$$\mathbf{Q}_\omega = 10^{-6} \begin{bmatrix} 0.626971172288596 & -0.013110365352934 & 0.000789606456770 \\ -0.013110365352934 & 0.910350625630546 & 0.017163337018573 \\ 0.000789606456770 & 0.017163337018573 & 0.220689746487556 \end{bmatrix} \tag{B.6}$$

Process noise in real scenario experiments:

$$\mathbf{Q}_a = \begin{bmatrix} 0.000632509173762 & -0.000245978700233 & -0.000389405556730 \\ -0.000245978700233 & 0.000437059027560 & 0.000121862530890 \\ -0.000389405556730 & 0.000121862530890 & 0.001121826275062 \end{bmatrix} \tag{B.7}$$

$$\mathbf{Q}_\omega = 10^{-6} \begin{bmatrix} 0.732238579310663 & 0.078439747753299 & -0.006374615351573 \\ 0.078439747753299 & 0.961445427020322 & -0.005957886433237 \\ -0.006374615351573 & -0.005957886433237 & 0.467745340439080 \end{bmatrix} \tag{B.8}$$

Measurement noise in simulations:

$$\mathbf{R}_p = \begin{bmatrix} 0.01^2 & 0 & 0 \\ 0 & 0.015^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{B.9}$$

$$\mathbf{R}_m = \begin{bmatrix} 0.001746378608419 & 0.000014296046762 & -0.000153129518536 \\ 0.000014296046762 & 0.000196785777573 & -0.000077265620674 \\ -0.000153129518536 & -0.000077265620674 & 0.003995485233041 \end{bmatrix} \tag{B.10}$$

Measurement noise in real scenario experiments:

$$\mathbf{R}_p = \begin{bmatrix} 0.01^2 & 0 & 0 \\ 0 & 0.015^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{B.11}$$

$$\mathbf{R}_m = 10^{-3} \begin{bmatrix} 0.000807716990114 & 0.000029054567993 & 0.000094986087659 \\ 0.000029054567993 & 0.005308108157573 & -0.000268972042345 \\ 0.000094986087659 & -0.000268972042345 & 0.005183433544995 \end{bmatrix} \tag{B.12}$$

C

Datasets

Additional simulated datasets that were used when evaluating the navigation methods. Since the focus was on estimating the pose of the vehicle, these are not as intuitive to analyze as the position and heading plots.

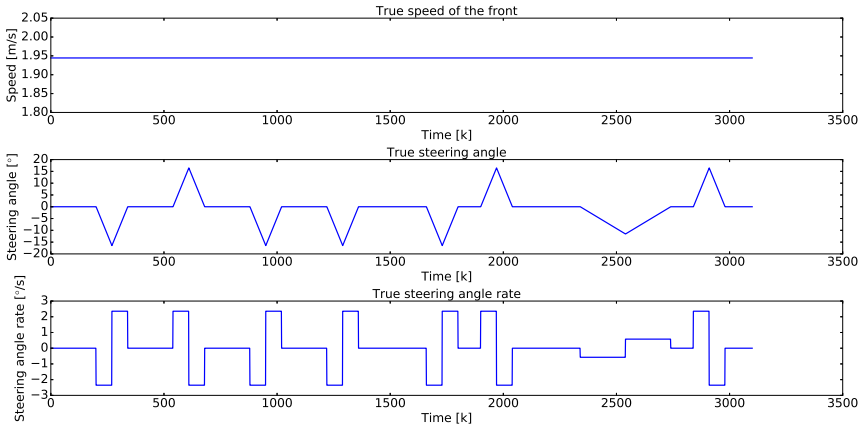


Figure C.1: Simulated ground truth speed, steering angle and steering angle rate.

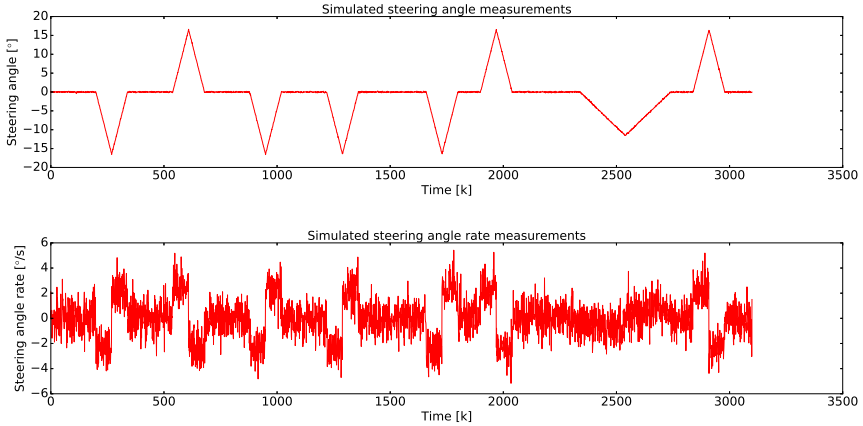


Figure C.2: Simulated noisy measurements of the steering angle and steering angle rate.

D

Results of Simulated Navigation Methods

Additional plots of the results of each navigation method in the simulated experiments.

D.1 Navigation method 1 - GNSS Previous point

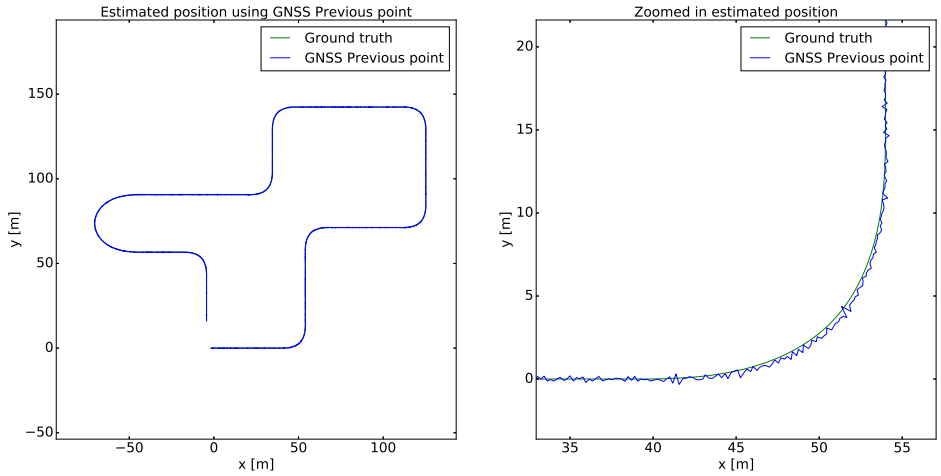


Figure D.1: Simulated ground truth position, along with the estimated position using GNSS Previous point (left) and a zoomed in part of the same plot (right).

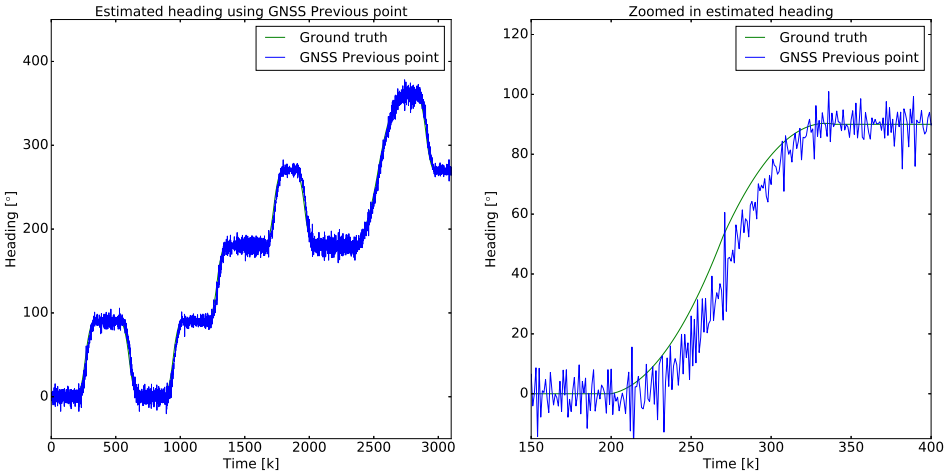


Figure D.2: Simulated ground truth heading, along with the estimated heading using GNSS Previous point (left) and a zoomed in part of the same plot (right).

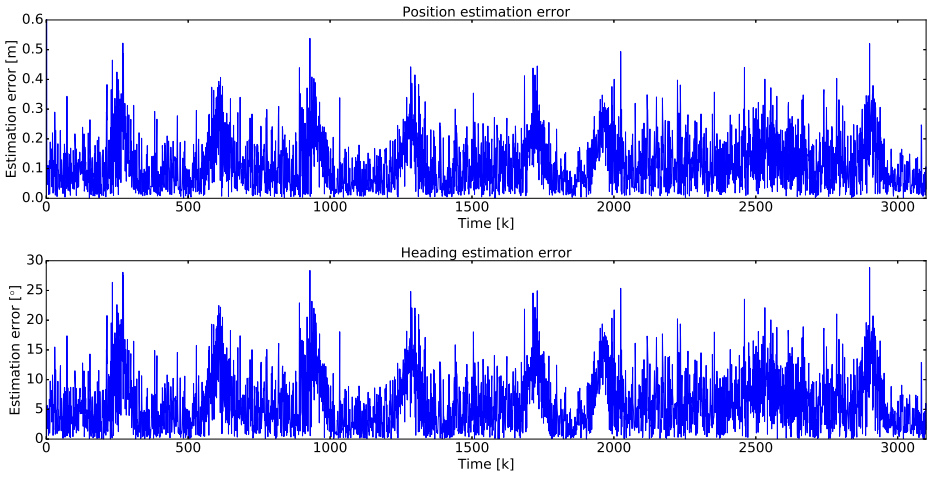


Figure D.3: The absolute value of the estimation error for position (upper) and heading (lower) using GNSS Previous point.

D.2 Navigation method 2 - EKF 1 GNSS

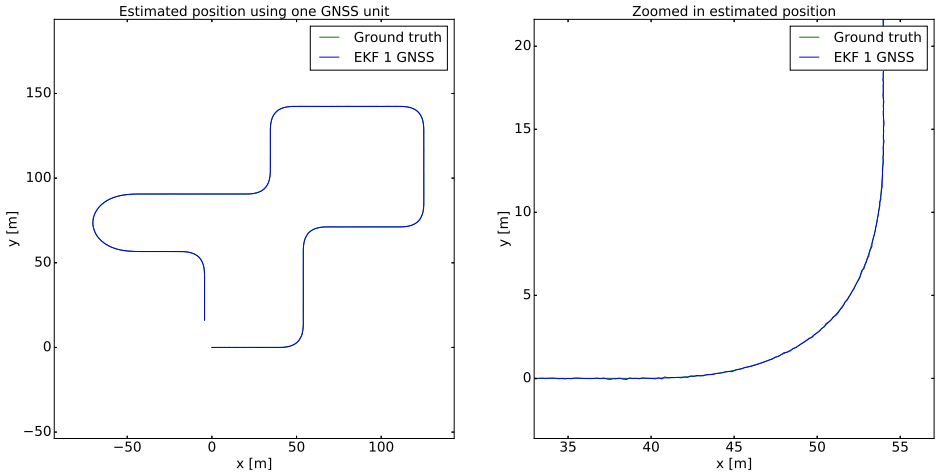


Figure D.4: Simulated ground truth position, along with the estimated position using EKF 1 GNSS (left) and a zoomed in part of the same plot (right).

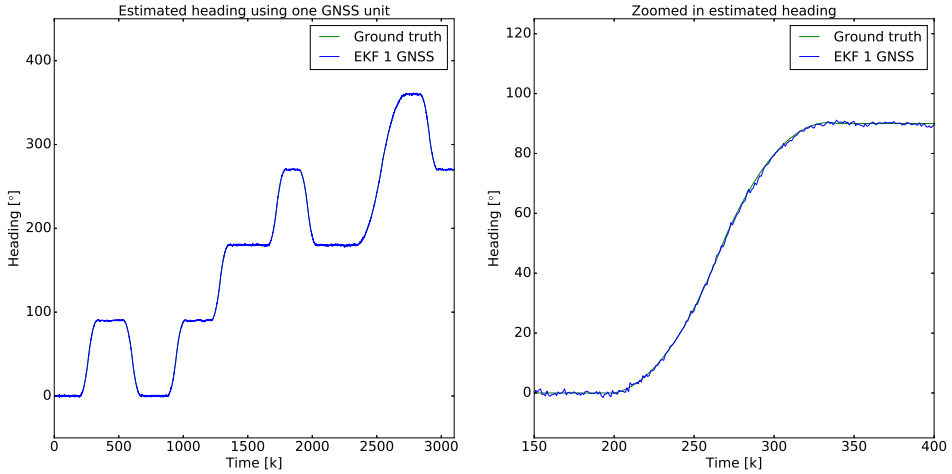


Figure D.5: Simulated ground truth heading, along with the estimated heading using EKF 1 GNSS (left) and a zoomed in part of the same plot (right).

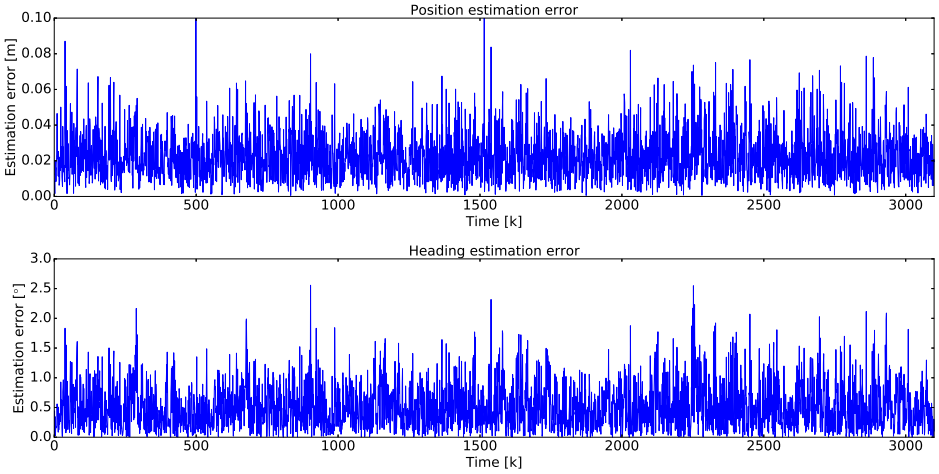


Figure D.6: The absolute value of the estimation error for position (upper) and heading (lower) using EKF 1 GNSS.

D.3 Navigation method 3 - EKF 2 GNSS

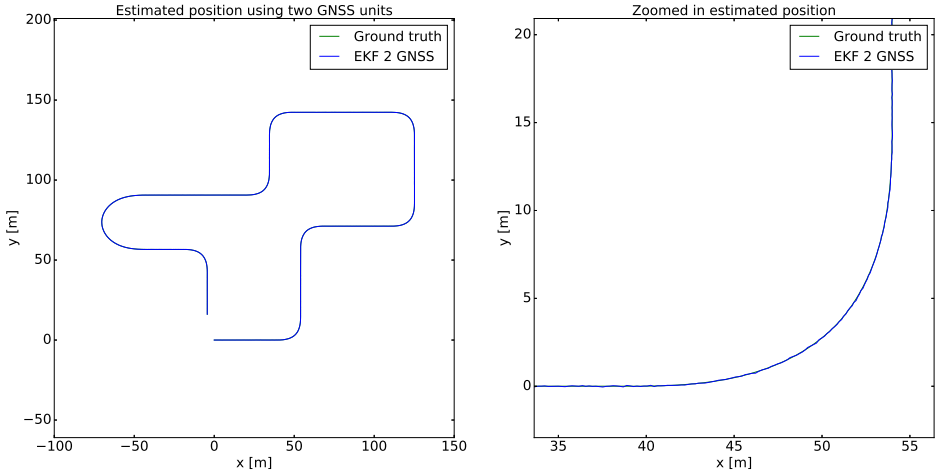


Figure D.7: Simulated ground truth position, along with the estimated position using EKF 2 GNSS (left) and a zoomed in part of the same plot (right).

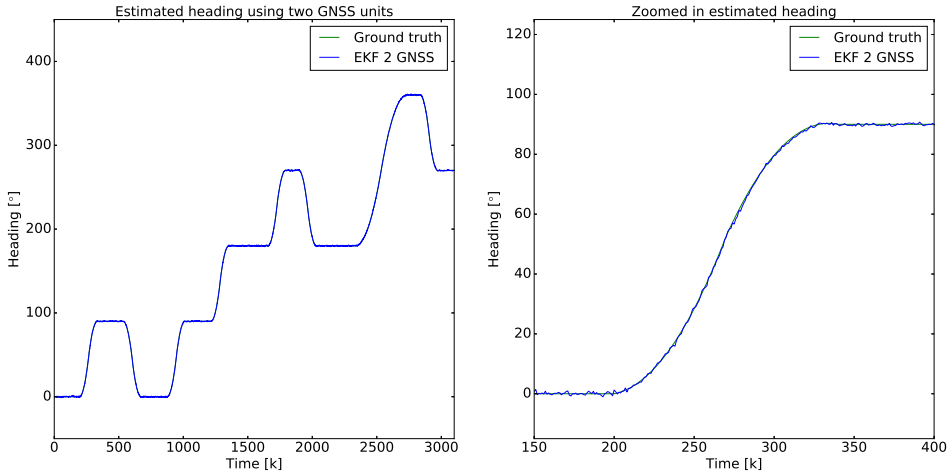


Figure D.8: Simulated ground truth heading, along with the estimated heading using EKF 2 GNSS (left) and a zoomed in part of the same plot (right).

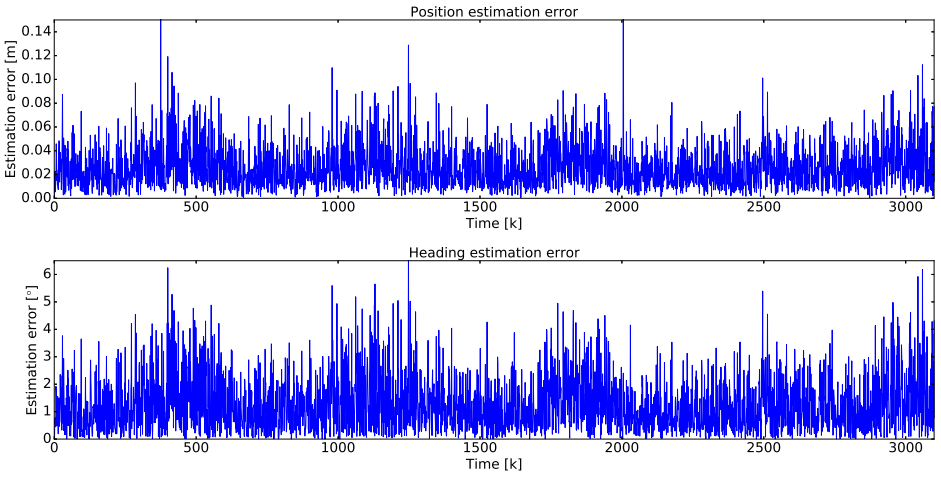


Figure D.9: The absolute value of the estimation error for position (upper) and heading (lower) using EKF 2 GNSS.

D.4 Navigation method 4 - EKF GNSS-IMU-mag

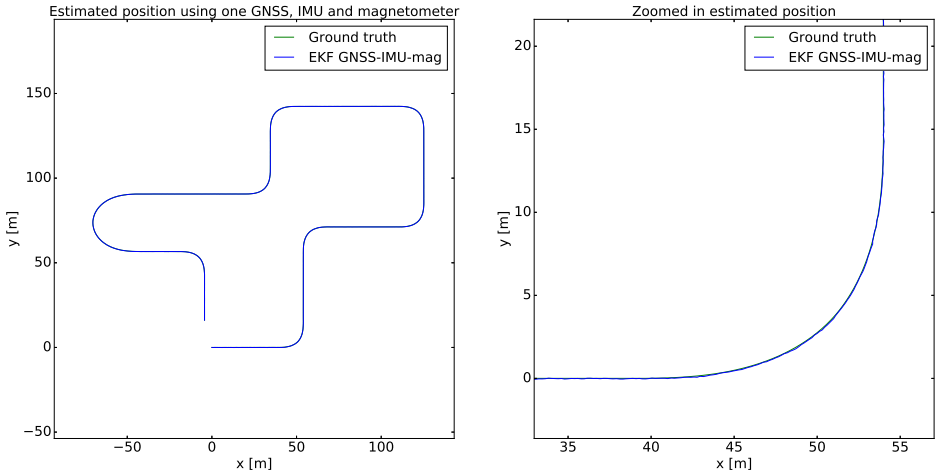


Figure D.10: Simulated ground truth position, along with the estimated position using EKF GNSS-IMU-mag (left) and a zoomed in part of the same plot (right).

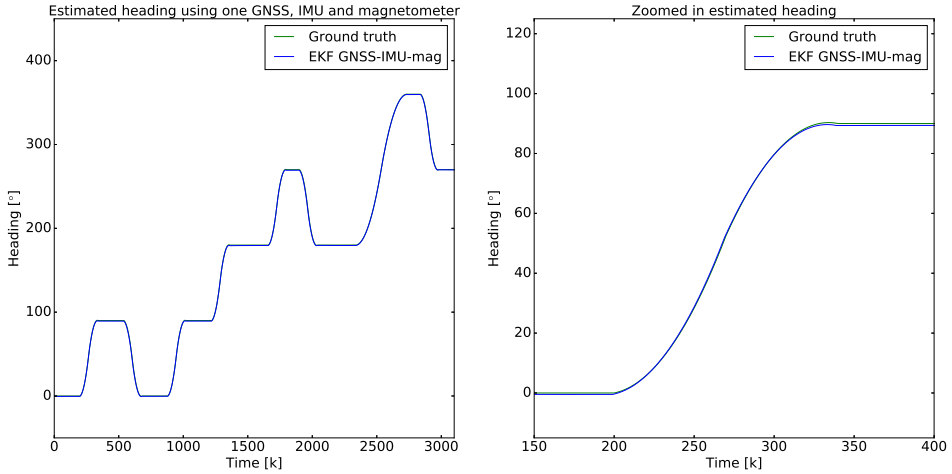


Figure D.11: Simulated ground truth heading, along with the estimated heading using EKF GNSS-IMU-mag (left) and a zoomed in part of the same plot (right).

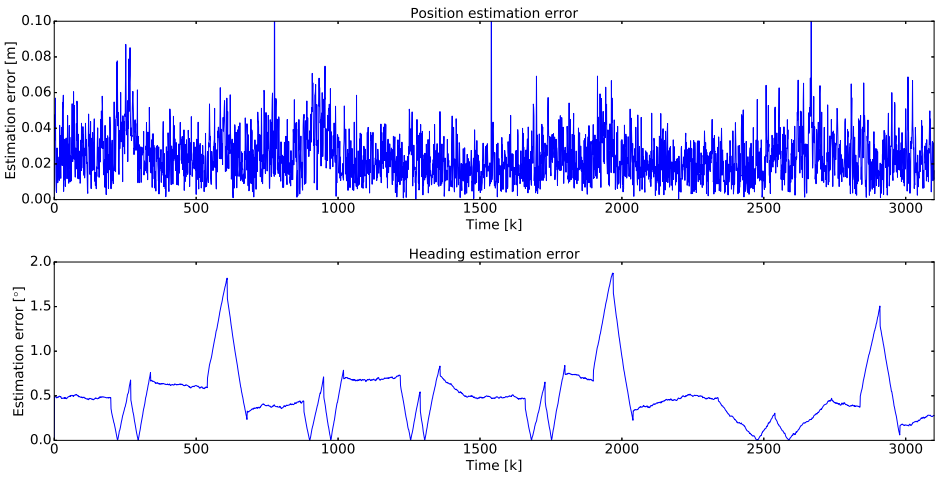


Figure D.12: The absolute value of the estimation error for position (upper) and heading (lower) using EKF GNSS-IMU-mag.