Lukas Neubauer, Bsc

Absolute Position Estimation with GPS/INS Sensor Fusion

MASTER'S THESIS

to achieve the university degree of

Master of Science

Master's degree programme: Electrical Engineering

submitted to

Graz University of Technology

Tutor

Univ.-Doz. ,Dipl.-Ing. ,Dr. techn.
Daniel Watzenig

Institute of Electrical Measurement
and Measurement Signal Processing

Graz, September 2015

Kurzfassung

Absolute Positionsbestimmung ist heutzutage global und zu jeder Zeit verfügbar, dank eines globalen Navigationssatellitensystems, wie zum Beispiel dem Globalen Positionsbestimmungssystem (GPS). Aufgrund einiger Fehlerquellen und anderer Einflüsse ist dessen Genauigkeit limitiert, beziehungsweise kann es zu einer totalen Signalblockierung kommen. Sensor Fusion kann eingesetzt werden, um eine bessere Genauigkeit und erhöhte Ausfallsicherheit der Navigationslösung zu erreichen. Eine der meist angewendeten Methoden ist die Integration eines inertialen Navigationssystems (INS), gestützt durch GPS-Messungen, verarbeitet in einen Kalman-Filter.

In dieser Arbeit wurden zwei verschiedene GPS/INS Integrationsmethoden umgesetzt, die auf inertialen Sensoren, Odometriedaten und GPS-Messungen beruhen. Die Algorithmen sind so gewählt, dass der Berechnungsaufwand klein gehalten wird und die Systeme für Echtzeitanwendungen einsetzbar sind. Auch die verschiedenen Abtastraten der Sensoren wurden berücksichtigt. Ebenfalls wurde eine Analyse der inertialen Sensorfehler durchgeführt. Die Resultate werden anhand einer MATLAB Simulation präsentiert, die auf verschiedenen zuvor durchgeführten Testmessungen basiert.

Abstract

Absolute position estimation is nowadays globally and always available due to Global Navigation Satellite Systems, like the Global Position System (GPS). Because of several error sources and other influences, the accuracy is limited or the signal may be blocked completely. Sensor fusion techniques are used to achieve better accuracy and to increase reliability. The most common integration method is an inertial navigation system (INS) aided with GPS measurements, processed by a Kalman filter.

In this thesis two different GPS/INS integration methods are implemented, utilizing inertial sensors, odometer data and GPS measurements. The algorithms are designed to keep the computational complexity low and to be suitable for real time applications. The multi-rate property of the sensors is considered as well. Also an inertial sensor error analysis is performed. The results are presented by a MATLAB simulation, based on prior taken measurement data sets.

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Graz,am ....
(Unterschrift)

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis dissertation.

Graz, ...
(signature)

Acknowledgements

I would like to thank VIRTUAL VEHICLE Research Center for providing a workspace and a comfortable work environment, where I was able to perform my research.

Special thanks to my advisor Dipl. Ing. Matthias Scharrer, who guided me throughout the whole thesis and supported me with new ideas and suggestions.

I am really grateful to the work-group of navigation from TU Graz Institute of Geodesy for their advice and cooperation. They have taken the time to hear my questions and were making an effort to answer them completely.

I would also like to mention Jasper De Smet from Flanders Drive, who was the person in charge regarding the measurement setup. He extensively answered all arising questions and also made the static measurement possible, although all preliminary measurements were completed.

Finally, I would like to thank my parents and especially my girlfriend, who were very supportive, not only during the time working on this thesis, but continuously during my studies.

Graz, 11th September 2015 Lukas Neubauer

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In the last years the trend to electro mobility is constantly growing. The VIRTUAL VEHICLE Research Center is a member of the European iCOMPOSE[1] project with main focus on energy efficiency enhancement in fully electric vehicles. To perform energy management in electric vehicles, it is very important to know not only about the traveled path, but also about the range the vehicle is able to reach with the remaining capacity. One of the key parts, is the estimation of the absolute position of the vehicle.

To get the absolute position on earth, there are different Global Navigation Satellite Systems (GNSS) available. The most common nowadays is the Global Positioning Service (GPS) (Hofmann-Wellenhof, Lichtenegger, and Wasle, 2007). This service provides 3D information about the current position. As the name indicates, GPS is globally available, but not all the time. Disturbances can occur and a reliable navigation solution is no longer granted. Because of the limited accuracy, GPS alone is not adequate for every application. The disadvantages of GPS can be compensated with sensor fusion techniques. The integration of GPS with an Inertial Navigation System (INS) is therefore a typical method (Wendel, 2011).

The goal of this thesis is to implement an sensor fusion algorithm for GPS/INS integration, which improves the absolute position estimation in terms of accuracy and redundancy. The requirements on the algorithm are that it can be used in real-time application. The hardware resources are limited and many other tasks should be performed on the same automotive control unit. Therefore the computational complexity has to be kept low.

The development of the GPS/INS integration is based on prior executed test measurements, provided by partner companies of the iCOMPOSE project. The derived fusion algorithm should make use of as much measurement information as possible. The kinematic vehicle dynamics are measured with an Inertial Measurement Unit (IMU) and odometry data is available as well. The absolute position of the vehicle is measured with a GPS receiver.

[1] Integrated Control of Multiple-Motor and Multiple -Storage Fully Electric Vehicles, www.i-compose.eu

This work presents an overview on sensor fusion methods, especially for position estimation. The theory behind different navigation systems is discussed and finally two different GPS/INS integration methods, their implementation and simulation results are shown.

1.2 Structure of Thesis

Chapter 2 gives an overview about sensor fusion and state of the art position estimation methods. Different implementation architectures and their characteristics are discussed, that the choice of the chosen fusion technique is reasonable.

In Chapter 3 important navigation principles for this thesis are discussed. More specifically, Section 3.1 is dealing with GPS and the principle behind absolute position estimation. The main part is however explaining the theory behind inertial navigation. The navigation equations for the INS mechanization and the second method based on odometry data are explained there.

Before the GPS/INS integration system model can be specified, an error analysis of the used IMU sensor is performed in Chapter 4. The error model is analytically derived and later used in the Kalman filter fusion algorithm to estimate the sensor bias drift.

The final GPS/INS integration with a detailed look at the system model, measurement model and performed Kalman filtering steps is given in Chapter 5. Section 5.4 points out how different sampling rates are handled and how the measurement data is preprocessed.

Finally the simulation results are presented and discussed in detail in Chapter 6. Chapter 7 summarizes the main aspects of this thesis and gives an outlook for further research based on this work.

Chapter 2

Sensor Fusion

Sensor fusion is a method of combining signals from different sources, to obtain more or better information with less uncertainty as it would be possible with only one signal. The sources are in general sensor signals, especially in context with sensor fusion. Sometimes the term „sensor integration“ is used in the same context, but should not be mistaken with embedding sensors into an system.

The signals from multiple devices are processed in fusion algorithms. Sasiadek (2002) classifies these algorithms into three groups:

- probabilistic models
- least-squares techniques
- intelligent fusion

Among the first group are Bayesian reasoning, evidence theory or recursive operators. Examples for the second category are Kalman filtering or uncertainty ellipsoids techniques. Intelligent fusion methods are fuzzy logic or neural networks (Sasiadek, 2002).

The chosen method for this thesis is a Kalman filtering (KF) approach, which is justified in Subsection 2.2.4.

Sensor fusion itself, is often classified as one of three groups in literature, regarding the level of fused information.

- information fusion
- sensor fusion
- data fusion

The highest level, information fusion, is referred, when data cannot be represented in numerical manner. Artificial intelligence methods are used for this kind of fusion. The strict distinction between sensor and data fusion is not always possible: If fusion takes place with raw data, rather than pre-processed sensor data, one can talk of data fusion. In the other case, the notation sensor fusion is appropriate (Gustafsson, 2012).

There are many reasons to perform sensor fusion. A main reason for sensor fusion is to increase redundancy and therefore increasing the quality compared to single sensors with respect to accuracy, reliability and integrity. Another reason is to reduce costs. Low-cost sensors, combined with a reasonable sensor fusion algorithm, yield results comparable to measurements of one expensive sensor with better accuracy. This is achieved when disadvantages of one sensor are compensated by advantages of the other sensor and vice versa (Hoffman-Wellenhof, Legat, and Wieser, 2003).

The variety of application in which sensor fusion takes place nowadays is immense. A field of application where the use of multi-sensor systems increased dramatically in the last decade, is automotive industry. This is due to the rising number of Advanced Driver Assistance Systems (ADAS) and subsequently, to develop autonomous driving vehicles. To accomplish such a task, sensor fusion is an essential key feature (Maurer et al., 2015; Siebenpfeiffer, 2014).

Fusion techniques often take advantage of complementary sensor properties. This is why GNSS and INS are particularly suitable for fusion purposes, to estimate absolute position. The long term stability, but limited accuracy of GNSS and the short term accuracy of INS, are well matched complementarities. Since the INS solution is drifting with time, mainly due to bias errors, it is possible to compensate this drift with aid of an absolute position estimate, e.g. from GNSS. A further advantage is that INS provides a full high rate position, velocity and attitude solution, with the capability to overcome GNSS outages, thanks to complementary redundancy (Noureldin, Karamat, and Georgy, 2012; Wendel, 2011).

Table 2.1 summarizes some of the characteristics of GPS and INS and shows complementary properties of each other (Hoffman-Wellenhof, Legat, and Wieser, 2003).

Property	**GPS**	**INS**
navigation type	absolute	relative
working principle	satellite-based (dependent)	autonomous (independent)
accuracy (long term)	high	low
accuracy (short term)	low	high
reliability	good	very good
external disturbance	very often	not affected
availability	average	$\sim 100\%$
sampling rate	1-10Hz	50Hz-2kHz

Table 2.1: Comparison of GPS and INS properties

2.1 State of the Art

In this Section advanced fusion techniques for absolute position estimation are presented. Not only sensor fusion methods, also solutions based on highly developed GNSS technology are mentioned. The main focus is on ground based applications, like land-based vehicle, robotic or pedestrian navigation.

Global Navigation Satellite System

Today there are different GNSS Systems, such as GPS (America), GLONASS (Russia) or the upcoming Galileo (Europe). All these systems provide an absolute global location solution. The most important system today is GPS. The accuracy of GPS for civilian users is limited to several meters. In addition to this limitation, the accuracy is also degraded due to external signal disturbances like ionosphere delay, tropospheric delay or multipath errors. But there are also internal error sources like receiver/satellite clock error or receiver noise.

So called Satellite Based Augmentation Systems (SBAS) are used to improve GPS accuracy. There are different systems for each continent. The European service for example is called European Geostationary Navigation Overlay Service (EGNOS). SBAS belongs to the Differential GPS (DGPS) category and is transmitting additional information to compensate errors from the various error sources. The achieved accuracy is about 1-3 meter (Noureldin, Karamat, and Georgy, 2012);(Wendel, 2011).

DGPS itself is further categorized into Wide Area DGPS (WADGPS) and Local Area DGPS (LADGPS). An example for WADGPS is the already mentioned SBAS. The counterpart to SBAS is Ground Based Augmentation Systems (GBAS). In LADGPS, only a single reference station serves users in a near surrounding. The idea is that errors which affect the GPS receiver, are the same for the reference receiver, because of the small distance. If the exact position of the reference device is known, the occurred errors can be estimated and transmitted to compensate them in the users GPS receiver. With advanced LADGPS methods accuracy of a few millimeter is possible (Wendel, 2011).

Mansfeld (2013), Kaplan and Hegarty (2005) and Prasad and Ruggieri (2005) explain GNSS systems in more detail.

GPS/INS Sensors

There are sensors available, which are already performing sensor integration of GNSS/INS to achieve very accurate and robust navigation solution. An example of a highly accurate INS with embedded GNSS receiver for real-time applications from SBG systems, is the Apogee-N series. It supports Real Time Kinematics (RTK), *TerraStar* and *Veripos* positioning services for centimeter precision. It is suited for marine, aerospace and land applications. Even configurations for odometry data fusion are available (SBG systems, 2015).

Vision Based and Digital Information Methods

With increasing computational power, vision based methods became more relevant in the last years. The aim is to detect lane markings, landmarks or traffic signs to estimate the position of the car relative to object. Manufacturers offer Lane Departure Warning (LDW) systems based on camera or laser systems such as Light Detection and Ranging (LIDAR) to detect road lanes (Allen, 2011).

In Krzikalla et al. (2013) a project is described, which uses cameras and detailed digital map data, providing information about landmarks and traffic signs. In combination with a GNSS/INS integration it is possible to a achieve sub-meter accuracy. The idea is to reduce car accidents by transmitting the position of the vehicle with car-to-car communication. Enhanced safety is only possible, if the position of the car is well known, not only absolute, but also relative to its surrounding.

To solve such a complex task, Simultaneous Localization and Mapping (SLAM) must be performed. This means to simultaneously create a map of the surrounding and to evaluate the location in this map. In Schleicher et al. (2009), Estrada, Neira, and Tardos (2005) and Tuna et al. (2012), SLAM techniques and its performance are well discussed. An example for landmark based fusion methods can be found in Schmackers and Glasmachers (2011).

There is also a less complex method than SLAM, which uses external map information. Map-matching uses the characteristic of a vehicle to move along roads. If the vehicle is placed on a road and the navigation solution indicates a position slightly apart, the location can be corrected with information from a map database. This can be used especially in urban areas, where the GNSS signal is often blocked or affected by multipath errors (Jabbour, Bonnifait, and Cherfaoui, 2008).

For map-matching and other vision based methods, highly accurate digital maps are necessary. The company *Atlatec* for example, offers a simple plug & play solution, to build a personal high fidelity map. With such a system testing the own ADAS is possible or use the data in presented fusion algorithms (Atlatec, 2015).

Artificial Intelligence

Learning algorithms are often denoted as Artificial Intelligence (AI) methods, because they adapt to the current situation, by using old recorded data of similar situations. With such an approach, SLAM can be improved, when the traveled path is repeated (Schleicher et al., 2009).

In Wang and Gao (2007) two AI techniques, namely fuzzy logic and expert system are discussed and their performance is validated with field tests. In some applications traditional KF for low-cost Micro Electro Mechanical System (MEMS) GPS/INS is insufficient, due to hard controllable INS error drift and correct GPS data characterization. With expert knowledge of the vehicle dynamics to reduce the INS error and a fuzzy system to identify GPS degradation, a more reliable navigation solution is provided.

For a more detailed look at fuzzy Kalman filtering, expert systems or adaptive particle filtering, see Jang, Sun, and Mizutani (1997), Kandel (1991) and Asadian, Moshiri, and Sedigh (2005).

KF represents one of the best GPS/INS integration solution. Still KF operates on given dynamic models, which are crucial for system performance. El-Sheimy, Chiang, and Noureldin (2006) suggest an artificial neural network integration for INS and DGPS measurements.

All the discussed methods so far are used for outdoor navigation. With the desire to get a navigation solution in every environment, much research is done on indoor navigation in recent years. The evolved coverage of Wifi networks and the now common smartphone technology are enabling Received Signal Strength (RSS) localization approaches

(Avinash et al., 2010). Ahmad et al. (2006) use neural networks to determine the location from the collected data.

Goswami (2012) and Werner (2014) describe indoor location methods in depth. Recent PhD theses (Bauer, 2014; Dutzler, 2013) from Graz University of Technology, are also dealing with indoor smartphone navigation.

Indoor navigation is not only interesting for pedestrian or robotic application. It can also used for vehicle navigation in multi-storey car parks or environments, where GPS is not available for a long period of time. Systems that navigate to the next free parking lot are conceivable. Even indoor localization with GNSS is possible (Lachapelle, 2004).

2.2 GPS/INS Filter Design

This Section points out multiple architectures for GPS/INS integration. Characteristics of different filter designs are discussed and as a result of the advantages or disadvantages, the chosen method for this thesis is announced. Furthermore it will provide the reader with an insight into terminology that is used in literature.

2.2.1 Uncoupled System

The simplest implementation for GPS/INS fusion is the uncoupled integration, also denoted as open-loop configuration (see Subsection 2.2.4). As it can be seen in Figure 2.1, no information of the new estimated position is fed back. Since the INS solution is drifting with time, so will the estimated result aided with GPS, without further correction. The inertial data of the IMU and the measured GPS signals are pre-processed in the INS, respectively in the GPS receiver using Kalman filtering techniques (*decentralized* filtering). A simple selection scheme, where the INS solution (with last GNSS measurement as initial position) is only used during GPS outages, is given as example in Hoffman-Wellenhof, Legat, and Wieser (2003).

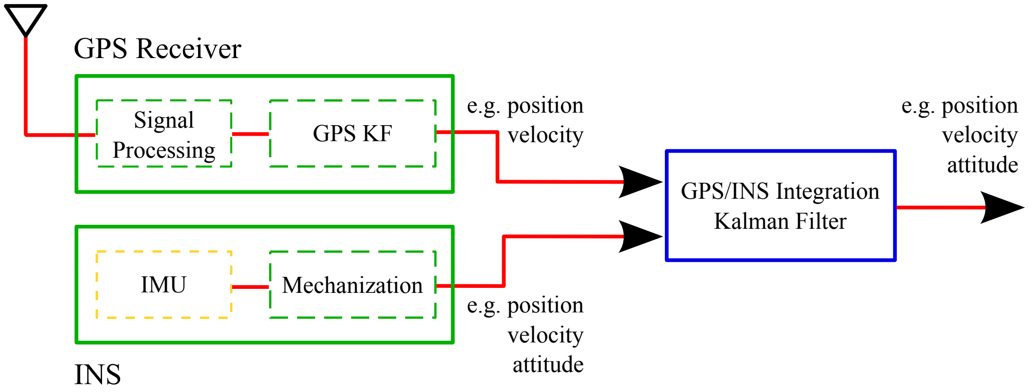
GPS Receiver
Signal Processing
GPS KF
e.g. position velocity
e.g. position velocity attitude
GPS/INS Integration Kalman Filter
IMU
Mechanization
e.g. position velocity attitude
INS

Figure 2.1: High level block diagram of an uncoupled system

2.2.2 Loosely-Coupled System

A Loosely-Coupled (LC) system is characterized by feedback from the integration filter to the INS. This feedback can be inertial bias sensor drifts or the newly estimated position to correct the INS solution. This implementation is simple and robust. The block diagram of a loosely-coupled system is shown in Figure 2.2. The main difference to a Tightly-Coupled (TC) system (see Subsection 2.2.3) is that, like in the uncoupled structure, pre-processed data is computed within the integration filter (*decentralized* filtering). Since the data from the GPS receiver is filtered internally, the assumption of uncorrelated measurement noise is violated. This can impair the performance or even cause instabilities. A disadvantage compared to the TC system is that a GPS aiding is only available when the GPS receiver is receiving a signal from more than four satellites. An advantage is that any GPS receiver can be used for integration. The estimated errors, in general the state vector of the ESKF (see Subsection 2.2.4) in an indirect LC formulation, must be set to zero after the error is corrected. This is why no KF propagation step has to be computed (see Section 5.4) (Wendel, 2011; Noureldin, Karamat, and Georgy, 2012).

In this thesis a LC structure is implemented. A TC system is not feasible with the existing measurement data and available sensors (see Chapter 6).

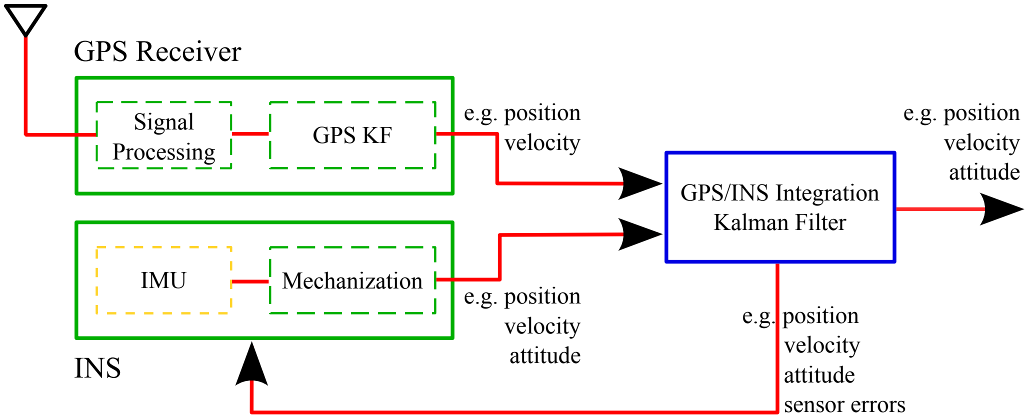
GPS Receiver
Signal Processing
GPS KF
e.g. position velocity
e.g. position velocity attitude
GPS/INS Integration Kalman Filter
IMU
Mechanization
e.g. position velocity attitude
e.g. position velocity attitude sensor errors
INS

Figure 2.2: High level block diagram of a loosely-coupled system

2.2.3 Tightly-Coupled System

This architecture is a *centralized* integration, using only one master filter. With this approach the problem of correlated measurements, due to cascaded Kalman filtering from the GPS, is eliminated. Because raw measurement data from the GPS receiver (e.g. pseudo-ranges) is processed, a GPS update is even possible with fewer than four visible satellites. A TC architecture gives almost always a more accurate and robust solution than a LC system. On the other hand, the integration complexity is much higher and a standalone GPS solution is not available. Furthermore, the GPS receiver must provide pseudo-range measurements and an input to feed back information (Figure 2.3).

There is also a *ultra-tightly coupled* or *deep* integration, with the aim to exchange as much information as possible between each block. In such a design it is possible to adjust the GPS signal processing, which can only be done with access to internal hardware (Wendel, 2011; Noureldin, Karamat, and Georgy, 2012).

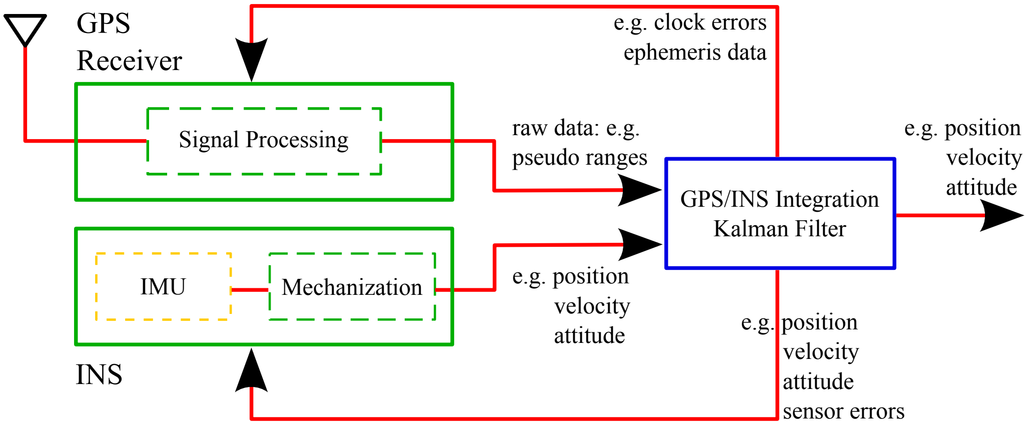

Figure 2.3: High level block diagram of a tightly-coupled system

2.2.4 Direct vs. Indirect Method

There are two formulations for GPS/INS sensor fusion algorithms. The main difference between these two, are the estimated states of the Kalman filter.

In *direct* formulation the KF estimates total quantities like position and velocity. In *indirect* formulation, the KF operates on error states, like position or velocity error. These errors are consequently corrected to get a full state estimate. The indirect KF is therefore often called Error State Kalman Filter (ESKF) (Wendel, 2011; Gustafsson, 2012).

In Munguia (2014) it is stated, that most of the GPS/INS implementation with Kalman filtering estimation technique, are using the indirect approach. This is because of the reduced computation complexity compared to the direct method. In direct formulation the Kalman filtering algorithm must be computed every time a new IMU measurement is available. In indirect formulations this is only necessary when a new GPS measurement is available. The sampling rate of the inertial data is usually higher than 50Hz, whereas GPS measurements are taken every second for example. For the indirect method, the non-linear INS navigation equations are used to derive the error propagation equations, which are then represented as a linear, time varying system, to perform Kalman filtering. The direct method interprets the navigation equations as non-linear system for the filter design. Due to the non-linear behavior of the problem, an Extended Kalman Filter (EKF) has to be used. The comparison between direct and indirect formulation shows that the direct method has a slightly better performance during GPS outages. This is mainly because the sensor noise of the IMU can be modeled correctly as measurement noise. In contrast, the IMU noise is considered as system noise with indirect formalism. Otherwise the overall system performance can be assumed as equal (Wendel and Trommer, 2001).

In Noureldin, Karamat, and Georgy (2012) and Titterton and Weston (2004) the ESKF is further divided into two groups. The *feed-forward* ESKF refers to the uncoupled design. No information is fed back to the INS. The errors are growing unlimited with time. The *feedback* ESKF corresponds to a loosely coupled system. The information of the estimated errors is fed back to the INS to correct the navigation solution. With this method, the estimated states of the ESKF, must be set to zero after the correction is done. It is stated that an ESKF in feedback configuration, equals extended Kalman filtering (Gustafsson, 2012).

(Wendel, 2011) and (Noureldin, Karamat, and Georgy, 2012) also use the terms *open-loop* or *closed-loop* for feed-forward and feedback formulation. They do not distinguish explicitly between uncoupled and loosely-coupled systems. A loosely-coupled system, in open-loop configuration, is basically uncoupled (see Figure 2.2).

Due to the reduced computational complexity the feedback ESKF (indirect approach) is chosen for further investigations within this thesis instead of the direct method.

Chapter 3

Navigation Systems

In this Chapter different navigation systems used in this thesis are explained. They are divided into two groups. The first one is a representative of a global absolute position system, namely the well known GPS. The second one, is the group of Dead Reckoning (DR) systems. Two methods, using a combination of both systems, are used in this project.

3.1 Global Positioning System

GPS is the American GNSS for absolute position estimation on earth. The development began in the early 1970's and was declared operational in 1995. The system was first called Navigation System for Timing and Ranging (NAVSTAR) for military purposes. Later on the system was renamed in GPS for civilian applications. In the beginning the signal quality was artificially reduced for civil uses, but since 2000, the jamming transmitters are off-line.

The service is segmented into three parts:

- space segment
- control segment
- user segment

The *space segment* consists of minimal 24 satellites in the near earth orbit. With a special constellation it is ensured that always, at least four satellites are visible (with a clear view of the sky) for the GPS receiver. The satellites transmitting coded radio signals, which are processed in the GPS receiver to estimate the current position.

The *control segment* is responsible for monitoring the state of the system. If necessary they have to correct orbit trajectories and watch over clock integrity. There is one master station controlled by the US.

With *user segment*, all military or civilian GPS receivers are meant. Today there are countless application for GPS. In the beginning the most intended use was car navigation. Nowadays nearly every smartphone has build-in a GPS receiver. Even digital cameras are using GPS for geo-tagging.

3.1.1 Working Principle

The GPS position estimation is based on a triangulation technique. With the knowledge of time, the signal from a satellite traveled until it reached the GPS receiver and its velocity, the distance to that satellite can be calculated. These are so called pseudo-ranges.

Figure 3.1 is an idealized example of a two dimensional triangulation problem. If there are two satellites, i.e. two ranges, there are two intersection points of the circles. Only with a additional satellite a unique solution can be found. In three dimensions, also three satellites are theoretically sufficient, because the other possible solution would be in space and is therefore not plausible.

This is only an simple explanation for triangulation and not usable in practice. To get the time passed during transmission and reception, there would have to be a time synchronization between each receiver and the satellite clocks which is not feasible. The problem of the uncertain time measurement can be solved if a signal from a fourth satellite is received. This is why GPS needs at least four satellites to provide a reliable navigation solution.

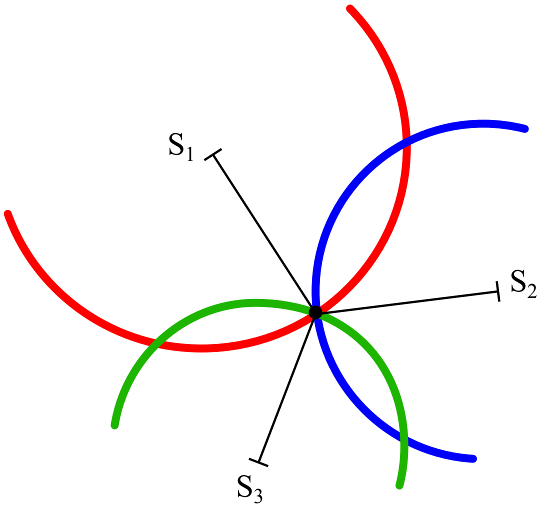

Figure 3.1: Two dimensional triangulation example

In Noureldin, Karamat, and Georgy (2012) it is stated that there is also the method of carrier-phase measurement to calculate pseudo-ranges. The Doppler effect can be used to calculate the velocity of the user.

3.1.2 Error Sources

As already mentioned, the accuracy and availability of GPS is limited. There are many error sources that have negative impact on the received satellite radio signal. There are not only external disturbances, but also internal ones. Here are some of the typically encountered errors of a receiver from Noureldin, Karamat, and Georgy (2012).

Multipath Errors
Multipath errors occur primary in urban environments, where the satellite signal is reflected from various surfaces. The signal reaches the receiver from different paths, not only a direct line of sight. The signals with longer path are delayed and cause an position error. With carrier phase measurements, multipath errors can dramatically be reduced.

Atmospheric Delay
The transmission time is affected by the atmosphere of the earth. Depending on the angle of incidence, the signal has to travel a longer way through the atmosphere, changing the transit time of the GPS signal. The ionosphere delay depends on solar activity. There is also a tropospheric delay due to a decreased speed relative to free space.

Clock Errors
Over time, the satellite clock is drifting away from the GPS time. The control segment estimates the drift and transmit correction parameters to the satellites. The satellites broadcast them to the GPS receivers via the navigation message. The GPS receiver itself has a clock bias. With the pseudo-range measurement from a fourth satellite (see Subsection 3.1.1), the bias can be estimated.

Receiver Noise
A random measurement noise caused by the electronics of a GPS receiver. It causes an incorrect measurement of the transit time. The effect can be reduced with carrier phase measurements.

A much more detailed explanation of GNSS and satellite based navigation is given in Prasad and Ruggieri (2005), Hofmann-Wellenhof, Lichtenegger, and Wasle (2007), and Mansfeld (2013).

3.2 Dead Reckoning

DR is a navigation principle, which uses sensors to measure the kinematic of a vehicle to calculate its position. This is only possible if the initial position and other initial states are known. In that sense, DR is a relative navigation system compared to GNSS. The new calculated position depends on the previous one and so on.

Based on the used sensors, DR is separated into two groups. *Inertial Navigation* uses inertial sensors (accelerometers and gyroscopes) and *Odometry* utilizes data from wheel- and steering rotations. One of the implemented methods in this project uses wheel-speed measurements in combination with inertial sensors.

This Section gives an overview of inertial navigation, but only aspects that are relevant for this thesis are discussed. The here presented equations are required for GPS/INS integration in Chapter 5.

3.2.1 Inertial Navigation

Inertial navigation is an autonomous navigation method that provides information about attitude, velocity and position. Since inertial navigation works according to the DR principle, the vehicle's current position is determined with knowledge of previous position and sensor measurements. The sensor measurements for inertial navigation are accelerations and angular rates by a triad of accelerometers and gyroscopes, measuring the six Degrees-of-Freedom (6-DOF). The idea is to integrate the accelerations to calculate velocities and a second integration leads to position. The angular rates are processed to achieve attitude in terms of roll, pitch and yaw angle. To start the integration, initial conditions for attitude, velocity and position are needed.

It is important that the mentioned integration is not a simple, separate integration of the 6 DOF measurements. They are coupled through differential equation to get attitude, velocity and position. The IMU measurements are taken in the body-frame of the vehicle. For navigation purposes, velocity and position are only useful in the navigation-frame, also called local-level frame. A transformation has to be performed. Coordinate frames are discussed in Subsection 3.2.2.

The mechanism that resolves the IMU measurements into attitude, velocity and position in the desired coordinate frame, is called mechanization process (see Subsection 3.2.2). Especially for a strapdown (SD) system it is called strapdown-algorithm. It is named SD because the IMU is strapped down onto the body of the moving platform. That is why the IMU measurements are in body-frame coordinates.

The first inertial navigation systems were gimballed platform systems. In these systems the inertial sensors are mounted on a platform, which always remains aligned with the navigation frame. Torque motors rotate the platform in response to the gyroscope measurements. The orientation angles (Euler angles) of the platform can be directly picked off from the gimbals, without further integration. The accelerations have to be double integrated to get the position. Gimballed systems are mechanically complex, expensive and bigger than similar strapdown systems. Strapdown systems have a higher computation complexity but are now the dominant type, due to low power usage, flexibility, cheap in purchase and lightweight construction thanks to MEMS technology.

The mechanization process is performed in the so called INS. The IMU is a module of the INS and contains the Inertial Sensor Assembly (ISA). The ISA only consists of an assembly of 3 orthogonal accelerometers and gyroscopes for all body-frame axis. The different parts of an INS can be seen in Figure 3.2. In connection with a strapdown system it is also called Strapdown Inertial Navigation System (SDINS) (Noureldin, Karamat, and Georgy, 2012).

Since integration steps are performed, a bias offset in the measurements is crucial, because it leads to a drift over time of the solution. An acceleration offset is causing an second order error for position calculation. In Skog and Handel (2009) it is stated that a gyroscope bias error will be the most influential part of velocity and position error. Since the measured specific forces from the accelerometers are resolved into the navigation frame, using the attitude information calculated from angular rates, a third order position error is the result. Another explanation why attitude errors are fatal, is incorrect gravity compensation. If there is an error in the transformation of specific forces to the navigation frame, the gravity can not be corrected properly. The earth gravity is much bigger than the occurring vehicle accelerations under normal vehicle dynamics. This is why false compensation leads to a large acceleration error, causing the INS solution to drift with time (Noureldin, Karamat, and Georgy, 2012).

There is so much theory about inertial navigation that can not be covered in this thesis. Excellent books that are the basis for this project are (Titterton and Weston, 2004), (Noureldin, Karamat, and Georgy, 2012). A detailed explanation about strapdown algorithms and inertial navigation can be found there.

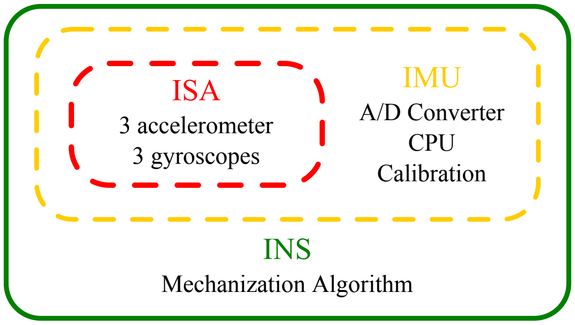

Figure 3.2: Modules of an INS

3.2.2 INS Mechanization

The implemented mechanization equations are presented in this Section. Before the final SD algorithm is declared, some conventions and definitions about coordinate frames, attitude representation and the used world model must be made.

Coordinate Frames

In inertial navigation various coordinate frames are used. They are used to express the position of a point in relation to a reference point. The four relevant frames are:

- The **body-frame (b-frame)** is fixed with the moving platform. The origin is within the vehicle. It is very important that the sensitive axes of the IMU coincide with the body-frame. If the IMU has an other internal orientation or is mounted otherwise, the axes must be assigned correctly before computation. Precise orthogonal sensitive axes are a prerequisite. The IMU measurements are taken in the b-frame and have therefore a *b-superscript* ($\boldsymbol{f}^b, \boldsymbol{\omega}^b$). The body-frame for the SDINS is shown in Figure 3.3. Be aware that the Reduced Inertial Sensor System (RISS) has a different b-frame (see Figure 3.6).

- The center of the **navigation-frame (n-frame)** and the b-frame are the same. The axes points to true north, east and down (NED) direction, whereas down means parallel to the gravity vector. There is also a east, north, up (ENU) convention. Both are right-handed coordinate frames. The NED-frame is used for SDINS and the ENU-frame for RISS mechanization. For navigation applications it is useful to express velocities in the n-frame. The *n-superscript* notation indicates the reference frame. If a component of the three dimensional velocity vector in the NED-frame is selected, a corresponding subscript is used. For example, $v^n_{eb,N}$ is the north-velocity in the n-frame. Noureldin, Karamat, and Georgy (2012) is using the term local-level-frame with a *l-superscript* instead of n-frame.

- The **inertial-frame (i-frame)** is centered in the middle of the earth ellipsoid. The z-axis is along axis of earth rotation. The x-axis is in the equatorial plane pointing towards the vernal equinox (Noureldin, Karamat, and Georgy, 2012). The y-axis completes a right-handed system. The i-frame is defined stationary in space. All inertial sensors produce measurements in the b-frame relative to the i-frame. The gyroscope measurement subscript notation $\boldsymbol{\omega}^b_{ib}$ is taking account of this fact.

- The **earth-centered earth-fixed frame (e-frame)** is almost equal to the i-frame. The center and the z-axis are the same but rotates along with the earth. The x-axis passes through the intersection of equatorial plane and Greenwich meridian. The term $\boldsymbol{\omega}^e_{ie} = [0, 0, \omega_E]^T$ represents the earth rotation with respect to the i-frame, resolved in the e-frame. There are two common representation of position in the e-frame. Rectangular and geodetic coordinates. Rectangular coordinates are like traditional Cartesian coordinates and therefore hard to interpret for navigating on a spherical shaped surface. Geodetic coordinates are way more intuitive for navigation application. They are defined as Latitude φ, Longitude λ and Height h (LLH) (Noureldin, Karamat, and Georgy, 2012).

World Geodetic System 1984

The earth is not a perfect sphere, but somehow the shape of earth has to be approximated. This is done by the World Geodetic System announced in 1984 (WGS84). It models the earth as a ellipsoid that is made by rotating an ellipse about its minor axis. The axis is coincident with the rotational axis of earth and the center is earth's center of mass (Noureldin, Karamat, and Georgy, 2012).

This approximation is sufficient for most navigation applications. The WGS84 is the common reference model nowadays and is also used in most GPS receivers. It is crucial that the absolute positioning system and the inertial navigation system are using the same reference world model.

Table 3.1 lists the most important parameters of the WGS84 (Wendel, 2011).

The two radii of curvature are of particular interest. As it can be seen, the radii are functions of the position on earth, namely from the latitude φ. R_e is the radius in east-west direction and R_n in north-south direction. With knowledge of the n-frame velocity and the radii of curvature, the position on earth in LLH coordinates can be calculated through integration.

name	symbol	value
semi-major axis	a	6378137 m
semi-minor axis	b	6356752.3142 m
flatness	f	0.003352810664747
eccentricity	e	0.0818191908426
meridian radius of curvature	R_n	$a\frac{1-e^2}{(1-e^2sin^2\varphi)^{3/2}}$
normal radius of curvature	R_e	$\frac{a}{\sqrt{1-e^2sin^2\varphi}}$
earth's rotation rate	ω_E	$7.292115 \times 10^{-5}\frac{rad}{s}$

Table 3.1: WGS84 World Model parameter

Gravity Model

Because of the shape of earth, the gravity is not constant everywhere on the surface. It depends on the latitude φ and on the height h over the WGS84 ellipsoid. The gravity vector combines the effect of gravitation and centripetal acceleration due to earth's rotation. The normal gravity vector in the n-frame has only one entry in the third component (Equation 3.1), because it coincides with the ellipsoidal normal. Note that the normal gravity vector refers to the WGS84 ellipsoid and that the true gravity vector deviates from the normal gravity vector by the deflection of the vertical.

$$\boldsymbol{g}^n = [0, 0, \gamma(\varphi, h)]^T \tag{3.1}$$

$$\gamma(\varphi, h) \approx \gamma(\varphi)\left(1 - \frac{2}{a}\left(1 + f + m - 2fsin^2\varphi\right)h + \frac{3}{a^2}h^2\right) \tag{3.2}$$

where

$$m = \frac{\omega_E^2 a^2 b}{GM_E} \tag{3.3}$$

and

$$\gamma(\varphi) = \frac{a\gamma_a cos^2\varphi + b\gamma_b sin^2\varphi}{\sqrt{a^2cos^2\varphi + b^2sin^2\varphi}} \tag{3.4}$$

Equation (3.4) is called *Formula of Somigliana* and is only a function of latitude φ. In Equation (3.2) it is extended to get a height dependency. This gravity model is typically used in geodesy applications (Hoffman-Wellenhof and Moritz, 2006).

The parameters *a,b,f* and ω_E refer to the WGS84 world model (Table 3.1). The other parameters of Equations (3.4) and (3.3) are listed in Table 3.2.

name	symbol	value
gravity constant × earth mass	GM_E	$3.986004418 \times 10^{14} \frac{m^3}{s^2}$
equatorial earth acceleration	γ_a	$9.7803267715 \ \frac{m}{s^2}$
polar earth acceleration	γ_b	$9.8321863685 \ \frac{m}{s^2}$

Table 3.2: Gravity model parameter

Attitude Representation

The main methods for attitude representation are in terms of Euler angles, direction cosines or quaternions. The three methods have different characteristics and vary in computational complexity. In this thesis direction cosines are not treated.

Solving the mechanization equations requires the parameterization of the rotation matrix $\boldsymbol{C}_b^n$, also labeled as Direction Cosine Matrix (DCM), especially used in aerospace application. The DCM is needed to resolve the measured specific forces from the b-frame into the n-frame.

Euler Angles
There are different ways to represent the attitude of an object. The most intuitive one is in terms of Euler angles. The three angles *roll* ϕ, *pitch* θ and *yaw* ψ represent the angles of a rotation about the x-,y- and z-axis in the n-frame. The relation between angular rates and Euler angles is given in Equation (3.5 - 3.7).

$$\dot{\phi} = (\omega_{nb,y}^b sin\phi + \omega_{nb,z}^b cos\phi) tan\theta + \omega_{nb,x}^b \tag{3.5}$$

$$\dot{\theta} = \omega_{nb,y}^b cos\phi - \omega_{nb,z}^b sin\phi \tag{3.6}$$

$$\dot{\psi} = (\omega_{nb,y}^b sin\phi + \omega_{nb,z}^b cos\phi)/cos\theta \tag{3.7}$$

with the angular rate vector

$$\boldsymbol{\omega}_{nb}^b = [\omega_{nb,x}^b, \omega_{nb,y}^b, \omega_{nb,z}^b]^T \tag{3.8}$$

The DCM can be written as a function of Euler angles.

$$\boldsymbol{C}_b^n = \begin{bmatrix} cos\theta cos\psi & -cos\phi sin\psi + sin\phi sin\theta cos\psi & sin\phi sin\psi + cos\phi sin\theta cos\psi \\ cos\theta sin\psi & cos\phi cos\psi + sin\phi sin\theta sin\psi & -sin\phi cos\psi + cos\phi sin\theta sin\psi \\ -sin\theta & sin\phi cos\theta & cos\phi cos\theta \end{bmatrix} \tag{3.9}$$

The advantage of attitude computation with Euler angles is that only three differential equations are needed. The result is in Euler angles and the initialization can be done directly with Euler angles too. On the other hand, the differential equations are nonlinear and therefore hard to solve. There is also a singularity in the solution for a pitch of ±90 degrees (see Equation 3.7), which is called gimbal-lock. Thus, this representation is not suitable for some applications.

Quaternions
The quaternion representation offers many advantages and is the most popular method nowadays (Kuipers, 1999). The idea of quaternions is based on the Euler theorem. It states that a transformation from one coordinate frame into another can be represented by a single rotation about a vector $\boldsymbol{\Theta}$ with respect to the reference frame. The magnitude of the rotation is Θ and $\frac{\Theta_x}{\Theta}$, $\frac{\Theta_y}{\Theta} and \frac{\Theta_z}{\Theta}$ are direction cosines of the rotation axis (Noureldin, Karamat, and Georgy, 2012).

A quaternion is defined as a four parameter vector

$$\boldsymbol{q}_b^n = \begin{bmatrix} q_a \\ q_b \\ q_c \\ q_d \end{bmatrix} = \begin{bmatrix} cos(\frac{\Theta}{2}) \\ \frac{\Theta_x}{\Theta} sin(\frac{\Theta}{2}) \\ \frac{\Theta_y}{\Theta} sin(\frac{\Theta}{2}) \\ \frac{\Theta_z}{\Theta} sin(\frac{\Theta}{2}) \end{bmatrix} \tag{3.10}$$

with following relation

$$\Theta = \sqrt{\Theta_x^2 + \Theta_y^2 + \Theta_z^2} \tag{3.11}$$

and the constraint

$$\sqrt{q_a^2 + q_b^2 + q_c^2 + q_d^2} = 1 \tag{3.12}$$

The advantages of quaternions are no gimbal-lock and a reduced computational complexity, because only four linear differential equations are needed do perform the attitude computation. With an additional equation compared to Euler representation, there is now one degree of freedom, described in Equation (3.12). This indicates that only three independent quaternion parameter are sufficient to describe the rotation of a rigid body (Noureldin, Karamat, and Georgy, 2012). The constrained is used during attitude computation to normalize the quaternion and assure the orthogonality of the DCM.

The quaternion differential equation is shown in Equation (3.13). The performed mathematical operation is a quaternion multiplication. Equation (3.14) is similar to (3.13), only in matrix notation.

$$\dot{\boldsymbol{q}}_b^n = \frac{1}{2}\boldsymbol{q}_b^n \bullet \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{nb}^b \end{bmatrix} \tag{3.13}$$

$$\dot{\boldsymbol{q}}_b^n = \frac{1}{2}\boldsymbol{Q}_b^n \overline{\boldsymbol{\omega}}_{nb}^b = \frac{1}{2} \begin{bmatrix} q_a & -q_b & -q_c & -q_d \\ q_b & q_a & -q_d & q_c \\ q_c & q_d & q_a & -q_b \\ q_d & -q_c & q_b & q_a \end{bmatrix} \begin{bmatrix} 0 \\ \omega_{nb,x}^b \\ \omega_{nb,y}^b \\ \omega_{nb,z}^b \end{bmatrix} \tag{3.14}$$

The DMC can also be calculated from quaternions:

$$\boldsymbol{C}_b^n = \begin{bmatrix} (q_a^2 + q_b^2 - q_c^2 - q_d^2) & 2(q_b q_c - q_a q_d) & 2(q_b q_d + q_a q_c) \\ 2(q_b q_c + q_a q_d) & (q_a^2 - q_b^2 + q_c^2 - q_d^2) & 2(q_c q_d - q_a q_b) \\ 2(q_b q_d - q_a q_c) & 2(q_c q_d + q_a q_b) & (q_a^2 - q_b^2 - q_c^2 + q_d^2) \end{bmatrix} \tag{3.15}$$

A disadvantage when using quaternions is that Euler angles are not directly available. For visualization of the attitude, Euler angles are the most intuitive representation. One way to convert quaternions into Euler angles is to construct the DCM with Equation (3.15) and then calculate Equation (3.16-3.18).

$$\phi = arctan\left(\frac{C^n_{b,32}}{C^n_{b,33}}\right) \tag{3.16}$$

$$\theta = arcsin\left(-C^n_{b,31}\right) \tag{3.17}$$

$$\psi = arctan\left(\frac{C^n_{b,21}}{C^n_{b,11}}\right) \tag{3.18}$$

To formulate the integration initial condition in terms of quaternion $\boldsymbol{q}_0$ from Euler angles, Equations (3.19-3.22) is used.

$$q_a = cos\frac{\phi}{2}cos\frac{\theta}{2}cos\frac{\psi}{2} + sin\frac{\phi}{2}sin\frac{\theta}{2}sin\frac{\psi}{2} \tag{3.19}$$

$$q_b = sin\frac{\phi}{2}cos\frac{\theta}{2}cos\frac{\psi}{2} - cos\frac{\phi}{2}sin\frac{\theta}{2}sin\frac{\psi}{2} \tag{3.20}$$

$$q_c = cos\frac{\phi}{2}sin\frac{\theta}{2}cos\frac{\psi}{2} + sin\frac{\phi}{2}cos\frac{\theta}{2}sin\frac{\psi}{2} \tag{3.21}$$

$$q_d = cos\frac{\phi}{2}cos\frac{\theta}{2}sin\frac{\psi}{2} - sin\frac{\phi}{2}sin\frac{\theta}{2}cos\frac{\psi}{2} \tag{3.22}$$

The *Aerospace Toolbox* in MATLAB is used for implementation, because it provides functions to convert freely between Euler angles and quaternions. See Appendix A for more useful relations regarding attitude representation and quaternions. A detailed explanation of quaternions can be found in Kuipers (1999) and Sola (2015).

Strapdown Algorithm

The SD algorithm is the mechanization process of converting the IMU measurements into 3D position, velocity and attitude. The triad of gyroscopes and accelerometers of an IMU, mounted on a moving platform measures rotation rates $\boldsymbol{\omega}^b_{ib}$ and specific forces $\boldsymbol{f}^b_{ib}$ about the body axes with respect to the inertial frame. The accelerometers measure specific forces rather than accelerations, because earth's gravity is superimposed. For compensation a gravity model is needed (Section 3.2.2).

In Figure 3.3 the body-frame axes and the direction of the positive rotation rate is shown. The rotation rate is positive in clockwise orientation along the axis. The IMU used in thesis is mounted in the vehicle that the sensitive axes (see Figure 3.4) align with the assumed b-frame.

For attitude computation, the measured IMU rotation rates $\boldsymbol{\omega}^b_{ib}$ have to be corrected due to Coriolis effect. Equations (3.23-3.25) are used for implementation. The two rotation rate vectors are earth's rotation rate vector (3.24) and transport rate vector (Equation 3.25), given in the n-frame. Therefore they have to be resolved into the b-frame using the rotation matrix $\boldsymbol{C}^b_n$ (Equation 3.23). The sum of the two vectors is forming a

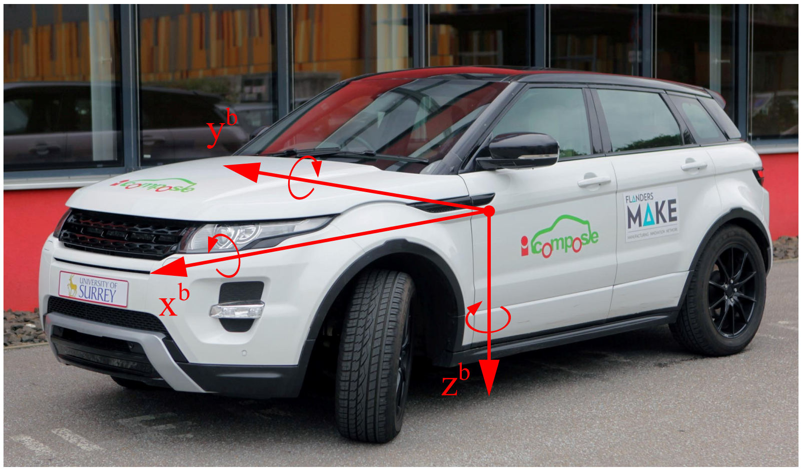

Figure 3.3: Body-frame of the vehicle for SDINS

new rotation vector (see Appendix A) like in Equation (3.26). This relation will be used in Chapter 5. Now all components are know to compute Equation (3.14).

$$\boldsymbol{\omega}^b_{nb} = \boldsymbol{\omega}^b_{ib} - \boldsymbol{C}^b_n(\boldsymbol{\omega}^n_{ie} + \boldsymbol{\omega}^n_{en}) \tag{3.23}$$

$$\boldsymbol{\omega}^n_{ie} = \begin{bmatrix} \omega_E cos\varphi \\ 0 \\ -\omega_E sin\varphi \end{bmatrix} \tag{3.24}$$

$$\boldsymbol{\omega}^n_{en} = \begin{bmatrix} \dfrac{v^n_{eb,E}}{R_e - h} \\ -\dfrac{v^n_{eb,N}}{R_n - h} \\ -\dfrac{v^n_{eb,E} tan\varphi}{R_e - h} \end{bmatrix} \tag{3.25}$$

$$\boldsymbol{\omega}^n_{in} = \boldsymbol{\omega}^n_{ie} + \boldsymbol{\omega}^n_{en} \tag{3.26}$$

A correction of the measured specific forces due to Coriolis acceleration and the gravity compensation is done in Equation (3.27). It is to mention that the cross product can be rewritten into a matrix multiplication by using skew symmetric matrices. Equation (3.28) is therefore equivalent to Equation (3.27). Appendix A provides more information about skew symmetric matrices.

$$\dot{\boldsymbol{v}}^n_{eb} = \boldsymbol{C}^n_b \boldsymbol{f}^b_{ib} - (2\boldsymbol{\omega}^n_{ie} + \boldsymbol{\omega}^n_{en}) \times \boldsymbol{v}^n_{eb} + \boldsymbol{g}^n \tag{3.27}$$

$$\dot{\boldsymbol{v}}^n_{eb} = \boldsymbol{C}^n_b \boldsymbol{f}^b_{ib} - (2\boldsymbol{\Omega}^n_{ie} + \boldsymbol{\Omega}^n_{en})\boldsymbol{v}^n_{eb} + \boldsymbol{g}^n \tag{3.28}$$

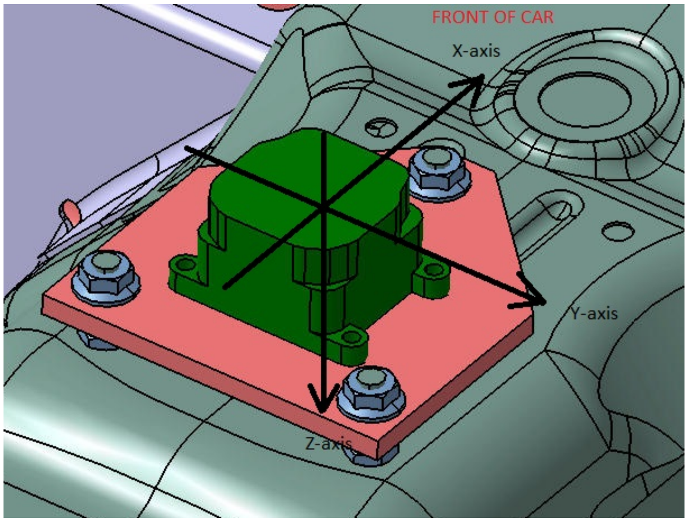

Figure 3.4: IMU mounting in test vehicle

The position representation is done in LLH coordinates. If the NED velocity $\boldsymbol{v}_{eb}^n = [v_{eb,N}^n, v_{eb,E}^n, v_{eb,D}^n]^T$ is known, using the radii of curvature, the position is calculated through Equations (3.29-3.31). The matrix notation of the differential equations for the vector $\boldsymbol{p}_{LLH}^n = [\varphi, \lambda, h]^T$ is given in (3.33).

$$\dot{\varphi} = \frac{v_{eb,N}^n}{R_n(\varphi) - h} \tag{3.29}$$

$$\dot{\lambda} = \frac{v_{eb,E}^n}{(R_e(\varphi) - h)cos\varphi} \tag{3.30}$$

$$\dot{h} = v_{eb,D}^n \tag{3.31}$$

$$\boldsymbol{D}^{-1} = \begin{bmatrix} \frac{1}{R_n(\varphi) - h} & 0 & 0 \\ 0 & \frac{1}{(R_e(\varphi) - h)cos\varphi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.32}$$

$$\dot{\boldsymbol{p}}_{LLH}^n = \boldsymbol{D}^{-1}\boldsymbol{v}_{eb}^n \tag{3.33}$$

The SD computation equations can be written in a compact vector notation. Equation (3.34) is a set of non-linear differential equations. To solve the differential equations, a simple Euler method is used for integration (Equations 3.35-3.37). Chapter 6 shows that this approximation method is sufficient with a sampling frequency of 50Hz ($\Delta t = 0.02s$).

$$\dot{\boldsymbol{x}} := \begin{bmatrix} \dot{\boldsymbol{q}}_b^n \\ \dot{\boldsymbol{v}}_{eb}^n \\ \dot{\boldsymbol{p}}_{LLH}^n \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\boldsymbol{Q}_b^n \overline{\boldsymbol{\omega}}_{nb}^b \\ \boldsymbol{C}_b^n \boldsymbol{f}_{ib}^b - (2\boldsymbol{\Omega}_{ie}^n + \boldsymbol{\Omega}_{en}^n)\boldsymbol{v}_{eb}^n + \boldsymbol{g}^n \\ \boldsymbol{D}^{-1}\boldsymbol{v}_{eb}^n \end{bmatrix} =: \boldsymbol{f}\left(\boldsymbol{x}(t)\right) \tag{3.34}$$

$$\dot{\boldsymbol{x}} = \frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}\left(\boldsymbol{x}(t)\right) \tag{3.35}$$

$$\frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{\Delta t} \approx \boldsymbol{f}(\boldsymbol{x}_k) \tag{3.36}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{f}(\boldsymbol{x}_k)\Delta t \tag{3.37}$$

Figure 3.5 is the block diagram of a INS mechanization in the n-frame with all presented dependencies in this Section. The strapdown computation is a recursive process. The computation of the gravity g, Coriolis effect $(\boldsymbol{\omega}_{ie}^n, \boldsymbol{\omega}_{en}^n)$ and radii of curvature (R_e, R_n) are done in every iteration, based on the previous navigation result.

Note that the height is always negative with the chosen formulation (Equation 3.31), since both, the body and the navigation z-axis, point downwards. This is why the sign in front of the height h in Equations (3.29-3.30) may differ from other common formulations found in literature.

A derivation of the INS equations and detailed discussed implementation methods for SD computation is done by Titterton and Weston (2004).

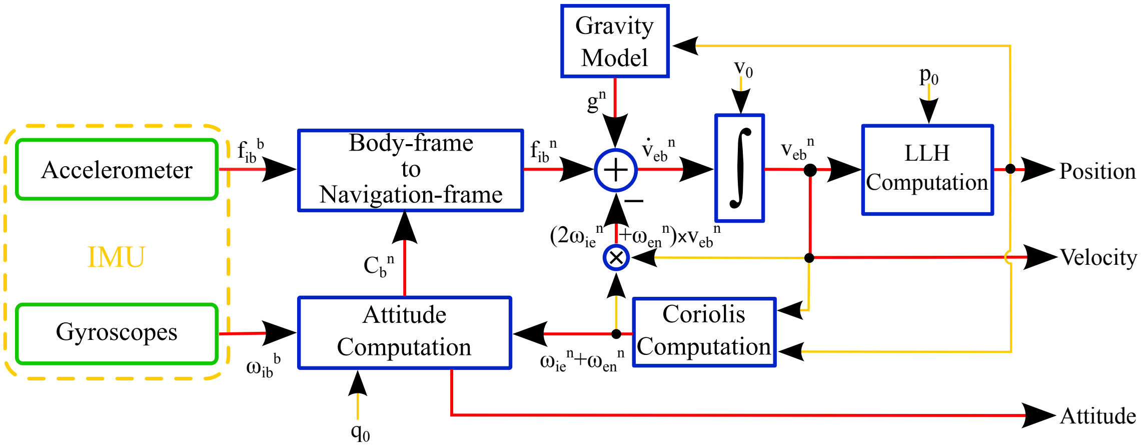

Figure 3.5: Block diagram of an INS mechanization in the navigation frame. The gravity model is described by Equations (3.1-3.4). The b-frame to n-frame transformation of accelerations $\boldsymbol{C}_b^n \boldsymbol{f}_{ib}^b$ is a part of Equation (3.27), where the DCM $\boldsymbol{C}_b^n$ is a result of the attitude computation (Equation 3.14). The terms due to Coriolis effect (Equations 3.24-3.25) must be considered for attitude and velocity calculations. Finally the position in LLH coordinates, given by Equations (3.29-3.31), is the navigation result of the INS mechanization.

3.2.3 RISS Mechanization

The second DR method in this project is a RISS providing 3D position, velocity and attitude. The navigation principle is based on a *rho-theta* technique, where the new position is calculated from the traveled distance (here velocity) and knowledge about the direction of movement. The difference to the conventional SDINS is a reduced set of inertial sensors. With this approach only two accelerometer to calculate the off-plane motion (pitch & roll) and one gyroscope to calculate the heading are needed. The velocity of the vehicle is measured using odometer data, namely wheel-speed sensors. This offers some advantages compared to a full 3D IMU. This method is also chosen to make a proper use of the wheel-speed measurements.

The main advantage is the reduced effect of bias errors in the navigation solution. The accelerometers are used to calculate the off-plane motion without any integration process. A small bias is therefore not as critical as the bias errors of the three gyroscopes in the full IMU approach, involving an integration. The same goes for the calculation of the position by a direct measurement of the velocity, rather than accelerations. Only one integration is needed instead of two, reducing the consequences of accelerometer errors. The calculation of a correct heading is possible by capturing the vertical rotation rate with a gyroscope. The RISS approach is especially suitable for low-cost MEMS, which reduces the costs for inertial sensors. Another advantage is the reduced computational complexity for attitude and velocity calculation (Noureldin, Karamat, and Georgy, 2012).

As a disadvantage, the need of odometry data can be stated. But since the velocity is measured anyway, the sensor data is nowadays accessible through internal CAN-Buses of the car.

A RISS for 2D navigation is presented in Noureldin, Iqbal, and Okou (2008) to cut the costs for land vehicle navigation. The first 3D RISS solution was proposed by Georgy et al. (2010).

In Figure 3.6 the body-frame for the RISS, with all used sensors, is displayed. The y-axis points toward forward direction and the x-axis towards transversal direction. The navigation frame uses the ENU convention.

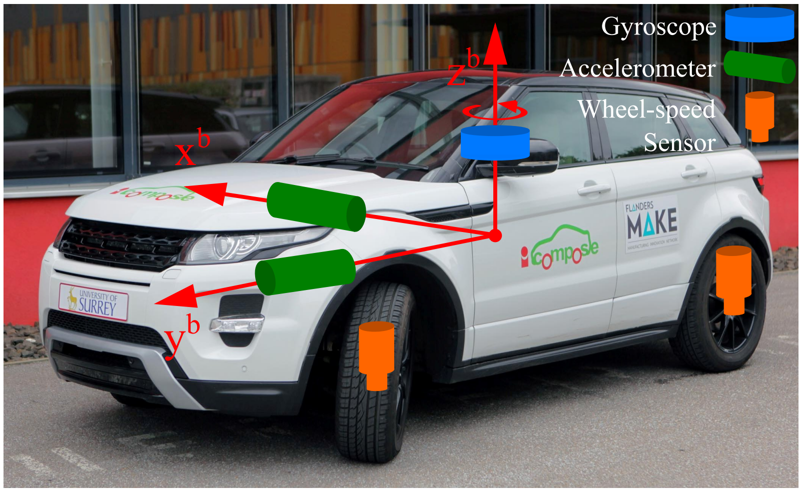

Figure 3.6: Body-frame of the vehicle for RISS

The following RISS equations are given in discrete time formulation. The subscript k indicates the current time epoch, whereas *k-1* is the previous one. For a better readability the superscript notation that indicates the coordinate frame is omitted. Instead the corresponding coordinate axes are denoted with superscripts.

To calculate the tilt with respect to the E-N plane for static conditions using accelerometers, check Equations (3.38 - 3.39). The pitch θ is calculated with the specific force measurement f^y of the y-axis in forward direction. The roll angle ϕ depends on the transversal specific force f^x and the pitch. The gravity g is calculated with the same gravity model as in the SDINS approach.

$$\theta = sin^{-1}\left(\frac{f^y}{g}\right) \tag{3.38}$$

$$\phi = sin^{-1}\left(\frac{-f^x}{g\ cos\theta}\right) \tag{3.39}$$

If the vehicle is moving, the specific force f^y has to be compensated by the vehicle acceleration a_k^{od}, derived from the measured odometer velocity (Equation 3.40). The transversal accelerometer measurement must be compensated for the normal component of acceleration (Equation 3.41), where ω_k^z is the measured rotation rate of the vertical gyroscope.

$$\theta_k = sin^{-1}\left(\frac{f_k^y - a_k^{od}}{g}\right) \tag{3.40}$$

$$\phi_k = -sin^{-1}\left(\frac{f_k^x + v_k^{od}\omega_k^z}{g\ cos\theta_{k-1}}\right) \tag{3.41}$$

The positive yaw angle $\psi = tan^{-1}\left(\frac{U^E}{U^N}\right)$ is defined relative to the north in clockwise direction. Due to earth's rotation rate and the change of orientation of the navigation frame, two additional correction terms have to be considered in Equation (3.42). Similar corrections are done in the SDINS mechanization.

$$\psi_k = tan^{-1}\left(\frac{U^E}{U^N}\right) + (\omega_E sin\varphi_{k-1})\Delta t + \frac{v_{k-1}^E tan\varphi_{k-1}}{R_e + h_{k-1}}\Delta t \tag{3.42}$$

Equations (3.43-3.45) are relations needed to calculate the yaw angle.

$$U_E = sin\psi_{k-1}cos\theta_{k-1}cos\gamma_k^z - (cos\psi_{k-1}cos\phi_{k-1} + sin\psi_{k-1}sin\theta_{k-1}sin\phi_{k-1})sin\gamma_k^z \tag{3.43}$$

$$U_N = cos\psi_{k-1}cos\theta_{k-1}cos\gamma_k^z - (-sin\psi_{k-1}cos\phi_{k-1} + cos\psi_{k-1}sin\theta_{k-1}sin\phi_{k-1})sin\gamma_k^z \tag{3.44}$$

$$\gamma_k^z = \omega_k^z\Delta t \tag{3.45}$$

There is also a simple approximation (Equation 3.46) for the yaw calculation, in case of 2D navigation, neglecting the off-plane motions. It is not used in this project because we consider 3D motion.

$$\psi_k = \psi_{k-1} + (\omega_E sin\varphi_{k-1})\Delta t + \frac{v_{k-1}^E tan\varphi_{k-1}}{R_e + h_{k-1}}\Delta t \tag{3.46}$$

The overall system model with state vector $\boldsymbol{x}_k$ is given in Equation (3.47). The calculation of the LLH position is similar to the SDINS. The ENU velocity components, calculated from the absolute measured vehicle velocity, are derived from Equation (A.14). See Appendix A for more details.

$$\boldsymbol{x}_k = \begin{bmatrix} \varphi_k \\ \lambda_k \\ h_k \\ v_k^E \\ v_k^N \\ v_k^U \\ \theta_k \\ \phi_k \\ \psi_k \end{bmatrix} = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) = \begin{bmatrix} \varphi_{k-1} + \frac{v_{k-1}^N}{R_n + h_{k-1}}\Delta t \\ \lambda_{k-1} + \frac{v_{k-1}^E}{(R_e + h_{k-1})cos\varphi_{k-1}}\Delta t \\ h_{k-1} + v_{k-1}^U \Delta t \\ v_k^{od} sin\psi_{k-1} cos\theta_{k-1} \\ v_k^{od} cos\psi_{k-1} cos\theta_{k-1} \\ v_k^{od} sin\theta_{k-1} \\ sin^{-1}\left(\frac{f_k^y - a_k^{od}}{g}\right) \\ -sin^{-1}\left(\frac{f_k^x + v_k^{od}\omega_k^z}{g\ cos\theta_{k-1}}\right) \\ tan^{-1}\left(\frac{U^E}{U^N}\right) + (\omega_E sin\varphi_{k-1})\Delta t + \frac{v_{k-1}^E tan\varphi_{k-1}}{R_e + h_{k-1}}\Delta t \end{bmatrix} \tag{3.47}$$

The input vector $\boldsymbol{u}_k$ contains all measurement quantities.

$$\boldsymbol{u}_k = \begin{bmatrix} f_k^x & f_k^y & \omega_k^z & v_k^{od} & a_k^{od} \end{bmatrix}^T \tag{3.48}$$

The block diagram of the RISS mechanization is shown in Figure 3.7. With the chosen b-frame and ENU convention, it is to mention, that the height h, in the RISS mechanization, is always positive. Fore a more detailed look at RISS see Noureldin, Karamat, and Georgy (2012).

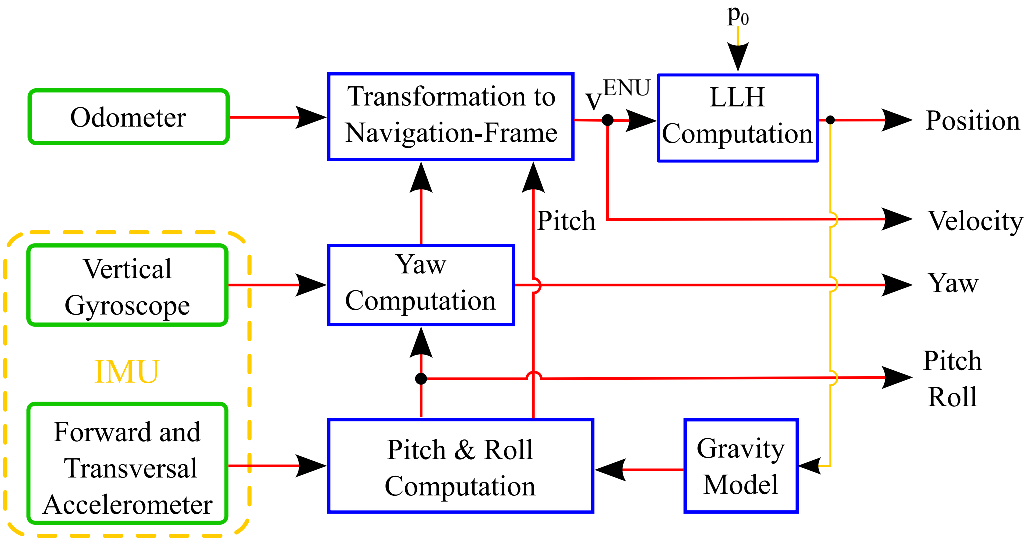

Figure 3.7: Block diagram of an RISS mechanization in the navigation frame. The attitude computation is given by Equation (3.42) for the yaw angle and Equations (3.40-3.41) respectively for pitch and roll angle. The gravity model (Equations 3.1-3.4) is the same as for the strapdown algorithm. The transformation of the absolute odometer velocity into the ENU frame is done according to Equation (A.14). The LLH position is calculated similar to Equations (3.29-3.31), considering the ENU coordinate convention. The overall system model for the RISS mechanization process is given by Equation (3.47).

The presented system models in this Chapter for SDINS and RISS mechanization are used in Chapter 5 for GPS/INS integration. A system error model, necessary for the the indirect integration method, is derived based on these models. To perform a correct system error modeling, the inertial sensor errors are analyzed in the following Chapter 4.

Chapter 4

MEMS Sensor Error Analysis

This Chapter presents several of the error sources of inertial sensors, how to identify them and sets up a sensor error model that is used for GPS/INS integration.

The analysis of inertial sensors is important, especially when low-cost sensors (MEMS grade) are used. With an appropriate sensor error model the navigation error of an INS can be reduced. In an GPS aided INS approach, this is important during GPS outages.

The measurements of the IMU are affected by different error sources (ISO/IEC, 2008). They can be classified as deterministic and stochastic errors. Deterministic errors are for example bias, scale factor, dead zone and misalignment. Stochastic errors are random errors that occur randomly, for example in bias or scale factor errors over time. The bias-drift or the non-linearity of the scale factor can be described as stochastic processes. The deterministic errors are due to manufacturing or mounting defects and can be corrected with an calibration of the sensors (Quinchia et al., 2013).

Since the measurements were performed by partner companies and the IMU was not available to do test measurements by our own, deterministic errors are not further discussed. This Chapter focuses on the modeling of the stochastic bias-drift error. Widely used techniques to analyze stochastic errors and to estimate stochastic error model parameters are variance and the autocorrelation.

In Section 4.1 the theory behind sensor error models is described while in Section 4.2 and 4.3 methods to identify the model parameters are discussed.

4.1 Sensor Error Modeling

A typical bias-drift of inertial sensors can be modeled as a combination of different stochastic processes like White Noise (WN), Random Walk (RW) and first order Gauss Markov (GM) processes (Quinchia et al., 2013). In Section 4.2 it will be shown that this statement is valid for the used IMU in this project too.

A first order GM process is used to describe a large number of physical random process. A simple mathematical expression is another benefit with this kind of modeling, shown by Equation (4.1) in continuous form.

$$\dot{x} = -\frac{1}{T_c} x + w_{GM} \tag{4.1}$$

$$\sigma^2_{w,GM} = 2\beta\sigma^2_{GM} \tag{4.2}$$

The driven noise variance $\sigma^2_{w,GM}$ (Equation 4.2) depends on the inverse correlation time $\beta = \frac{1}{T_c}$ and the covariance of the process σ^2_{GM}. The zero mean random process x, with correlation time T_c and driven noise w_{GM} can also be expressed in time discrete from (Equations 4.3-4.4). Since the model is used in an discrete KF algorithm, a discrete representation of all processes is necessary.

$$x_k = (1 - \beta\Delta t)\, x_{k-1} + w_{k,GM} \tag{4.3}$$

$$\sigma^2_{w_k,GM} = \sigma^2_{GM}\left(1 - e^{-2\beta\Delta t}\right) \tag{4.4}$$

A continuous RW process (Equation 4.5) describes the results of uncorrelated noise during integration in the mechanization process. The discrete form is shown by Equation (4.6), where w_{RW} is uncorrelated white noise with variance $\sigma^2_{w_k,RW}$.

$$\dot{x} = w_{RW} \tag{4.5}$$

$$x_k = x_{k-1} + w_{k,RW} \tag{4.6}$$

$$\sigma^2_{w_k,RW} = \sigma^2_{RW}\Delta t \tag{4.7}$$

The combination of GM, RW and WN is done with a discrete time invariant state space model, where the state vector is $\boldsymbol{x}_k = [x_{k,GM}, x_{k,RW}]^T$ and σ_{WN} is the standard deviation of the WN process.

$$\boldsymbol{x}_k = \begin{bmatrix} (1-\beta\Delta t) & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_{k-1} + \begin{bmatrix} \sigma_{GM}\sqrt{(1-e^{-2\beta\Delta t})} \\ \sigma_{RW}\sqrt{\Delta t} \end{bmatrix} w_k \tag{4.8}$$

$$y_k = \begin{bmatrix} 1 & 1 \end{bmatrix} \boldsymbol{x}_k + \left(\frac{\sigma_{WN}}{\Delta t}\right) v_k \tag{4.9}$$

The output y_k represents the modeled bias-drift, where w_k and v_k are uncorrelated white noise terms with unit variance. Note that the state-space model from Equation (4.8) is of second order when all three processes are involved. For a 6-DOF IMU, a 12th order state-space model is needed, only to model the sensor errors. The proposed conventional sensor error approach is chosen because it can be comfortably integrated into discrete Kalman filtering and computational complexity is kept small, compared to autoregressive processes for example.

In Section 4.2 it will be shown that a bias-drift does not necessarily consist of a combination of all three processes. The following two Sections are presenting methods to identify the underlying random processes of a inertial sensor and how to find the parameters $\beta, \sigma_{GM}, \sigma_{RW}$ and σ_{WN}.

4.2 Allan Variance

The Allan Variance (AV) is a method to provide information on the types and magnitude of error terms of an inertial sensor. It is a technique operating in the time domain and is basically representing the root mean square random-drift errors as a function of averaging times (El-Sheimy, Hou, and Niu, 2008).

The AV was originally developed to observe the frequency stability of oscillators. Today it is successfully used to analyze random inertial sensor errors. To perform an AV analysis inertial measurement data under static conditions is needed. The method is popular because it is simple to compute and easy to understand since it is a graphical analysis. A drawback is that the accuracy of the estimation depends on the recorded data length. Several hours of measurement data are suggested to get a reliable result. Since the AV is working with the entire data length, the computation can take some time (Quinchia et al., 2013).

There is also a method operating in the frequency domain by using the power spectral density (Slavov and Petkov, 2010; Quinchia et al., 2013), but it is more difficult to understand and not that intuitive.

4.2.1 Theory

With AV analysis it is possible to identify five basic noise terms for inertial sensors (IEEE, 1997). The terminology for two noise terms differs for gyroscopes and accelerometers due to different measured quantities. Both sensor types have quantization noise, bias instability and rate ramp in common. In case of gyroscopes there are angle random walk and rate random walk, whereas for accelerometers they are called velocity random walk and acceleration random walk.

There is a unique relation between the power spectral density and AV. In El-Sheimy, Hou, and Niu (2008) a derivation of the AV from the power spectral density is done. A detailed explanation of the AV analysis can also be found there and the basis for the performed sensor error analysis is explained.

The different error sources appear with different slopes in the AV plot. It is normally plotted as the square root of the AV versus averaging time τ with a log-log scale. Quantization noise has a slope of -1 and rate ramp for example a slope of +1. These two error sources are not further discussed because they do not appear in the sensor data. An example for an AV plot is shown in Figure 4.1.

Three different error sources, resulting in three different slopes, can be detected for the z-axis accelerometer. Velocity random walk results in a $-\frac{1}{2}$ slope and is caused basically by WN. Bias instability appears as slope 0 and can be modeled with a GM process. Acceleration random walk has a slope of $+\frac{1}{2}$ and refers to a RW process. The same is valid for gyroscope angle random walk (slope $-\frac{1}{2}$) and rate random walk (slope $\frac{1}{2}$).

The missing parameters σ_{GM}, σ_{RW} and σ_{WN} are calculated from identified error coefficients through AV analysis. Velocity-/Angle random walk coefficients are labeled as N_a and N_g. Bias instability coefficients are B_a for accelerometers and B_g for gyroscopes. The notation for acceleration and rate random walk is K_a and K_g, respectively.

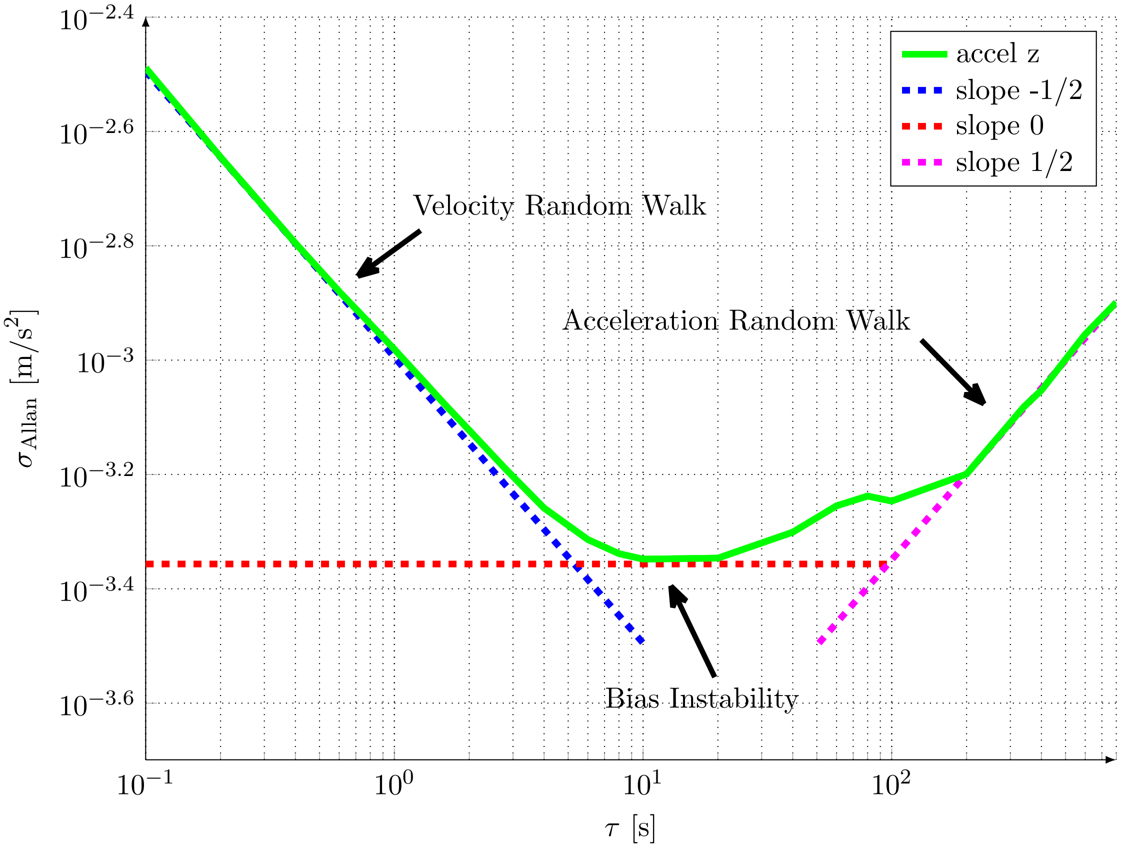

Figure 4.1: Example of a AV plot for z-accelerometer with different error sources

The parameters N_a and N_g are found by reading the intersection point of slope $-\frac{1}{2}$ with $\tau = 1s$ on the y-axis, representing the Allan standard deviation. The bias instability coefficients B_a and B_g are found at the minimum of the AV plot, where the slope is 0. The numerical value for coefficients K_a and K_g are found at the intersection point of slope $\frac{1}{2}$ with $\tau = 3s$ at the y-axis.

Note that the units in Figure 4.1 are in SI units s and m/s^2. Often other units, like $m/s/h$ or deg/h are used in the AV plot. To find the correct intersection point, it is important that both axis share the same unit in time. For the example of $m/s/h$, the intersection with the x-axis is at $\tau = 1h$ (for N_a, N_g) or $\tau = 3h$ (for K_a, K_g).

If other units than SI units are used in the AV plot, the error coefficients have to be transformed afterwards to get the corresponding noise variances. This is presented in the Subsection 4.2.3, regarding AV analysis.

4.2.2 Computation

One way to compute the AV is in terms of output angle or velocity , since the AV analysis can be used for gyroscopes as well as for accelerometers (El-Sheimy, Hou, and Niu, 2008).

In Equation (4.10) the term $\Omega(t')$ refers to the gyroscope or acceleration measurement. No lower integration limit is defined because only angle or velocity differences are used in the definition. The integral quantity $\Theta(t)$ represents the output angle or velocity. The measurements are made at discrete times $t = k\Delta t, k = 1, 2, 3..., N$, where N is the number of measured data points. For a discrete set of samples Equation (4.10) is a cumulative sum with simplified notation $\Theta_k = \Theta(t = k\Delta t)$ (IEEE, 1997).

$$\Theta(t) = \int_0^t \Omega(t')dt' \tag{4.10}$$

The cluster averages $\overline{\Theta}_k(\tau)$ are defined in Equation (4.11) and the index $m = \frac{\tau}{\tau_0}$ depends on the desired averaging time τ, where $\tau_0 = \Delta t$. The definition of the AV is given by Equation (4.12) where $\langle\rangle$ denotes the ensemble average.

$$\overline{\Theta}_k(\tau) = \frac{\Theta_{k+m} - \Theta_k}{\tau} \qquad \overline{\Theta}_{k+m}(\tau) = \frac{\Theta_{k+2m} - \Theta_{k+m}}{\tau} \tag{4.11}$$

$$\sigma^2_{Allan}(\tau) = \frac{1}{2}\langle\left(\overline{\Theta}_{k+m}(\tau) - \overline{\Theta}_k(\tau)\right)^2\rangle \tag{4.12}$$

To get an AV estimator, Equation (4.11) is substituted into Equation (4.12), which leads to following Equation,

$$\sigma^2_{Allan}(\tau) = \frac{1}{2\tau^2}\langle(\Theta_{k+2m} - 2\Theta_{k+m} + \Theta_k)^2\rangle \quad . \tag{4.13}$$

According to the AV definition, an AV estimator, by overlapping method (Freescale, 2015), is given by Equation (4.14).

$$\sigma^2_{Allan}(\tau) = \frac{1}{2\tau^2}\frac{1}{N-2m}\sum_{k=1}^{N-2m}(\Theta_{k+2m} - 2\Theta_{k+m} + \Theta_k)^2 \tag{4.14}$$

To get the Allan standard deviation $\sigma_{Allan}(\tau)$, used in the log-log plot for the analysis, the square root is applied to the Allan Variance.

$$\sigma_{Allan}(\tau) = \sqrt{\sigma^2_{Allan}(\tau)} \tag{4.15}$$

Note that τ_0 is the sampling time Δt of the measurement data and therefore the smallest possible averaging time, resulting in an averaging factor $m = 1$. The averaging factor is restricted by $m < \frac{N}{2}$ (see Equation 4.14), which implies that the biggest possible averaging time τ is half of the measurement time span. The desired τ value for AV calculation must be an integer multiple of τ_0.

4.2.3 Allan Variance Analysis

The AV analysis is performed with a 30 minute static IMU measurement at a sampling rate of 50Hz.

Figure 4.2 shows the results for the triad of accelerometers. All accelerometers have a similar behavior for short and long cluster times. Only the x-axis accelerometer shows a different characteristic for long cluster times. The reason could be that for long cluster times, less measurements data points are available for AV calculation and the estimate becomes not reliable any more. The velocity random walk is almost identical for the three accelerometers. The smallest bias instability belongs to the y-axis accelerometer. The AV analysis for accelerometers shows that the proposed sensor error model from Section 4.1, consisting of a combination of WN, first order GM and RW processes can be applied. The x-axis random error for long averaging times is modeled with a RW process too, assuming the same behavior as the other accelerometers.

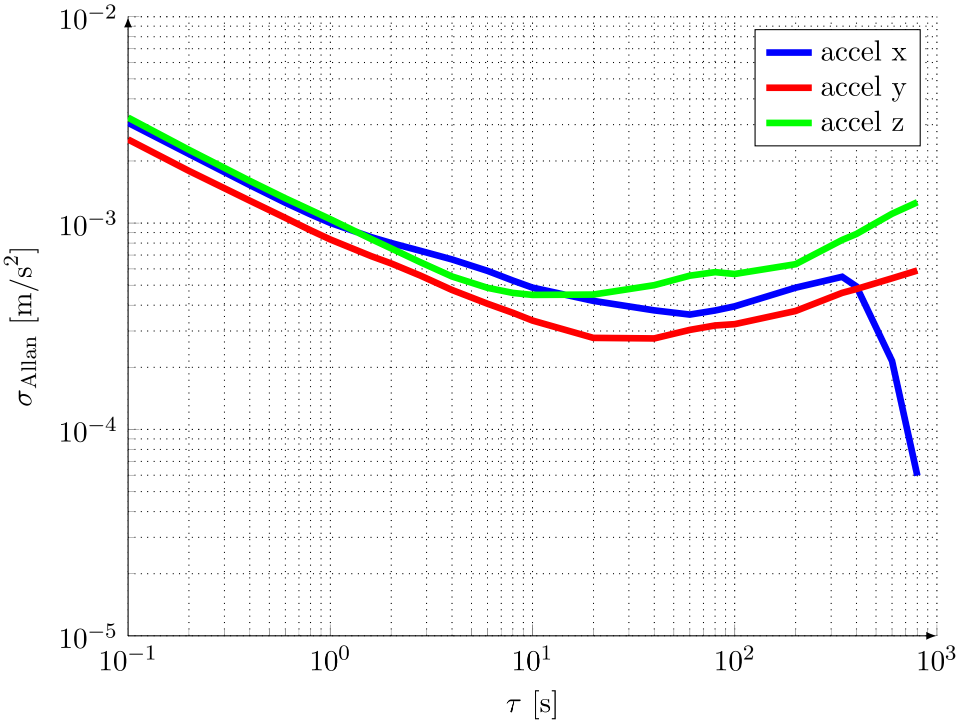

Figure 4.2: Allan standard deviation for accelerometers

The Allan standard deviation plot in Figure 4.3 displays the result for gyroscopes. Only one error source is dominant for all cluster times. An angle random walk with slope $\frac{1}{2}$ is the main random error caused by WN. Therefore only values for N_g, but no values for B_g and K_g are found.

The identified error coefficients are listed in Table 4.1. The AV plots are created with a self implemented function for AV calculation, but the results are also validated with an independent MATLAB function (Czerwinski, 2010), which provides an overlapping AV

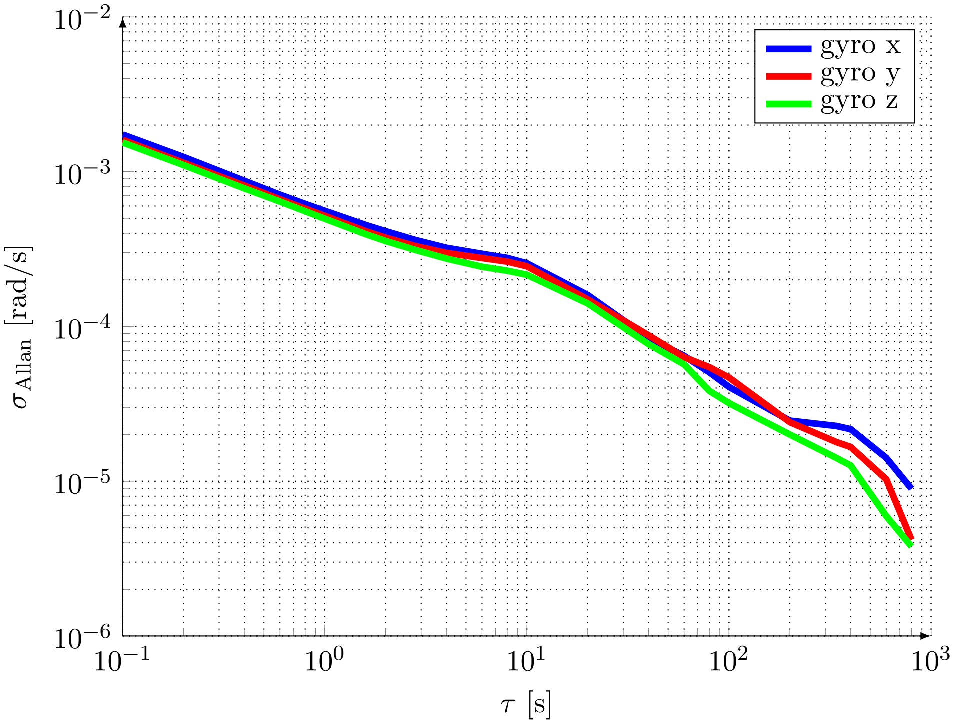

Figure 4.3: Allan standard deviation for gyroscopes

estimate. The results are almost identical, up to a residual of 10^{-7}. Quinchia et al. (2013) states that the parameters from AV estimation have to be considered as initial approximation and manually tuned in the KF to obtain the desired performance. This is also done in this thesis. The error coefficients are assumed larger as identified, because under realistic dynamic road measurements, the errors can be assumed larger due to vibrations caused by the engine, road condition and other influences affecting the measurements. The parameters used for simulation, are listed at Table C.1 in the Appendix.

	velocity random walk N_a $[m/s/\sqrt{s}]$	bias instability B_a $[m/s^2]$	acceleration random walk K_a $[m/s/s^{3/2}]$
accel x	1.05×10^{-3}	4×10^{-4}	6.2×10^{-5}
accel y	9.8×10^{-4}	2.76×10^{-4}	4.2×10^{-5}
accel z	1.05×10^{-3}	4.2×10^{-4}	8.4×10^{-5}
	angle random walk N_g $[rad/\sqrt{s}]$	bias instability B_g $[rad/s]$	rate random walk K_g $[rad/s^{3/2}]$
gyro x	5.7×10^{-4}	-	-
gyro y	5.1×10^{-4}	-	-
gyro z	4.97×10^{-4}	-	-

Table 4.1: Identified error coefficients from AV analysis

The error coefficients are used to calculate the system noise covariance matrix for the

ESKF in the GPS/INS integration. In this approach the random sensor errors are treated as system noise, like it is suggested in Wendel (2011).

The system noise covariance matrix $\mathbf{Q}_k$ is explained in more detail in Chapter 5. This Chapter only shows how to calculate the individual entries. In Equation (4.16) the diagonal structure of the matrix is defined.

$$\boldsymbol{Q}_k = \begin{bmatrix} diag(\boldsymbol{q}_{N_a}) & 0_{(3x3)} & 0_{(3x3)} & 0_{(3x3)} \\ 0_{(3x3)} & diag(\boldsymbol{q}_{N_g}) & 0_{(3x3)} & 0_{(3x3)} \\ 0_{(3x3)} & 0_{(3x3)} & diag(\boldsymbol{q}_{B_a}) & 0_{(3x3)} \\ 0_{(3x3)} & 0_{(3x3)} & 0_{(3x3)} & diag(\boldsymbol{q}_{K_a}) \end{bmatrix} \tag{4.16}$$

The *diag()* operator creates a diagonal matrix from the vectors $\boldsymbol{q}_{N_a} = [q_{N_{ax}}, q_{N_{ay}}, q_{N_{az}}]^T$, $\boldsymbol{q}_{N_g} = [q_{N_{gx}}, q_{N_{gy}}, q_{N_{gz}}]^T$, $\boldsymbol{q}_{B_a} = [q_{B_{ax}}, q_{B_{ay}}, q_{B_{az}}]^T$ and $\boldsymbol{q}_{K_a} = [q_{K_{ax}}, q_{K_{ay}}, q_{K_{az}}]^T$. All those quantities are obtained through the AV analysis and parameters from Table 4.1. To describe the relation between identified parameters and calculated covariances, the x-axis accelerometer parameters are used as an example.

The spectral density in discrete time of the process driving noise of a WN process is expressed in Equation 4.17, where $\sigma^2_{WN_{ax}}$ is the variance of the WN process. Since the identified parameters N_a from the AV analysis are already in international system units, no further conversion of units has to be done.

$$q_{N_{ax}} = \frac{\sigma^2_{WN_{ax}}}{\Delta t} = \frac{N^2_{ax}}{\Delta t} \tag{4.17}$$

The spectral density in discrete time for the first order GM process is calculated from the bias instability parameter B_{ax} and corresponding correlation time $T_{c,ax}$ from Table 4.2. In IEEE (1997) it is suggested to multiply B_{ax} with factor 0.664 to get the same variance $\sigma^2_{GM_{ax}}$ as from the power spectral density approach.

$$q_{B_{ax}} = \sigma^2_{GM_{ax}} \left(1 - e^{-2\Delta t/T_{c,ax}}\right) \tag{4.18}$$

$$\sigma_{GM_{ax}} = 0.664 B_{ax} \tag{4.19}$$

The discrete spectral density for the acceleration random walk is expressed in Equation (4.20). No further unit conversion has to be done, due to the usage of international system units.

$$q_{K_{ax}} = \sigma^2_{RW_{ax}} \Delta t = K^2_{ax} \Delta t \tag{4.20}$$

4.3 Autocorrelation

The autocorrelation analysis is used to identify the correlation time parameter T_c in the GM process.

The first order GM process (Equation 4.1) has an exponentially decaying autocorrelation function (Noureldin, Karamat, and Georgy, 2012). The autocorrelation is described by Equation (4.21) and shown in Figure 4.4.

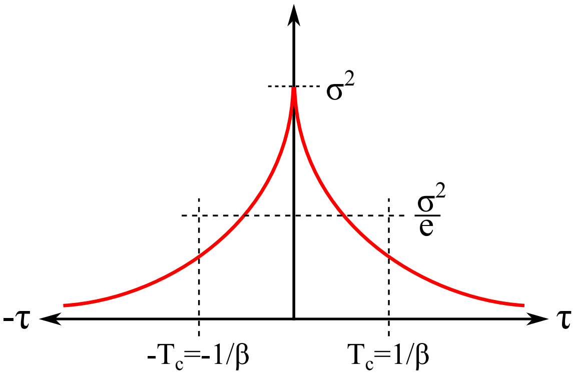

Figure 4.4: Autocorrelation of a first order GM process

$$R_{xx}(\tau) = \sigma^2 e^{-\beta|\tau|} \tag{4.21}$$

Before the autocorrelation is calculated, the mean of the measurement data is removed. To calculate a one dimensional autocorrelation function, jointly stationary random processes are assumed. To identify the correlation time, it is suggested in Quinchia et al. (2013) to perform wavelet de-noising, because otherwise the time correlation property of the signal can not be seen. Without the de-noising the uncorrelated noise terms are dominant, resulting in a peak, basically at time lag zero. A significant property of an uncorrelated white noise process. The performed de-noising is done with a non-causal moving average filter. The results are compared with the suggested level six wavelet de-noising using the integrated MATLAB function *wden* in exemplary configuration. Both methods provide a similar solution.

The autocorrelation is normalized, where the correlation time is reached at the $\frac{1}{e}$ intersection line. In Figure 4.5 the autocorrelation for accelerometers is shown before de-noising. After the de-noising (Figure 4.6) a correlation in the signals is visible. According to the trend of the signals it must be presumed that more complex error processes, than a first order GM process, are present. As a rough approximation, a first order GM process is still chosen.

The autocorrelation analysis for gyroscopes is also performed. The comparison between Figure 4.7 and Figure 4.8 indicates that even after de-noising, the remaining signal still has a representative white noise component. This corresponds with the results from AV analysis, where only an angle random walk is identified, but no bias instability or rate random walk.

The identified correlation times are listed in Table 4.2.

	accel x	accel y	accel z	gyro x	gyro y	gyro z
correlation time $T_c[s]$	2.6	42	66	-	-	-

Table 4.2: Identified correlation times T_c for GM process

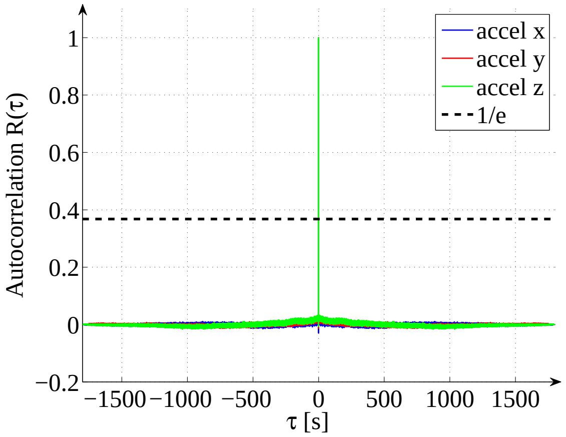

Figure 4.5: Autocorrelation for accelerometers before de-noising

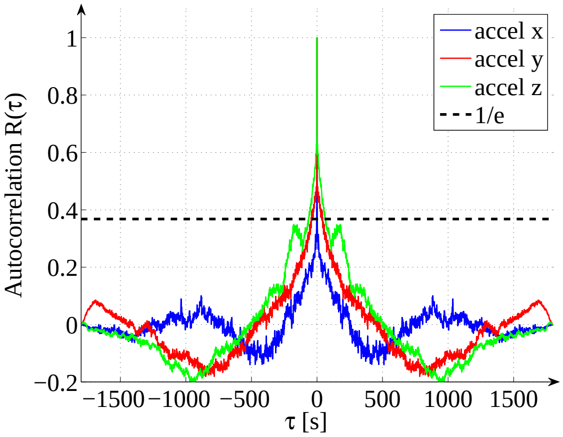

Figure 4.6: Autocorrelation for accelerometers after de-noising

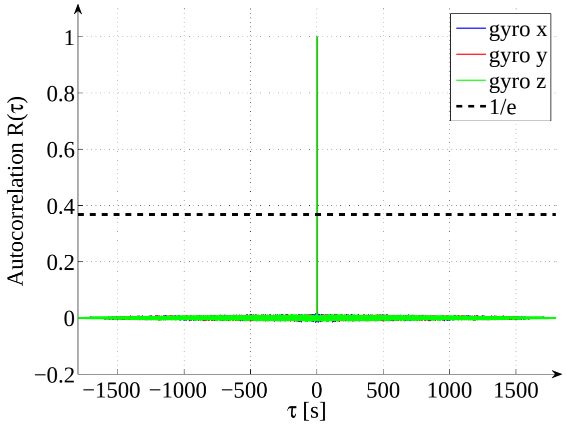

Figure 4.7: Autocorrelation for gyroscopes before de-noising

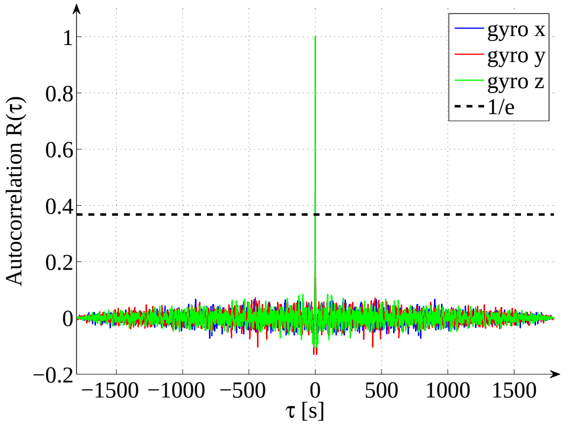

Figure 4.8: Autocorrelation for gyroscopes after de-noising

The autocorrelation analysis has its limitations. The calculated autocorrelation is an estimation, because only a finite number of data points is available. The accuracy depends therefore on the recorded data length. Furthermore due to the graphical fitting of an exponential decaying function with a signal course of different behavior, there is room for parameter variations. Like in the case of error coefficients, the correlation times can be manually tuned inside the KF. This is not done for correlation times in this thesis, because a variation (by factor 1/2 and 2) shows no effect on the simulation result.

The results from the AV and autocorrelation analysis presented in this Chapter are used in the error model approach for GPS/INS integration. The system noise covariance matrix $\boldsymbol{Q}_k$, for the ESKF in Subsection 5.2.2, is calculated by the identified error coefficients. The bias-drift is estimated within the KF and used for online-calibration of the IMU.

Chapter 5

GPS/INS Integration

This Chapter presents how the GPS/INS integration is performed in detail. The chosen sensor fusion method is Kalman filtering. In Section 5.1 the basic principles and relevant equations of different kinds of KF are discussed. In Section 5.2 and 5.3 the equations for the indirect approach are declared. The implemented fusion algorithm and other peculiarities regarding the measurement data are presented in Section 5.4.

5.1 Kalman Filtering

Kalman filtering is a well known state estimation technique, not only for GPS/INS integration. It is used to estimate the states and the uncertainty of a system, that is corrupted by noise terms. If the system is linear and the noise is unbiased, uncorrelated Gaussian distributed, the KF is an optimal estimator, in a sense of minimal variance. There exist some other variations of the KF, like the EKF, the unscented KF for non-linear systems, or many others.

The Kalman filter is especially attractive for digital computation because the algorithm works in a recursive scheme and is suitable for real time applications. The algorithm is basically divided into two steps: the prediction step and the update or measurement step. A notation is chosen, where $\widehat{\boldsymbol{x}}$ denotes an estimated state. The (-) superscript symbolizes a predicted or so called *a priori* quantity, whereas (+) labels a updated or *a posteriori* value. The predicted state is based on the previous estimated state. With a new available measurement, the predicted state is corrected with a weighted update to get the new estimated state.

Kalman filtering is known for many years now and is successfully applied in numerous technical domains. The theory behind Kalman filtering is complex and different mathematical disciplines are forming the foundation of the derivation. Only the relevant relations and main equations which are describing the KF algorithm are discussed in this work. There exists an extensive amount of literature related to Kalman filtering. For general information on Kalman filtering the books by Grewal and Andrews (2008) and Kim (2011) are recommended. The work of Wendel (2011) and Noureldin, Karamat, and Georgy (2012) influenced the approach in this thesis to a large extend. Many aspects that can not be covered here and a more detailed explanation of the equations and models can be found there.

5.1.1 Kalman Filter

The application of Kalman filtering requires a linear system (Equation 5.1) and measurement model (Equation 5.2) in discrete time. If the system is in continuous time, the conversion is shown in Appendix B.

$$\boldsymbol{x}_k = \boldsymbol{\Phi}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{B}_{k-1}\boldsymbol{u}_{k-1} + \boldsymbol{G}_{k-1}\boldsymbol{w}_{k-1} \tag{5.1}$$

$$\boldsymbol{z}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k \tag{5.2}$$

It is assumed that the noise terms $\boldsymbol{w}_{k-1}$ and $\boldsymbol{v}_k$ are unbiased, uncorrelated and Gaussian distributed. The assumptions regarding system and measurement noise are described by Equations (5.3-5.6). $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are the system and measurement covariance matrices. Because a noise source can affect more than one system state, the noise distribution matrix $\boldsymbol{G}_{k-1}$ takes into account the coupling of noise disturbances.

$$E[\boldsymbol{w}_k] = 0 \qquad E[\boldsymbol{v}_k] = 0 \qquad \forall k \tag{5.3}$$

$$E[\boldsymbol{w}_k\boldsymbol{v}_j^T] = 0 \qquad \forall k, j \tag{5.4}$$

$$E[\boldsymbol{w}_k\boldsymbol{w}_j^T] = \begin{cases} \boldsymbol{Q}_k & k = j \\ 0 & k \neq j \end{cases} \tag{5.5}$$

$$E[\boldsymbol{v}_k\boldsymbol{v}_j^T] = \begin{cases} \boldsymbol{R}_k & k = j \\ 0 & k \neq j \end{cases} \tag{5.6}$$

The five Equations (5.7-5.11) are defining the KF algorithm. The first step is the prediction of the state $\widehat{\boldsymbol{x}}_k^-$ and the error covariance matrix $\boldsymbol{P}_k^-$.

$$\widehat{\boldsymbol{x}}_k^- = \boldsymbol{\Phi}_{k-1}\widehat{\boldsymbol{x}}_{k-1}^+ + \boldsymbol{B}_{k-1}\boldsymbol{u}_{k-1} \tag{5.7}$$

$$\boldsymbol{P}_k^- = \boldsymbol{\Phi}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{\Phi}_{k-1}^T + \boldsymbol{G}_{k-1}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^T \tag{5.8}$$

If a new measurement $\boldsymbol{z}_k$ is available, the Kalman gain matrix $\boldsymbol{K}_k$ can be computed. It serves as a kind of weighting matrix in Equation (5.10) for the new state estimate $\widehat{\boldsymbol{x}}_k^+$. With $\boldsymbol{K}_k$ the algorithm determines how much the new measurement can be “trusted“. The error covariance matrix is also updated (Equation 5.11).

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \tag{5.9}$$

$$\widehat{\boldsymbol{x}}_k^+ = \widehat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k\widehat{\boldsymbol{x}}_k^-) \tag{5.10}$$

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^- \tag{5.11}$$

After the calculation of all five Equations, $\widehat{\boldsymbol{x}}_k^+$ and $\boldsymbol{P}_k^+$ are used in the next iteration. The calculation of the prediction step is independent of new measurement data. This is used in Section 5.4, where the prediction step is calculated with a fixed step size, whereas the estimation step is only performed when a new measurement is available. This circumstance allows the realization of a multi-rate system, since GPS measurements have a different sampling rate compared to inertial sensors.

The iterative algorithm needs initial conditions to start the calculation. The initial state $\widehat{\boldsymbol{x}}_0$ has a known mean $\widehat{\boldsymbol{x}}_0^*$ and a covariance $\boldsymbol{P}_0$.

$$\widehat{\boldsymbol{x}}_0^* = E[\widehat{\boldsymbol{x}}_0] \tag{5.12}$$

$$\boldsymbol{P}_0 = E\left[(\widehat{\boldsymbol{x}}_0 - \widehat{\boldsymbol{x}}_0^*)(\widehat{\boldsymbol{x}}_0 - \widehat{\boldsymbol{x}}_0^*)^T\right] \tag{5.13}$$

5.1.2 Extended Kalman Filter

If the system model (Equation 5.14) or the measurement model (Equation 5.15) is non-linear, a so called EKF is used for Kalman filtering. The assumptions for the noise terms are the same as for linear Kalman filtering.

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{G}_{k-1}\boldsymbol{w}_{k-1} \tag{5.14}$$

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{5.15}$$

The Kalman filtering principle stays the same in the extended version. The five filter Equations (5.16-5.20) nearly are the same. The difference is that for the calculation of the predicted and estimated state, the non-linear models are used. The calculation of the Kalman gain matrix $\boldsymbol{K}_k$ and error covariance matrices are unchanged. The non-linear behavior is therefore approximated with the Jacobian matrices from Equation (5.21) and (5.22).

$$\widehat{\boldsymbol{x}}_k^- = \boldsymbol{f}(\widehat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}_{k-1}) \tag{5.16}$$

$$\boldsymbol{P}_k^- = \boldsymbol{\Phi}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{\Phi}_{k-1}^T + \boldsymbol{G}_{k-1}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^T \tag{5.17}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \tag{5.18}$$

$$\widehat{\boldsymbol{x}}_k^+ = \widehat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\left(\boldsymbol{z}_k - \boldsymbol{h}(\widehat{\boldsymbol{x}}_k^-)\right) \tag{5.19}$$

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^- \tag{5.20}$$

$$\boldsymbol{\Phi}_k = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\widehat{\boldsymbol{x}}_k} \tag{5.21}$$

$$\boldsymbol{H}_k = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\widehat{\boldsymbol{x}}_k} \tag{5.22}$$

Because the computation of the matrices is based on an approximation, the EKF can only be reasonably used if the non-linearities are not dominating and the noise parameters are chosen carefully (Boutayeb, Rafaralahy, and Darouach, 1997).

5.1.3 Error State Formulation

The indirect GPS/INS integration approach is using an error state formulation. The idea behind this approach is to estimate the errors of the INS and subsequently correcting them. This methodology has some benefits compared to the direct approach (see Subsection 2.2.4). Because the filter is estimating error states, rather than full states, the filter is called ESKF (Wendel, 2011).

Assume a model like in Equation (5.23).

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{G}\boldsymbol{w} \qquad \boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v} \tag{5.23}$$

An approximation for the system model by Taylor series expansion about some linearization point $\overline{\boldsymbol{x}}$ looks like:

$$\dot{\boldsymbol{x}} \approx \boldsymbol{f}(\overline{\boldsymbol{x}}) + \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\overline{\boldsymbol{x}}} \cdot (\boldsymbol{x} - \overline{\boldsymbol{x}}) + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{G}\boldsymbol{w} \tag{5.24}$$

The same principle can be applied to the estimated state vector.

$$\dot{\widehat{\boldsymbol{x}}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}) + \boldsymbol{B}\boldsymbol{u} \approx \boldsymbol{f}(\overline{\boldsymbol{x}}) + \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\overline{\boldsymbol{x}}} \cdot (\widehat{\boldsymbol{x}} - \overline{\boldsymbol{x}}) + \boldsymbol{B}\boldsymbol{u} \tag{5.25}$$

Subtracting Equation (5.24) from Equation (5.25) yields to

$$\dot{\widehat{\boldsymbol{x}}} - \dot{\boldsymbol{x}} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\overline{\boldsymbol{x}}} \cdot (\widehat{\boldsymbol{x}} - \boldsymbol{x}) - \boldsymbol{G}\boldsymbol{w} \tag{5.26}$$

With the definition of the error state vector $\Delta\boldsymbol{x} = \widehat{\boldsymbol{x}} - \boldsymbol{x}$, rewriting Equation (5.26) leads to

$$\Delta\dot{\boldsymbol{x}} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\overline{\boldsymbol{x}}} \cdot \Delta\boldsymbol{x} - \boldsymbol{G}\boldsymbol{w} \tag{5.27}$$

Analogous the measurement model is derived the same way. Now a new linear system in terms of error states is defined. The system and measurement matrices are Jacobian matrices of the non-linear functions.

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{F}\Delta\boldsymbol{x} - \boldsymbol{G}\boldsymbol{w} \tag{5.28}$$

$$\Delta\boldsymbol{z} = \boldsymbol{H}\Delta\boldsymbol{x} - \boldsymbol{v} \tag{5.29}$$

$$\boldsymbol{F} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\widehat{\boldsymbol{x}}} \tag{5.30}$$

$$\boldsymbol{H} = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\widehat{\boldsymbol{x}}} \tag{5.31}$$

After discretization of Equations (5.28-5.52), the KF algorithm from Subsection 5.1.1 can be applied to estimate the error states. If these estimated errors are used to correct the states of the INS, then the corrected states become basically estimated full states. These states are fed back to the INS for the next iteration. The state vector of the ESKF must be set to zero after every correction. Therefore a state prediction step for error states can be omitted, since the last estimated error state is zero. Only the error covariance matrix has to be calculated in every iteration. Wendel (2011) is calling this a closed-loop ESKF. It is stated that an ESKF in feedback configuration, equals an EKF (Gustafsson, 2012).

5.2 Loosely-Coupled Strapdown Integration

This Section explains in detail the loosely-coupled approach for the full IMU based strapdown algorithm of Subsection 3.2.2. An indirect design in error state space formulation is chosen (see Section 2.2). With this design the measurements of the IMU are processed in the prediction step of the filter, not in the measurement step, like it could be assumed. The identified noise parameters of Section 4.2.3 are therefore used to describe the system noise of the error state space model. The GPS measurements are aiding the INS navigation solution and are treated as real measurements, used in the update step within the KF. In an ESKF, the residuals between INS and GPS navigation solution are processed, rather than absolute position data. With this design it is possible to consider the multirate behavior of the system, since the INS solution is computed with 50Hz, whereas a GPS measurement is only available every second. Furthermore, the bias-drift is estimated within the KF, based on the random sensor error processes from Section 4.1, to enable an online sensor calibration.

To establish the error model for the ESKF, it is necessary to examine how errors propagate within the INS mechanization. This is very complex for velocity and attitude errors because the equations are non-linear differential equations, containing transformation matrices, direction cosines and cross products. To form a linear error model, they have to be linearized (Wendel, 2011). The derivation is only shown for the position error. Just the final velocity and attitude error equations are presented, which are used in the state space formulation. For the exact derivation see Wendel (2011) or Titterton and Weston (2004).

5.2.1 INS Error Equations

The error state vector is defined by (5.32). The position error $\Delta\boldsymbol{p}$ is in units of meter. The attitude error $\Delta\boldsymbol{\psi}$ is given by three error angles, which are later used to form a correction quaternion, since attitude representation is done by quaternions. Also the bias-drift is estimated, represented by the GM process state vector $\Delta\boldsymbol{b}_{GM,a}$ and RW state vector $\Delta\boldsymbol{b}_{RW,a}$ for accelerometers.

$$\begin{aligned}\Delta\boldsymbol{x} = \Bigg(& \underbrace{\Delta x_N, \Delta x_E, \Delta x_D}_{\textit{position error}\Delta\boldsymbol{p}}, \underbrace{\Delta v^n_{eb,N}, \Delta v^n_{eb,E}, \Delta v^n_{eb,D}}_{\textit{velocity error}\Delta\boldsymbol{v}^n_{eb}}, \underbrace{\Delta\alpha, \Delta\beta, \Delta\gamma}_{\textit{attitude error}\Delta\boldsymbol{\psi}}, \\ & \underbrace{\Delta b_{GM,ax}, \Delta b_{GM,ay}, \Delta b_{GM,az}}_{\textit{accel GM }\Delta\boldsymbol{b}_{GM,a}}, \underbrace{\Delta b_{RW,ax}, \Delta b_{RW,ay}, \Delta b_{RW,az}}_{\textit{accel RW }\Delta\boldsymbol{b}_{RW,a}} \Bigg)^T\end{aligned} \tag{5.32}$$

The error equations are formulated in terms of error state variables, to form an error state space model.

Position Error

The change of latitude is described by following Equation:

$$\dot{\varphi} = \frac{v^n_{eb,N}}{R_n - h} \tag{5.33}$$

The position error in north direction is basically caused by the velocity error in north direction $\Delta v^n_{eb,N}$ and the height error Δx_D. Under the assumption of constant radii of curvature, the Taylor series expansion of Equation (5.33) leads to

$$\Delta\dot{\varphi} = \frac{\partial\dot{\varphi}}{\partial h}\Delta x_D + \frac{\partial\dot{\varphi}}{\partial v^n_{eb,N}}\Delta v^n_{eb,N} = \frac{v^n_{eb,N}}{(R_n - h)^2}\Delta x_D + \frac{1}{R_n - h}\Delta v^n_{eb,N} \tag{5.34}$$

To calculate the position errors in units of meters, instead of radian, they are converted with the corresponding factors. The advantage is that the position error in meters is much more intuitive and that elements of the measurement covariance matrix do not cover a large value range. This is important, because the inversion of a matrix with values of a scope of $10^{-12} - 200$, in the case of radian, is not trivial. The same methodology is used for the longitudinal error, expressed as east error. The three position error Equations (5.35-5.37) are:

$$\Delta\dot{x}_N = (R_n - h)\Delta\dot{\varphi} = \frac{v^n_{eb,N}}{R_n - h}\Delta x_D + \Delta v^n_{eb,N} \tag{5.35}$$

$$\Delta\dot{x}_E = (R_e - h)cos\varphi\Delta\dot{\lambda} = \frac{v^n_{eb,E}\,tan\varphi}{R_n - h}\Delta x_N + \frac{v^n_{eb,E}}{R_e - h}\Delta x_D + \Delta v^n_{eb,E} \tag{5.36}$$

$$\Delta\dot{x}_D = \Delta v^n_{eb,D} \tag{5.37}$$

Velocity Error

The velocity error in Equation (5.38) is derived from Equation (3.28). The main causes of velocity errors are accelerometer biases and false attitude information for coordinate frame transformation.

$$\Delta\dot{\boldsymbol{v}}^n_{eb} = \boldsymbol{C}^{\hat{n}}_b\Delta\boldsymbol{f}^b_{ib} - [\boldsymbol{C}^{\hat{n}}_b\boldsymbol{f}^b_{ib}\times]\Delta\boldsymbol{\psi} - (2\boldsymbol{\Omega}^n_{ie} + \boldsymbol{\Omega}^n_{en})\Delta\boldsymbol{v}^n_{eb} + [\boldsymbol{v}^n_{eb}\times](2\Delta\boldsymbol{\omega}^n_{ie} + \Delta\boldsymbol{\omega}^n_{en}) \tag{5.38}$$

The relation for the terms $\Delta\boldsymbol{\omega}^n_{en}$, $\Delta\boldsymbol{\omega}^n_{ie}$ is given by Equations (5.39-5.40). The partial vector derivatives are further defined in Appendix B.2.

$$\Delta\boldsymbol{\omega}^n_{en} = \left[\frac{\partial\boldsymbol{\omega}^n_{en}}{\partial(\varphi,\lambda,h)}\right]\left[\frac{\partial(\varphi,\lambda,h)}{\partial(x_N,x_E,x_D)}\right]\Delta\boldsymbol{p} \tag{5.39}$$

$$\Delta\boldsymbol{\omega}^n_{ie} = \left[\frac{\partial\boldsymbol{\omega}^n_{ie}}{\partial(\varphi,\lambda,h)}\right]\left[\frac{\partial(\varphi,\lambda,h)}{\partial(x_N,x_E,x_D)}\right]\Delta\boldsymbol{p} + \left[\frac{\partial\boldsymbol{\omega}^n_{en}}{\partial\boldsymbol{v}^n_{eb}}\right]\Delta\boldsymbol{v}^n_{eb} \tag{5.40}$$

The estimated DCM $\boldsymbol{C}^{\hat{n}}_b$ is calculated from a matrix multiplication. The new DCM $\boldsymbol{C}^{\hat{n}}_n$ for correcting the standard DCM $\boldsymbol{C}^n_b$, contains the estimated attitude errors. An approximation of a DCM like in Equation (5.42) is only valid for small attitude errors (Wendel, 2011). A multiplication of direction cosine matrices is equivalent to a connection of rotations.

$$\boldsymbol{C}_b^{\widehat{n}} = \boldsymbol{C}_n^{\widehat{n}} \boldsymbol{C}_b^n \tag{5.41}$$

$$\boldsymbol{C}_n^{\widehat{n}} = \begin{bmatrix} 1 & -\Delta\gamma & \Delta\beta \\ \Delta\gamma & 1 & -\Delta\alpha \\ -\Delta\beta & \Delta\alpha & 1 \end{bmatrix} \tag{5.42}$$

Attitude Error

The main reasons for attitude errors are uncompensated sensor biases and errors caused by an incorrect compensation of the superimposed earth rotation rate in the angular rate measurement.

$$\Delta\dot{\boldsymbol{\psi}} = -\boldsymbol{\Omega}_{in}^n \Delta\boldsymbol{\psi} - \Delta\boldsymbol{\omega}_{in}^n + \boldsymbol{C}_b^{\widehat{n}} \Delta\boldsymbol{\omega}_{ib}^b \tag{5.43}$$

The effects of earth rotation rate error and transport rate error are combined in $\Delta\boldsymbol{\omega}_{in}^n$.

$$\Delta\boldsymbol{\omega}_{in}^n = \Delta\boldsymbol{\omega}_{ie}^n + \Delta\boldsymbol{\omega}_{en}^n \tag{5.44}$$

Sensor Errors

The only definitions missing in the error equations till now are the sensor errors $\Delta\boldsymbol{\omega}_{ib}^b$ and $\Delta\boldsymbol{f}_{ib}^b$. The measured specific force $\tilde{\boldsymbol{f}}_{ib}^b$ is the sum of real specific force $\boldsymbol{f}_{ib}^b$, a bias $\boldsymbol{b}_a$ and a noise term $\boldsymbol{w}_{WN,a}$.

$$\tilde{\boldsymbol{f}}_{ib}^b = \boldsymbol{f}_{ib}^b + \boldsymbol{b}_a + \boldsymbol{w}_{WN,a} \tag{5.45}$$

The estimated specific force can be written as

$$\widehat{\boldsymbol{f}}_{ib}^b = \tilde{\boldsymbol{f}}_{ib}^b - \boldsymbol{b}_a \tag{5.46}$$

The sensor error $\Delta\boldsymbol{f}_{ib}^b$ is now defined by

$$\Delta\boldsymbol{f}_{ib}^b = \widehat{\boldsymbol{f}}_{ib}^b - \boldsymbol{f}_{ib}^b = -\Delta\boldsymbol{b}_a + \boldsymbol{w}_{WN,a} \tag{5.47}$$

From the AV analysis (Subsection 4.2.3) it is known, that the acceleration bias drift is a combination of first order Gauss-Markov, random walk and white noise process.

$$\Delta\boldsymbol{f}_{ib}^b = -(\Delta\boldsymbol{b}_{GM,a} + \Delta\boldsymbol{b}_{RW,a}) + \boldsymbol{w}_{WN,a} \tag{5.48}$$

Analogous to the derivation of the acceleration error, the gyroscope error can be formulated. Only a WN term is involved.

$$\Delta\boldsymbol{\omega}_{ib}^b = \boldsymbol{w}_{WN,g} \tag{5.49}$$

5.2.2 Error State Space Model

A state space model of form (5.50) can be established from the error equations.

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{F}\Delta\boldsymbol{x} + \boldsymbol{G}\boldsymbol{w} \tag{5.50}$$

A detailed view on the structure of the system is shown in Equation (5.51). How the sub matrices of $\boldsymbol{F}$ are calculated is presented in Appendix B.2. The error state space model is in continuous time, but for discrete Kalman filtering a model in discrete time is required (see Appendix B.1). It is to mention that there exists a further simplified model in Noureldin, Karamat, and Georgy (2012).

$$\overset{\bullet}{\begin{bmatrix} \Delta\boldsymbol{p} \\ \Delta\boldsymbol{v}_{eb}^{n} \\ \Delta\boldsymbol{\psi} \\ \Delta\boldsymbol{b}_{GM,a} \\ \Delta\boldsymbol{b}_{RW,a} \end{bmatrix}} = \begin{bmatrix} \boldsymbol{F}_{11} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{F}_{21} & \boldsymbol{F}_{22} & \boldsymbol{F}_{23} & -\boldsymbol{C}_b^{\widehat{n}} & -\boldsymbol{C}_b^{\widehat{n}} \\ \boldsymbol{F}_{31} & \boldsymbol{F}_{32} & \boldsymbol{F}_{33} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F}_{44} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{p} \\ \Delta\boldsymbol{v}_{eb}^{n} \\ \Delta\boldsymbol{\psi} \\ \Delta\boldsymbol{b}_{GM,a} \\ \Delta\boldsymbol{b}_{RW,a} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{C}_b^{\widehat{n}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_b^{\widehat{n}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_{WN,a} \\ \boldsymbol{w}_{WN,g} \\ \boldsymbol{w}_{GM,a} \\ \boldsymbol{w}_{RW,a} \end{bmatrix} \tag{5.51}$$

The measurements $\Delta\boldsymbol{z}$ of the measurement model (5.52) are the differences of the navigation solution of INS and GPS. There is the possibility to use only position updates, only velocity updates or both. The position differences must be converted from units of radian into meter, since the position error state is in meters too. If velocity updates are used, the measured absolute velocity $\boldsymbol{v}_{abs}$ is either from GPS receiver or odometer. The absolute velocity has be converted into NED-velocity components to form differences.

$$\Delta\boldsymbol{z} = \boldsymbol{H}\Delta\boldsymbol{x} + \boldsymbol{v} \tag{5.52}$$

$$\Delta\boldsymbol{z}_{pos} = \begin{bmatrix} (\widehat{\varphi}_{INS}^{-} - \varphi_{GPS}) \cdot (R_n - \widehat{h}_{INS}^{-}) \\ (\widehat{\lambda}_{INS}^{-} - \lambda_{GPS}) \cdot (R_n - \widehat{h}_{INS}^{-}) cos\widehat{\varphi}_{INS}^{-} \\ \widehat{h}_{INS}^{-} - h_{GPS} \end{bmatrix} \tag{5.53}$$

$$\Delta\boldsymbol{z}_{vel} = \begin{bmatrix} \widehat{v}_{eb,N}^{n-} - v_{abs} cos\psi cos\theta \\ \widehat{v}_{eb,E}^{n-} - v_{abs} sin\psi cos\theta \\ \widehat{v}_{eb,D}^{n-} - (-v_{abs} sin\theta) \end{bmatrix} \tag{5.54}$$

$$\Delta\boldsymbol{z}_{pos,vel} = \begin{bmatrix} (\widehat{\varphi}_{INS}^{-} - \varphi_{GPS}) \cdot (R_n - \widehat{h}_{INS}^{-}) \\ (\widehat{\lambda}_{INS}^{-} - \lambda_{GPS}) \cdot (R_e - \widehat{h}_{INS}^{-}) cos\widehat{\varphi}_{INS}^{-} \\ \widehat{h}_{INS}^{-} - h_{GPS} \\ \widehat{v}_{eb,N}^{n-} - v_{abs} cos\psi cos\theta \\ \widehat{v}_{eb,E}^{n-} - v_{abs} sin\psi cos\theta \\ \widehat{v}_{eb,D}^{n-} - (-v_{abs} sin\theta) \end{bmatrix} \tag{5.55}$$

The error states $\Delta\boldsymbol{x}$ are connected with the residual measurements $\Delta\boldsymbol{z}$ through the measurement matrix $\boldsymbol{H}$. According to which update quantities are used, different measurement matrices are resulting.

$$\underset{(3\times15)}{\boldsymbol{H}_{pos}} = \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix} \tag{5.56}$$

$$\underset{(3\times15)}{\boldsymbol{H}_{vel}} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix} \tag{5.57}$$

$$\underset{(6\times15)}{\boldsymbol{H}_{pos,vel}} = \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix} \tag{5.58}$$

Due to different variations of measurements quantities, the measurement noise covariance matrix $\boldsymbol{R}_k$ has to be adjusted. If only GPS position updates are used the matrix is chosen like in Equation (5.59). The diagonal elements describe the uncertainty in north, east and down direction in meter. For GPS receivers the height standard deviation is larger (Hofmann-Wellenhof, Lichtenegger, and Wasle, 2007) than in latitude or longitude direction. The used GPS receiver does not provide any information on the measurement uncertainty by itself. Variances have to be assumed. Because of this lack in information a diagonal structure is chosen, neglecting the correlation of the GPS navigation solution caused by the internal KF.

$$\boldsymbol{R}_{pos} = \begin{bmatrix}\sigma_{x_N}^2 & 0 & 0\\ 0 & \sigma_{x_E}^2 & 0\\ 0 & 0 & \sigma_{x_D}^2\end{bmatrix} \tag{5.59}$$

If only velocity updates are used, the propagation of uncertainty has be considered, since the NED velocity components are only available with a non-linear transformation (see Equation A.14). According to the law of error propagation (Hoffman-Wellenhof, Legat, and Wieser, 2003), the calculation of the Jacobian matrix $\boldsymbol{J}$ of the transformation vector has to be performed. This yields

$$\boldsymbol{R}_{vel} = \boldsymbol{J}\begin{bmatrix}\sigma_{v_{abs}}^2 & 0 & 0\\ 0 & \sigma_{\psi}^2 & 0\\ 0 & 0 & \sigma_{\theta}^2\end{bmatrix}\boldsymbol{J}^T \tag{5.60}$$

$$\boldsymbol{J} = \begin{bmatrix}cos\psi cos\theta & -v_{abs}sin\psi cos\theta & -v_{abs}cos\psi cos\theta\\ sin\psi cos\theta & v_{abs}cos\psi cos\theta & -v_{abs}sin\psi sin\theta\\ -sin\theta & 0 & -v_{abs}cos\theta\end{bmatrix} \tag{5.61}$$

If both, position and velocity updates are used, the covariance matrix in Equation (5.62) is used.

$$\boldsymbol{R}_{pos,vel} = \begin{bmatrix}\boldsymbol{R}_{pos} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{R}_{vel}\end{bmatrix} \tag{5.62}$$

Now all components of the error state space model are defined to estimate the errors with an ESKF. In the next subsection it is shown how to correct the predicted states with the estimated errors, to get a full state estimate.

5.2.3 Correction Step

In the correction step, the estimated errors correct the INS navigation solution, which can be interpreted as a state prediction. The notation for predicted full states refers therefore to the INS solution. Why these quantities can be interpreted as a prediction will be explained in Section 5.4.

To get an absolute position estimate, the INS position has to be corrected such as in Equations (5.63-5.65). Because the error estimates are in units of meter, a transformation into radian for latitude and longitude is necessary.

$$\widehat{\varphi}^{+} = \widehat{\varphi}_{INS}^{-} - \frac{\Delta\widehat{x}_{N}^{+}}{R_n - \widehat{h}_{INS}^{-}} \tag{5.63}$$

$$\widehat{\lambda}^{+} = \widehat{\lambda}_{INS}^{-} - \frac{\Delta\widehat{x}_{E}^{+}}{(R_e - \widehat{h}_{INS}^{-})cos\widehat{\varphi}_{INS}^{-}} \tag{5.64}$$

$$\widehat{h}^{+} = \widehat{h}_{INS}^{-} - \Delta\widehat{x}_{D}^{+} \tag{5.65}$$

The velocity is estimated by simply subtracting the velocity errors.

$$\widehat{\boldsymbol{v}}_{eb}^{n+} = \widehat{\boldsymbol{v}}_{eb}^{n-} - \Delta\widehat{\boldsymbol{v}}_{eb}^{n+} \tag{5.66}$$

The correction of the attitude is more complex because the attitude error is estimated in terms of error angles. The attitude representation is in terms of quaternions. The correction is done with the formulation of a correction quaternion $\boldsymbol{q}_c$ (Equation 5.67-5.69). To avoid a division by zero in Equation (5.69), in the case of no attitude error, an approximation is used like suggested in Wendel (2011).

$$\boldsymbol{\sigma}_c = -\Delta\widehat{\boldsymbol{\psi}}^{+} \tag{5.67}$$

$$\sigma_c = \sqrt{\boldsymbol{\sigma}_c^T\boldsymbol{\sigma}_c} \tag{5.68}$$

$$\boldsymbol{q}_c = \begin{bmatrix} cos\frac{\sigma_c}{2} \\ \frac{\boldsymbol{\sigma}_c}{\sigma_c}sin\frac{\sigma_c}{2} \end{bmatrix} \approx \begin{bmatrix} 1 - \frac{1}{8}\sigma_c^2 + \frac{1}{384}\sigma_c^4 - \frac{1}{46080}\sigma_c^6 \\ \boldsymbol{\sigma}_c\left(\frac{1}{2} - \frac{1}{48}\sigma_c^2 + \frac{1}{3840}\sigma_c^4 - \frac{1}{645120}\sigma_c^6\right) \end{bmatrix} \tag{5.69}$$

A quaternion multiplication is performed to correct the attitude.

$$\widehat{\boldsymbol{q}}_b^{n+} = \boldsymbol{q}_c \bullet \widehat{\boldsymbol{q}}_b^{n-} \tag{5.70}$$

5.3 Loosely-Coupled RISS Integration

The LC RISS integration is very similar to the LC strapdown integration. The indirect approach with error state formulation is used here as well. The computation algorithm, later presented in Section 5.4, is also the same for this method. The difference is another definition of the error state vector, since not all inertial sensors are used and odometer measurements are involved. This approach is based on the work of Noureldin, Iqbal, and Okou (2008) and Noureldin, Karamat, and Georgy (2012).

The error state vector for the LC RISS integration is defined in Equation (5.71). The position error $\Delta \boldsymbol{p}_k$ is in units of radian for latitude and longitude errors. The velocity error $\Delta \boldsymbol{v}_k^{ENU}$ is given in the ENU frame convention. The last three error states are summarized in the vector $\Delta \boldsymbol{e}_k$, containing the yaw error $\Delta\psi_k$, odometer scale factor error Δv_k^{SF} and z-gyroscope error $\Delta b_{k,gz}$. Note that only the yaw error is estimated. It is not necessary to estimate pitch and roll. They are calculated from specific force measurements without any integration process causing a drift over time.

$$\Delta \boldsymbol{x}_k = \Bigg(\underbrace{\Delta\varphi_k, \Delta\lambda_k, \Delta h_k,}_{position\, error \Delta \boldsymbol{p}_k} \underbrace{\Delta v_k^E, \Delta v_k^N, \Delta v_k^U}_{velocity\, error \Delta \boldsymbol{v}_k^{ENU}}, \quad \underbrace{\Delta\psi_k, \Delta v_k^{SF}, \Delta b_{k,gz}}_{yaw/odometer/z-gyro\, error \Delta \boldsymbol{e}_k} \quad \Bigg)^T \tag{5.71}$$

It is to mention that no scale factor error for the wheel-speed sensors is modeled, because no test measurement data was available. The state Δv_k^{SF} is therefore not used, but included if needed for further researches. Noureldin, Karamat, and Georgy (2012) suggest a first order GM process to model the scale factor error.

5.3.1 Error State Space model

The structure of the error state space model in discrete form (Equation 5.72), is shown in Equation (5.73). The sub matrices are summarized by Equation (5.74-5.76).

$$\Delta \boldsymbol{x}_k = \boldsymbol{F}_{k-1} \Delta \boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1} \tag{5.72}$$

$$\begin{bmatrix} \Delta \boldsymbol{p}_k \\ \Delta \boldsymbol{v}_k^{ENU} \\ \Delta \boldsymbol{e}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{F}_{k-1,12} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{F}_{k-1,23} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F}_{k-1,33} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{p}_{k-1} \\ \Delta \boldsymbol{v}_{k-1}^{ENU} \\ \Delta \boldsymbol{e}_{k-1} \end{bmatrix} + \begin{bmatrix} w_{k-1,\Delta \boldsymbol{p}} \\ w_{k-1,\Delta \boldsymbol{v}} \\ w_{k-1,\Delta \boldsymbol{e}} \end{bmatrix} \tag{5.73}$$

$$\boldsymbol{F}_{k-1,12} = \begin{bmatrix} 0 & \frac{1}{R_n + h_{k-1}} \Delta t & 0 \\ \frac{1}{(R_e + h_{k-1}) cos\varphi_{k-1}} \Delta t & 0 & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \tag{5.74}$$

$$\boldsymbol{F}_{k-1,23} = \begin{bmatrix} v_{k-1}^{od} cos\psi_{k-1} cos\theta_{k-1} & v_{k-1}^{od} sin\psi_{k-1} cos\theta_{k-1} & 0 \\ -v_{k-1}^{od} sin\psi_{k-1} cos\theta_{k-1} & v_{k-1}^{od} cos\psi_{k-1} cos\theta_{k-1} & 0 \\ 0 & v_{k-1}^{od} sin\theta_{k-1} & 0 \end{bmatrix} \tag{5.75}$$

$$\boldsymbol{F}_{k-1,33} = \begin{bmatrix} 1 & 0 & -\Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.76}$$

The measurement model has the same structure as in Section 5.2.2. Depending on what GPS measurement quantities are used, the residuals (Equation 5.77-5.79) and the measurement matrices (Equation 5.80-5.82) differ. Because the plane position errors are in units of radian, no correction factor is needed like in Equation (5.53).

$$\Delta \boldsymbol{z}_{k,pos} = \begin{bmatrix} \widehat{\varphi}_{RISS}^{-} - \varphi_{GPS} \\ \widehat{\lambda}_{RISS}^{-} - \lambda_{GPS} \\ \widehat{h}_{RISS}^{-} - h_{GPS} \end{bmatrix} \tag{5.77}$$

$$\Delta \boldsymbol{z}_{k,vel} = \begin{bmatrix} \widehat{v}_{RISS}^{E-} - v_{GPS} sin\psi cos\theta \\ \widehat{v}_{RISS}^{N-} - v_{GPS} cos\psi cos\theta \\ \widehat{v}_{RISS}^{U-} - v_{GPS} sin\theta \end{bmatrix} \tag{5.78}$$

$$\Delta \boldsymbol{z}_{k,pos,vel} = \begin{bmatrix} \widehat{\varphi}_{RISS}^{-} - \varphi_{GPS} \\ \widehat{\lambda}_{RISS}^{-} - \lambda_{GPS} \\ \widehat{h}_{RISS}^{-} - h_{GPS} \\ \widehat{v}_{RISS}^{E-} - v_{GPS} sin\psi cos\theta \\ \widehat{v}_{RISS}^{N-} - v_{GPS} cos\psi cos\theta \\ \widehat{v}_{RISS}^{U-} - v_{GPS} sin\theta \end{bmatrix} \tag{5.79}$$

$$\underset{(3\times 9)}{\boldsymbol{H}_{k,pos}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{5.80}$$

$$\underset{(3\times 9)}{\boldsymbol{H}_{k,vel}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \tag{5.81}$$

$$\underset{(6\times 9)}{\boldsymbol{H}_{k,pos,vel}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \tag{5.82}$$

5.3.2 Correction Step

The correction step in this approach is very simple. The estimated errors have to be subtracted from the predicted values of the RISS mechanization. No further transformation, like in the strapdown integration, is needed.

$$\widehat{\boldsymbol{p}}_k^{+} = \widehat{\boldsymbol{p}}_{k,RISS}^{-} - \Delta \widehat{\boldsymbol{p}}_k^{+} \tag{5.83}$$

$$\widehat{\boldsymbol{v}}_k^{+} = \widehat{\boldsymbol{v}}_{k,RISS}^{-} - \Delta \widehat{\boldsymbol{v}}_k^{ENU+} \tag{5.84}$$

$$\widehat{\psi}_k^{+} = \widehat{\psi}_{k,RISS}^{-} - \Delta \widehat{\psi}_k^{+} \tag{5.85}$$

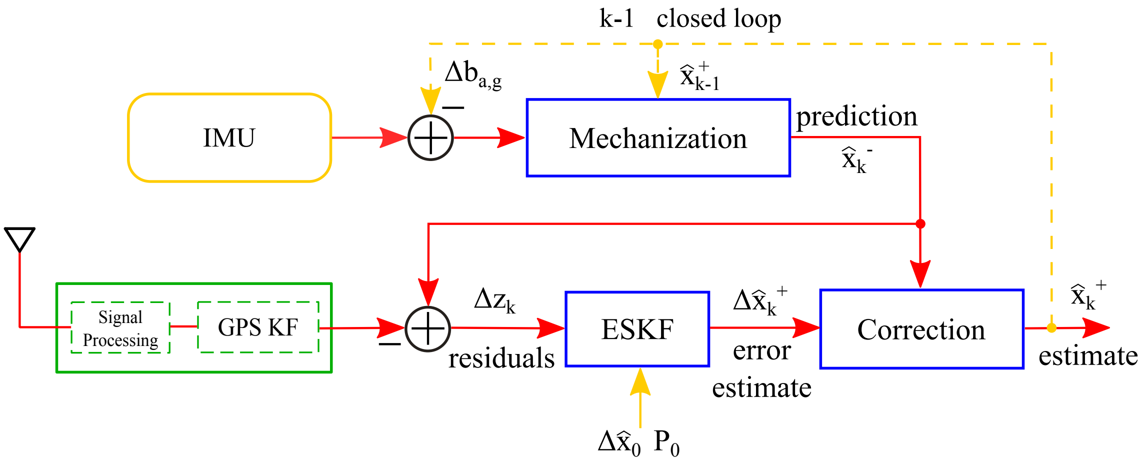

Figure 5.1: Block diagram of a loosely-coupled system in indirect formulation

5.4 Loosely-Coupled Implementation

This Section explains in detail how the fusion algorithm works and how the particular parts come together. The here presented principles are valid for both, the full IMU and the RISS method. Figure 5.1 is showing a block diagram of a loosely-coupled system in indirect formulation.

The closed loop ESKF is indicated with the feedback path. In this configuration the mechanization navigation solution can be interpreted as a prediction of the full states, like in an EKF, where the non-linear system model is used in the prediction step. The error estimates are used to estimate the full states. The structure of Figure 5.1 resembles that of an EKF (see Subsection 5.1.2). This will become more clear in the next Subsection.

Subsection 5.4.1 deals with the problem of different sampling rates of the sensors. The implemented system is therefore a multi-rate system. In Subsection 5.4.2 some issues are addressed regarding the measurement data.

5.4.1 Multi-rate System

The IMU and wheel-speed sensors have the same sampling rate of 50Hz. The GPS receiver instead, has a sampling rate of 1Hz. The conventional KF algorithm has to be adapted, because only in every 50th iteration, a measurement update is available. Taking a look at the KF equation, one can see that the prediction step is independed of measurement information. The idea is to compute the error covariance prediction matrix $\boldsymbol{P}_k^-$ in every iteration, whereas the other KF steps are only computed if a measurement is available. The period where no GPS measurement is available, is called prediction mode, because only predicted values from the mechanization process are obtained. Figure 5.2 shows a graphical visualization of the computation procedure.

While the algorithm is in prediction mode, the predicted quantities ($\boldsymbol{P}_k^-$ and $\hat{\boldsymbol{x}}_k^-$) are assigned to the estimated values ($\boldsymbol{P}_k^+$ and $\hat{\boldsymbol{x}}_k^+$) to continue with the iteration loop. With this method it is even possible to overcome GPS outages. It is no longer necessary that

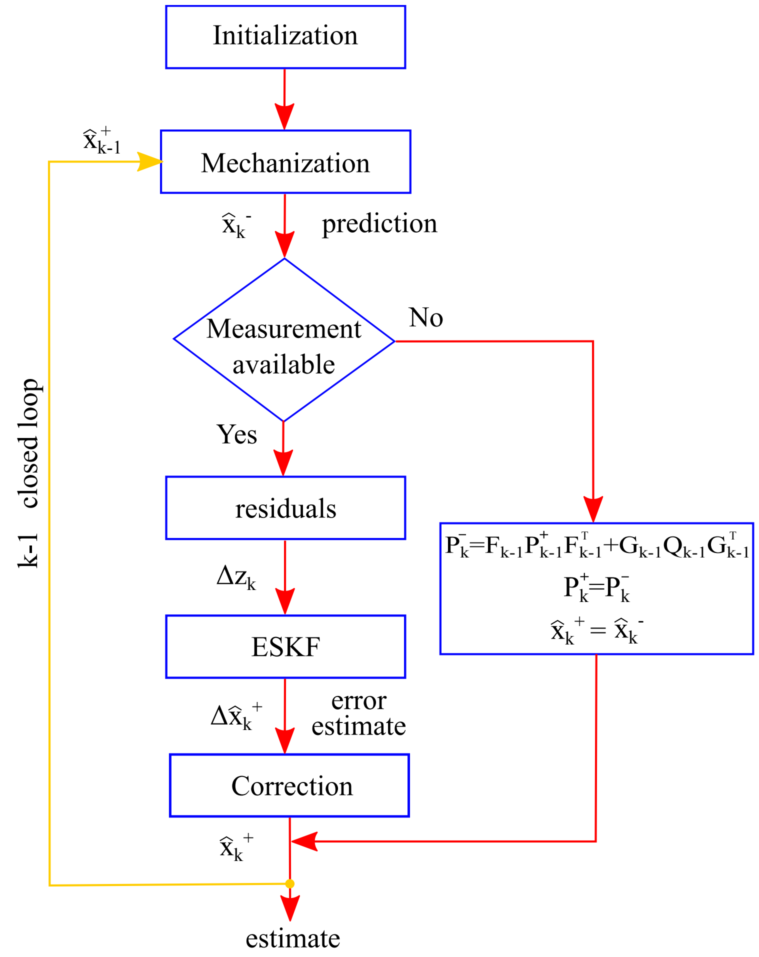

Figure 5.2: Flow chart of the GPS/INS integration algorithm

a GPS measurement is available every second. During prediction mode, the uncertainty, represented by $\boldsymbol{P}_k^-$, is growing with time. If the GPS signal is blocked for a long period of time, the navigation solution will eventually diverge.

In the following the particular steps of the ESKF are presented. If a measurement update is available, Equations (5.86-5.89) are computed. Like mentioned in previous Sections, with the closed loop configuration, a prediction step of the error states is not necessary. The the error states of the ESKF must be set to zero after the correction step. Then there is no old estimated error $\Delta\hat{\boldsymbol{x}}^+_{k-1}$, that could be used for prediction, like in a conventional KF.

The performed operations embedded within a structure like in Figure 5.1, are comparable to an EKF approach (see Subsection 5.1.2). For the calculation of the estimated error covariance matrix $\boldsymbol{P}_k^+$, the *Joseph* form is suggested by Noureldin, Karamat, and Georgy (2012). Equation (5.11) is a simplification under optimal conditions. A small error in the calculation of $\boldsymbol{K}_k$ (Equation 5.87), could lead to very large errors when using Equation (5.11).

$$\boldsymbol{P}_k^- = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{F}_{k-1}^T + \boldsymbol{G}_{k-1}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^T \tag{5.86}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \tag{5.87}$$

$$\Delta\widehat{\boldsymbol{x}}_k^+ = \boldsymbol{K}_k\Delta\boldsymbol{z}_k \tag{5.88}$$

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^-(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \tag{5.89}$$

If no measurement update is available, Equations (5.90-5.92) are computed by the ESKF. These are the same Equations like in Figure 5.2.

$$\boldsymbol{P}_k^- = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{F}_{k-1}^T + \boldsymbol{G}_{k-1}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^T \tag{5.90}$$

$$\Delta\widehat{\boldsymbol{x}}_k^+ = \Delta\widehat{\boldsymbol{x}}_k^- \tag{5.91}$$

$$\boldsymbol{P}_k^+ = \boldsymbol{P}_k^- \tag{5.92}$$

5.4.2 Time Synchronization

The test measurements were performed with a test vehicle, equipped with different measurement systems. The data of interest for this project are IMU, GPS and wheel-speed measurements. The data is transmitted and stored via CAN-Bus. This Subsection addresses some signal properties that have to be considered in the implementation.

Varying Sampling Rate

The proposed sampling rate of 50Hz, for IMU and wheel-speed sensors, is not exactly constant. The sampling time of $\Delta t = \frac{1}{50Hz} = 0.02s$ is slightly varying. Compare Figure 5.3 and 5.4 to see the different behavior of sampling rates.

Such an issue is often encountered when working with real sensors measurements. But for simulation purposes in discrete time, one synchronized time vector is needed. The wheel-speed measurements are linear interpolated to match with the time stamp of the IMU measurement. The same problem arises with the used low-cost GPS receiver, whose sampling rate is not constant as well.

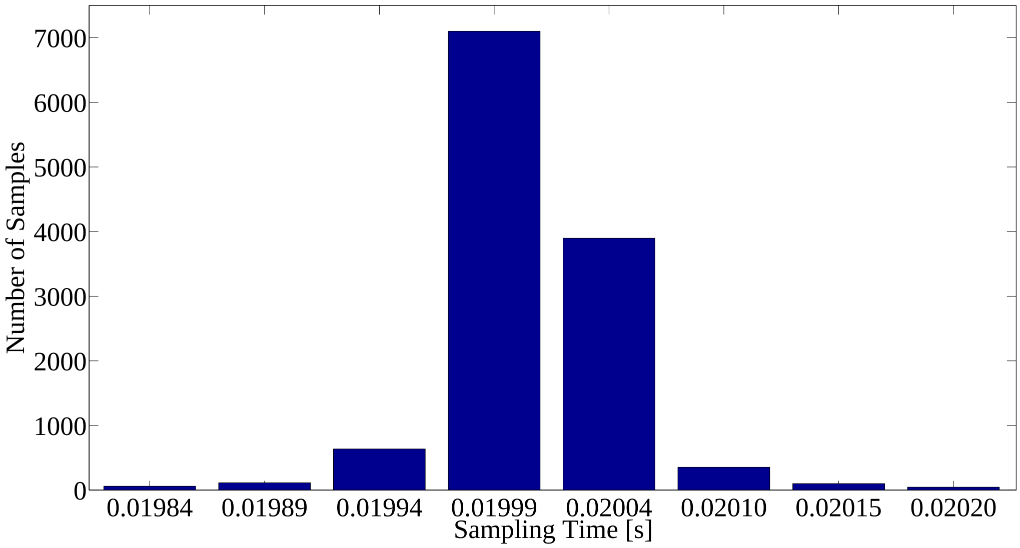

Figure 5.3: Varying 50Hz sampling rate for IMU measurements

Another characteristic of the GPS measurements is revealed by comparing the GPS velocity with the wheel-speed data. Figure 5.5 shows a comparison of both signals. This leads to the conclusion that the GPS velocity measurement is delayed. It may be expected that the GPS position is delayed as well. Such a delay causes problems within the GPS/INS integration, because the INS navigation solution, resulting from the IMU measurements, does not correspond with the aiding GPS signal. In the post processing a time delay can be corrected, but not for real time applications. The causes may be found in the internal signal processing of the GPS receiver and the subsequently CAN-Bus transmission.

Similar problems with CAN-Bus networks are encountered in the dissertation by Magnusson and Odenman (2012).

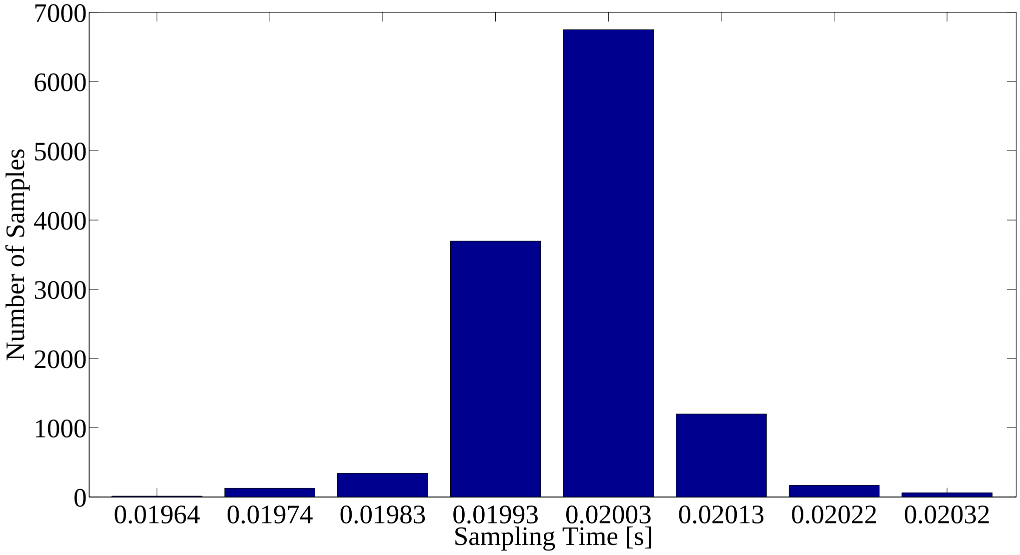

Figure 5.4: Varying 50Hz sampling rate for wheel-speed measurements

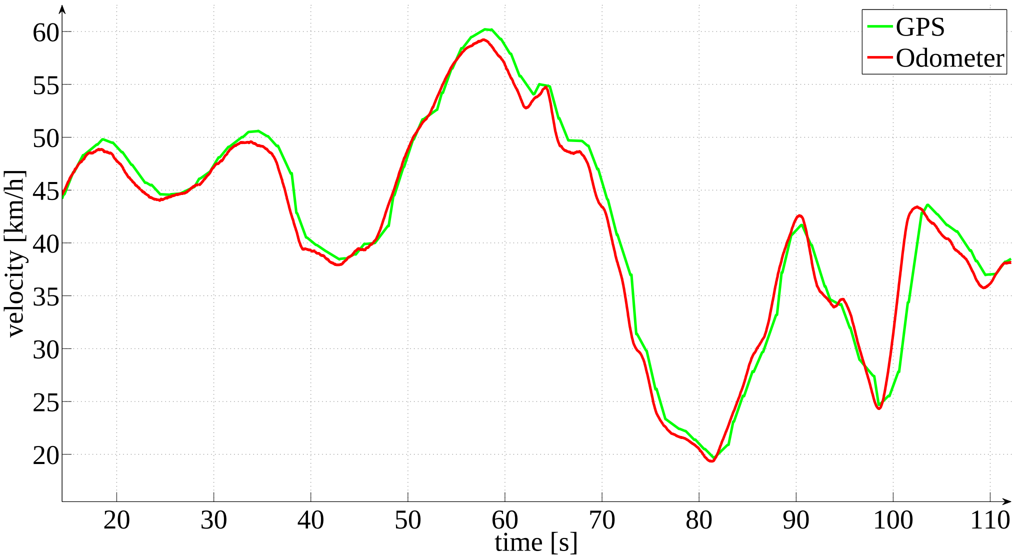

Figure 5.5: Comparison of measured GPS and Odometer velocity

Chapter 6

Simulation Results

This Chapter compares the performance of the two implemented LC methods. The abbreviation *LC SD* refers to the loosely-coupled strapdown approach, whereas *LC RISS* labels the loosely-coupled reduced inertial sensor system approach. Different estimated quantities are of interest, like attitude, height and primarily the position in the plane. One of the benefits of GPS/INS integration is the increased redundancy. The behavior during GPS outages is therefore analyzed as well. To demonstrate the differences of the SDINS and the RISS approach, first of all the navigation solutions of the mechanization processes, without any GPS aiding, is shown. Section6.6 discusses the results presented by this Chapter in more detail and provides conclusions for the simulation results.

This thesis is based on prior taken measurement data sets. The measurement itself was not part of this thesis. The derived algorithms and results in this work are not tested or validated under real time conditions. Only a post-processing analysis is done. The presented results in this Chapter are achieved through computational simulation in MATLAB. The measurements are taken in an rural area on particular test tracks. The road trajectories contain many turns, some hills and cover a distance of about 4-5 kilometers in 4-6 minutes.

Used Sensors

The carried out measurements and measurement data sets, which are the basis for this thesis, are provided by partner companies of the iCOMPOSE project. From the measurement setup is known that the sensor signals are transmitted via internal CAN-Bus of the car. This will have some effects on the measurements, which is discussed in Section 5.4.2.

All GPS measurements in this thesis are made with the low-cost Navilock NL-302U USB GPS receiver (NAVIlock, 2015).

The IMU, for INS mechanization, is a IG-500A sub-miniature Attitude Heading Reference System (AHRS) from SBG systems. This is a 9-DOF, MEMS technology based sensor, that uses a set of 3 gyroscopes, accelerometers and magnetometers to detect 3D motion. Thanks to the integrated EKF, which performs internal sensor fusion, the sensor can give drift-less and accurate 3D orientation (SBG systems, 2011).

6.1 SDINS and RISS Mechanization

To demonstrate the differences of the navigation solutions provided by the completely different SDINS and RISS approaches, a comparison of simulated tracks and heights is presented. As initial attitude in terms of Euler angles, the first angles provided by the AHRS sensor are chosen. The initial velocity is assumed as zero, since the vehicle is not moving at the beginning of the measurement. The initial position is provided by the GPS receiver.

Figure 6.1 compares the two simulated tracks without GPS aiding. A drift over time for both solutions is observed. The drift for the SDINS is much larger, leading to an divergent navigation solution after a few seconds. With the RISS approach on the other hand, the the test track is recognizable. Only a part of the simulated SDINS track is shown for better comparison with the RISS mechanization.

For RISS mechanization, signals from four wheel-speed sensors, each for every wheel, are available. The odometer velocity is chosen as the arithmetic mean of all four wheel-speed measurements.

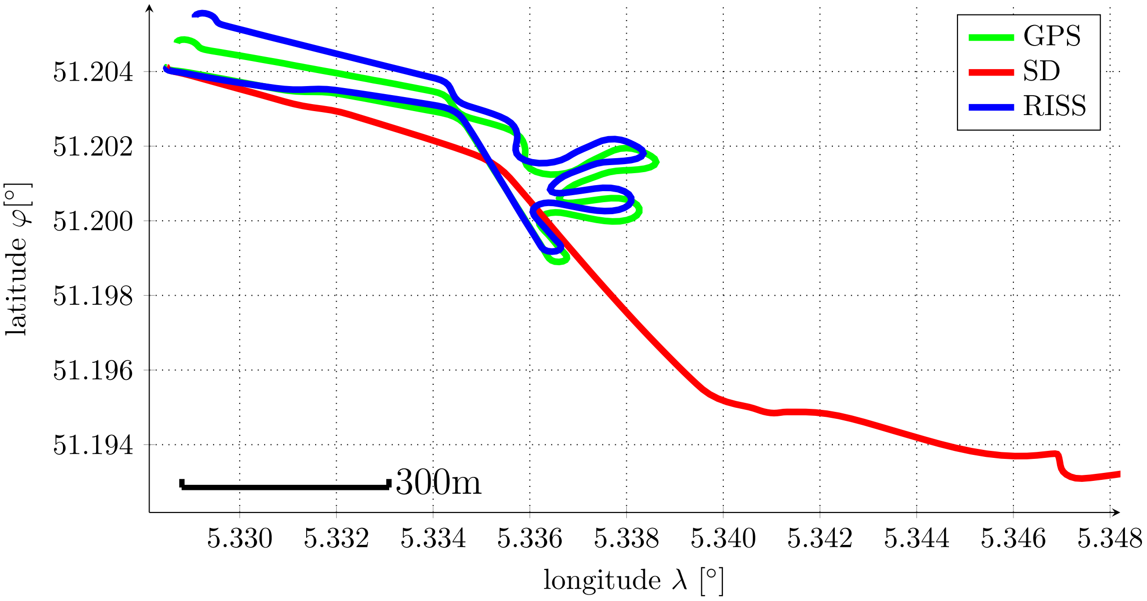

Figure 6.1: SD and RISS track comparison

The comparison of heights shows similar behavior as for tracks. In Figure 6.2 a diverging SDINS height is clearly visible. The RISS mechanization is able to provide a good height simulation without any drift over time.

6.2 Attitude

The attitude is compared in terms of Euler angles. The heading, represented by the yaw angle is compared based on four different data sources. These sources are the estimates

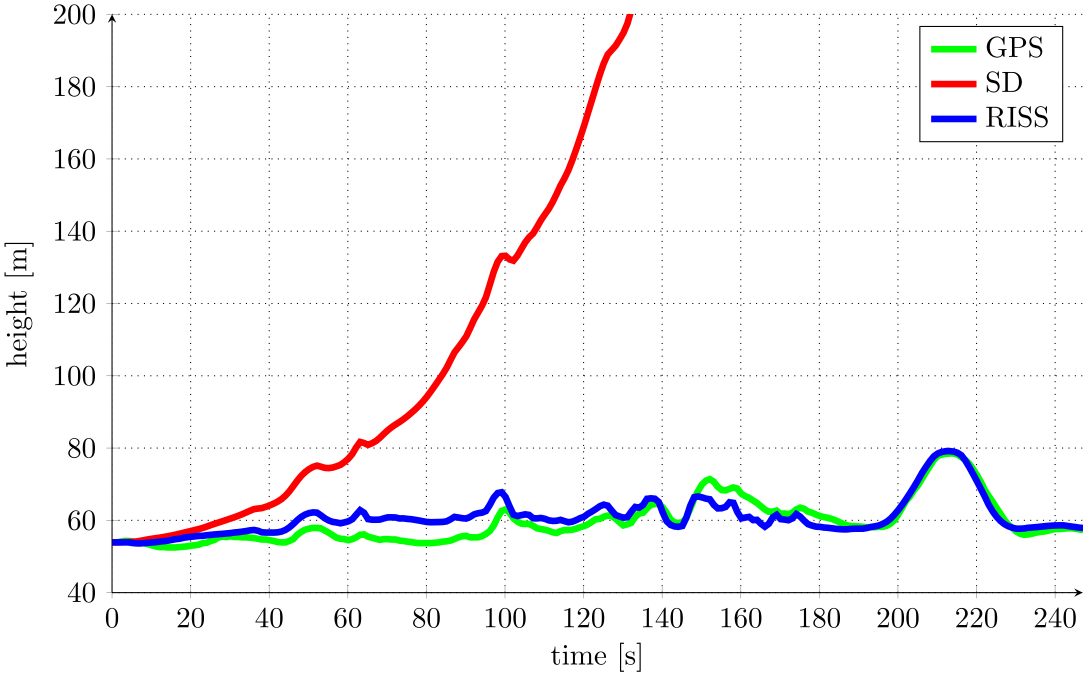

Figure 6.2: SD and RISS height comparison

of the two implemented methods in this thesis, the heading of the GPS receiver and the yaw angle provided by the AHRS sensor. For pitch and roll angle comparison, no GPS quantity is available. The initial attitude in terms of Euler angles is given by the first AHRS measurement.

6.2.1 Heading

Figure 6.3 shows the simulated results, compared to the GPS measurement and AHRS estimates. The AHRS yaw angle is the only one that is significantly different. The estimates of the two implemented methods are almost identical, even thought they are acquired through completely different approaches for yaw calculation (compare Equation 3.14 and 3.42).

At the beginning of the measurement the GPS heading takes some time until it is stabilized. No further information is available how the GPS receiver calculates the heading. A simple calculation via the direction of traveled path or a built in compass is conceivable.

The angles of the implemented LC methods are reliable estimates of the yaw angle. The low-cost GPS receiver is also able to give a good heading estimation. The AHRS sensor can not ensure the promised accurate heading. Based on the orientation the car should have had, according to the measured track, it can be examined, that the AHRS heading is not reliable.

It is worth mentioning that the *magnetic declination* has to be considered in navigation applications. Devices which are using compasses or magnetometer are only able to determine the heading with *magnetic north* as reference. Magnetic north and *true north*

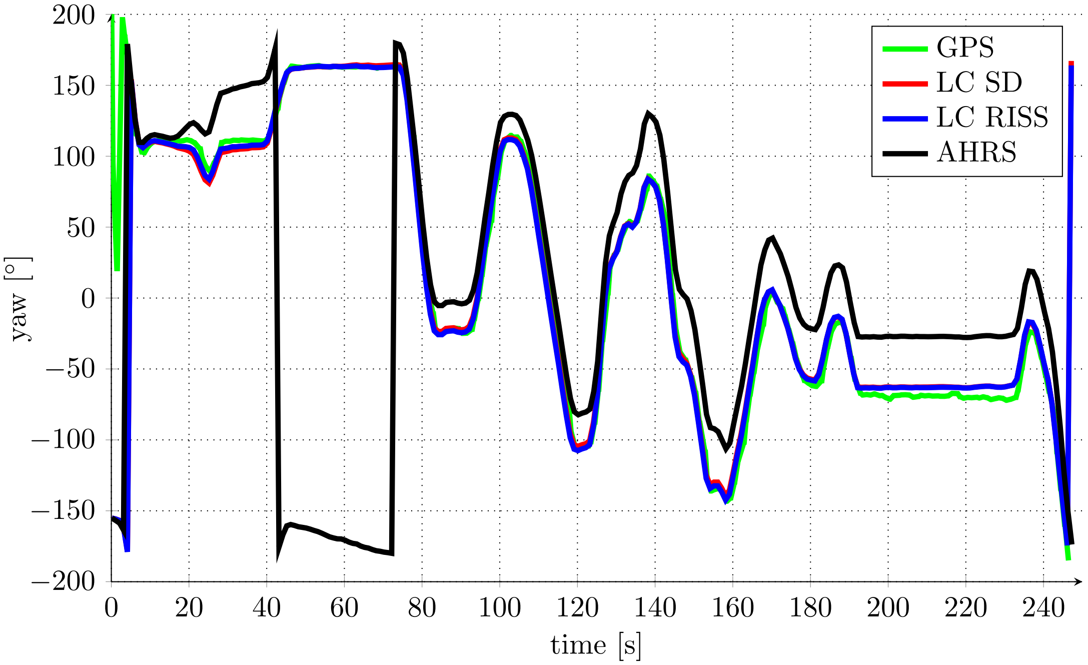

Figure 6.3: Comparison of different headings

are not aligned. The difference between magnetic north and true north direction is called magnetic declination. It depends on the position on earth and is slowly varying over time, because the location of the magnetic north pole is varying. For navigation applications which are using the ENU or NED frame, the true north direction is important. The declination is $< 1.05°$ (NCEI, 2015) at the position where the test measurements are taken ($\varphi \approx 51.19°, \lambda \approx 5.32°$). The AHRS sensor offers the option to consider the declination, which was enabled during the measurements.

6.2.2 Pitch and Roll

The estimated pitch and roll angle from *LC SD* and *LC RISS* are compared with the angles provided by the AHRS sensor in Figure 6.4 and 6.5.

For the calculation of the pitch angle the acceleration, measured by the odometer, is needed. The acceleration is the derivative of the velocity with respect to time. If the velocity measurement is noisy, a numerical derivative will increase the noise in the acceleration signal. The effects are direct noticeable in pitch and roll angle, too. Therefore the odometer velocity is smoothed with a causal moving average filter.

For the sake of a better comparison of pitch and roll angles, the LC RISS angles are smoothed with an moving average filter (for visualization only).

Normally it is not possible to validate the results for pitch and roll angle. Nevertheless one idea to validate the pitch estimates is to check the resulting heights of the RISS mechanization, if an external pitch angle is used, instead of the proposed one in Equation (3.40). The height is used as an indicator for the accuracy of pitch estimation, since the height

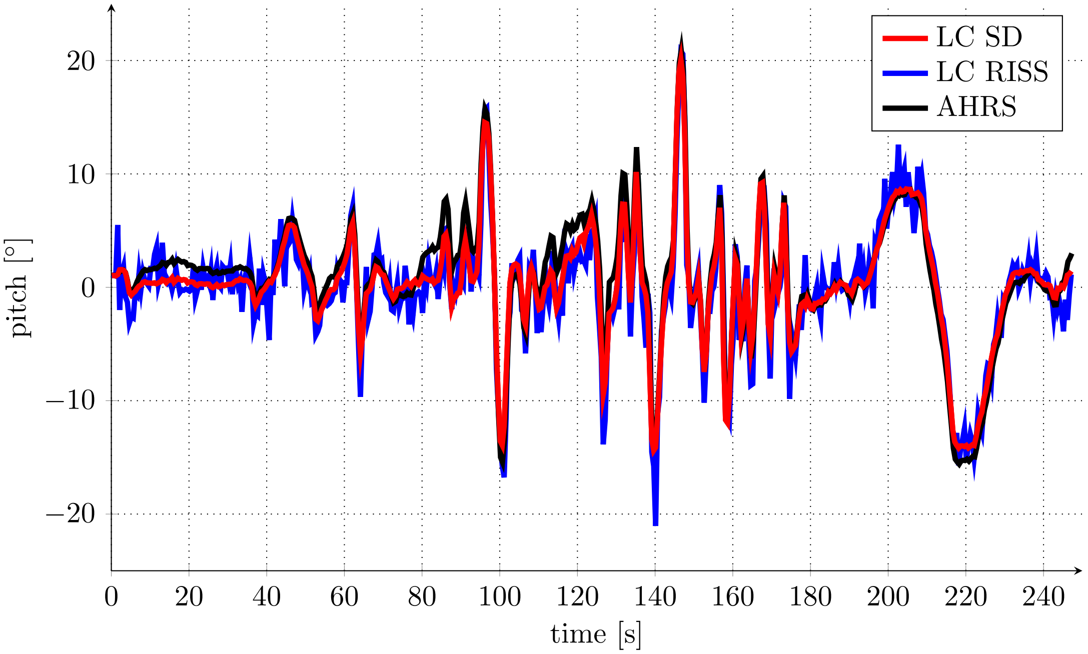

Figure 6.4: Comparison of pitch angles

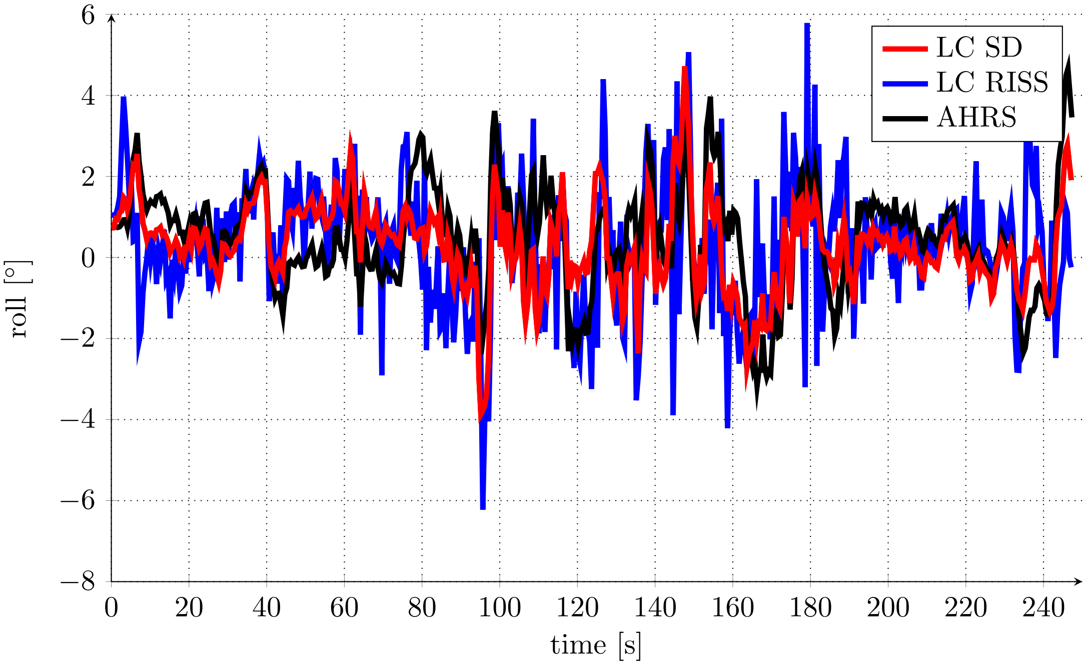

Figure 6.5: Comparison of roll angles

in RISS mechanization depends only on the odometer velocity and the pitch angle (see Equation 3.47).

Figure 6.6 shows the different simulated heights and the measured GPS height. As one can see the height of the RISS mechanization with internally calculated pitch, does not drift. All other solutions tend to drift over time, especially the height calculated with the pitch angle provided by the AHRS. A part of the drift could also be caused by a bias in the velocity measurement. However the pitch provided by the AHRS does not seem reliable, even though the difference to the other two pitch estimates is small.

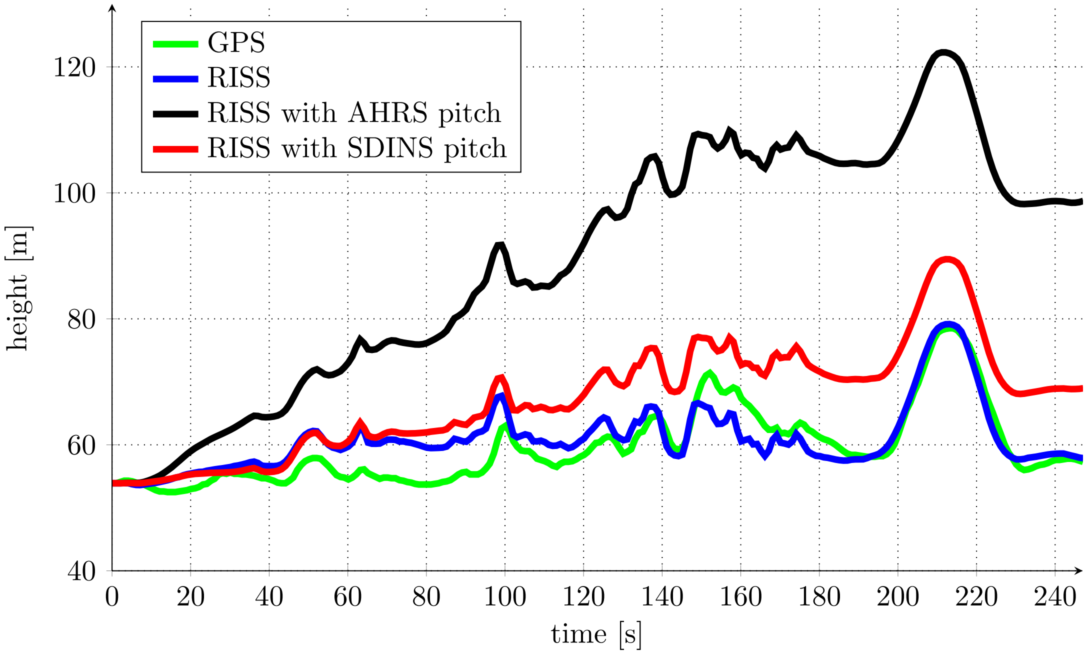

Figure 6.6: Comparison of RISS heights based on internal pitch, AHRS pitch and SDINS pitch

6.3 3D Position

In this Section the position simulation results of the *LC SD* and *LC RISS* are presented. One purpose of GPS/INS integration is to increase the accuracy of the absolute position estimation. Since no highly accurate GPS reference position measurement is available for this thesis, no reliable statement can be made if the accuracy really increased. In order to still validate if the new estimated position is more reasonable than the low-cost GPS measurement, a comparison of tracks with satellite plot overlay is chosen. Additionally error ellipsis are used to visualize the uncertainty of the estimated positions.

Another purpose of GPS/INS integration is to increase redundancy, e.g. to overcome GPS outages by providing a continuous navigation solution. Section 6.5 considers the results regarding GPS outages.

6.3.1 Track

The simulation of different tracks is shown in Figure 6.7 and 6.8. It is observed that at several parts of the tracks the GPS/INS estimated positions are more reliable than the GPS measurement, whereas there are also sections where the LC estimates are off the road. In general, the *LC RISS* solution tends to provide a more accurate position than the *LC SD*.

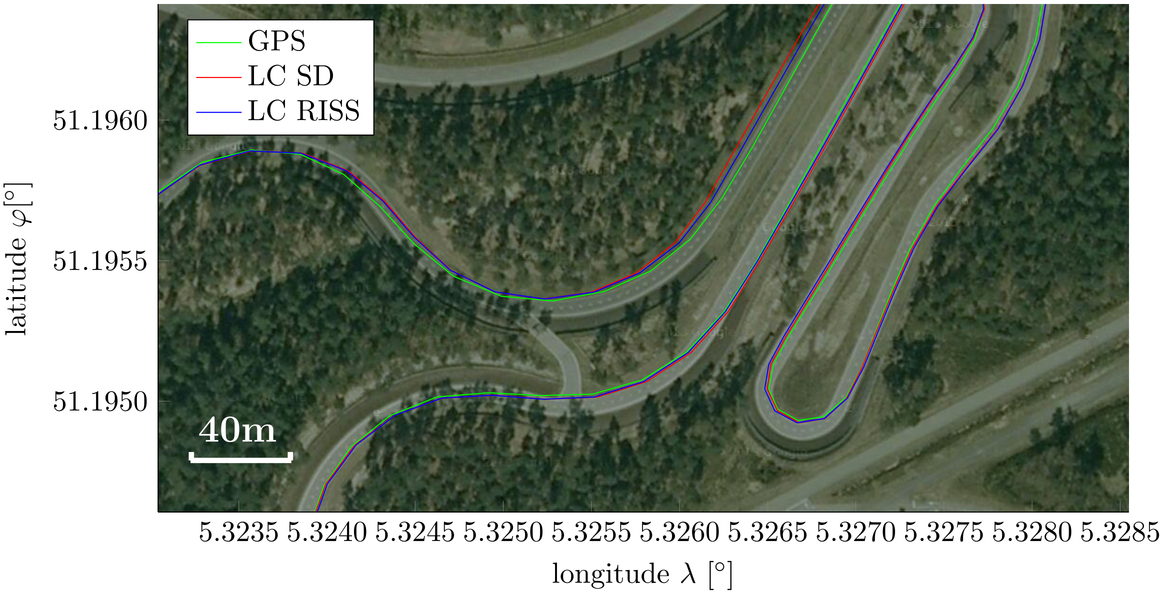

Figure 6.7: *LC SD* and *LC RISS* track comparison [1]

Even if no statement about the absolute position accuracy is possible, the uncertainty of the estimation can be displayed. The error ellipsis show the 95% confidence interval of the GPS measurement and the LC system estimation. The error ellipsis are calculated from the covariance matrix $\boldsymbol{R}_k$ corresponding to the GPS receiver and the error covariance matrix $\boldsymbol{P}_k^+$ from the ESKF estimate. In Figure 6.9 and 6.10 the error ellipsis of both implemented GPS/INS integrations are visualized. As a result of the applied fusion algorithm the uncertainty is reduced, compared to the GPS measurement alone.

6.3.2 Height

The two fusion methods are able to estimate the 3D position, thus also height. Figure 6.11 compares the different heights. The estimated *LC RISS* height is very similar to the measured GPS height. The *LC SD* on the other hand shows a different behavior. Through simulation it is observed that the *LC SD* height tends to drift over time, even with GPS aiding. The reason could be an uncompensated bias in the inertial sensors, causing an incorrect transformation into the navigation frame and resulting in a false gravity compensation. The bias-drift is modeled and compensated like discussed in Chapter 4, but a constant bias sensor error can not be identified from the static measurements, without

[1] Google Imagery©2015 Aerodata International Surveys,DigitalGlobe

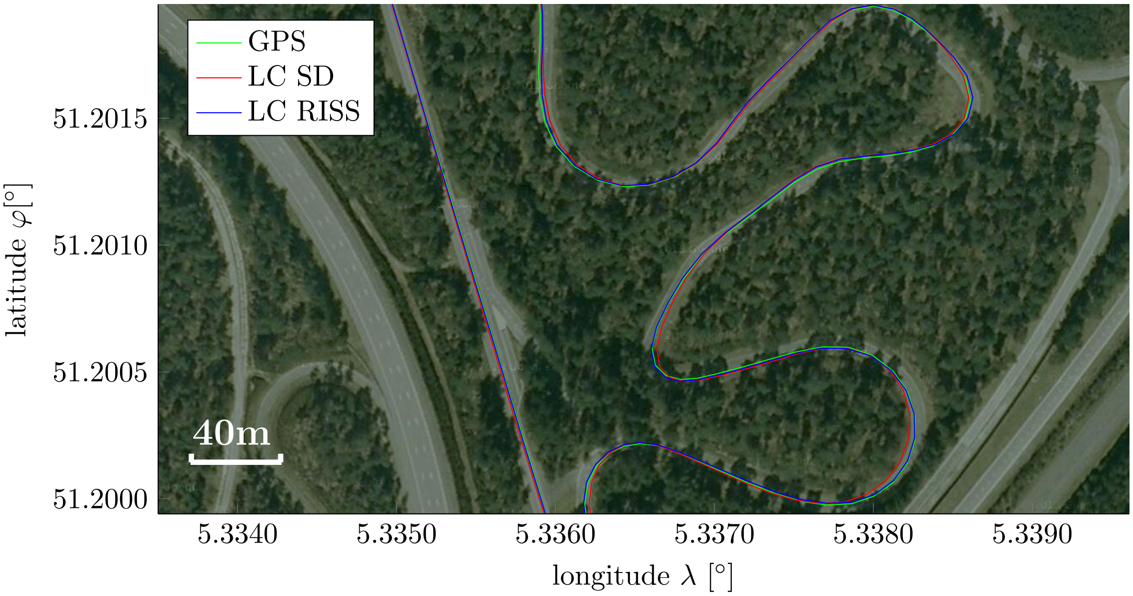

Figure 6.8: *LC SD* and *LC RISS* track comparison

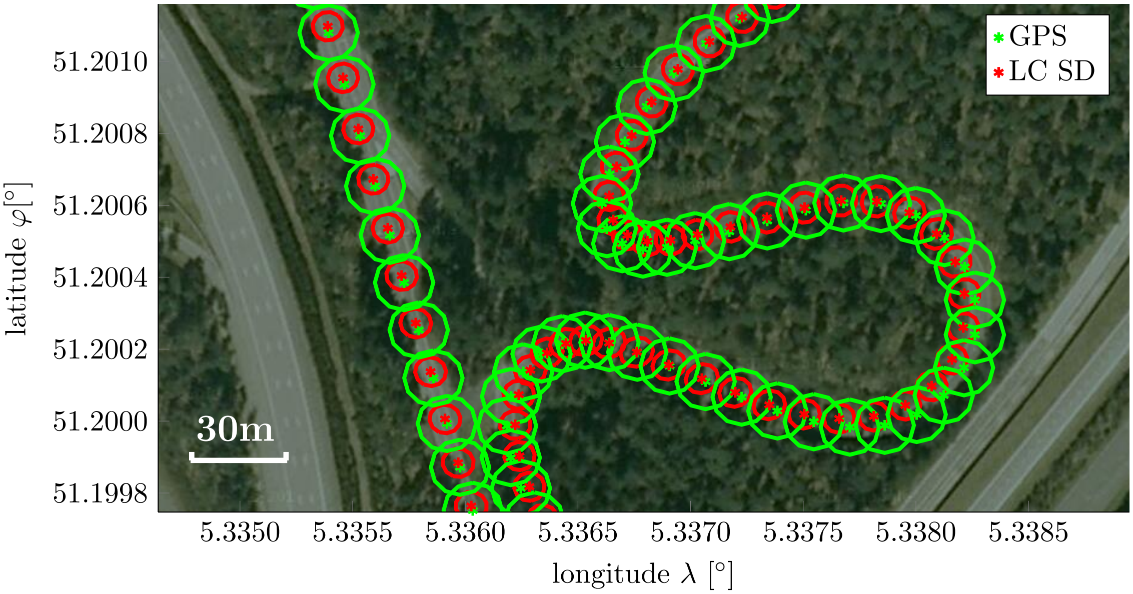

Figure 6.9: *LC SD* error ellipsis with 95% confidence interval

exact knowledge about the attitude at that time. In Figure 6.11 the drift is minimized by correcting an estimated bias in the z-accelerometer, found with trial and error method.

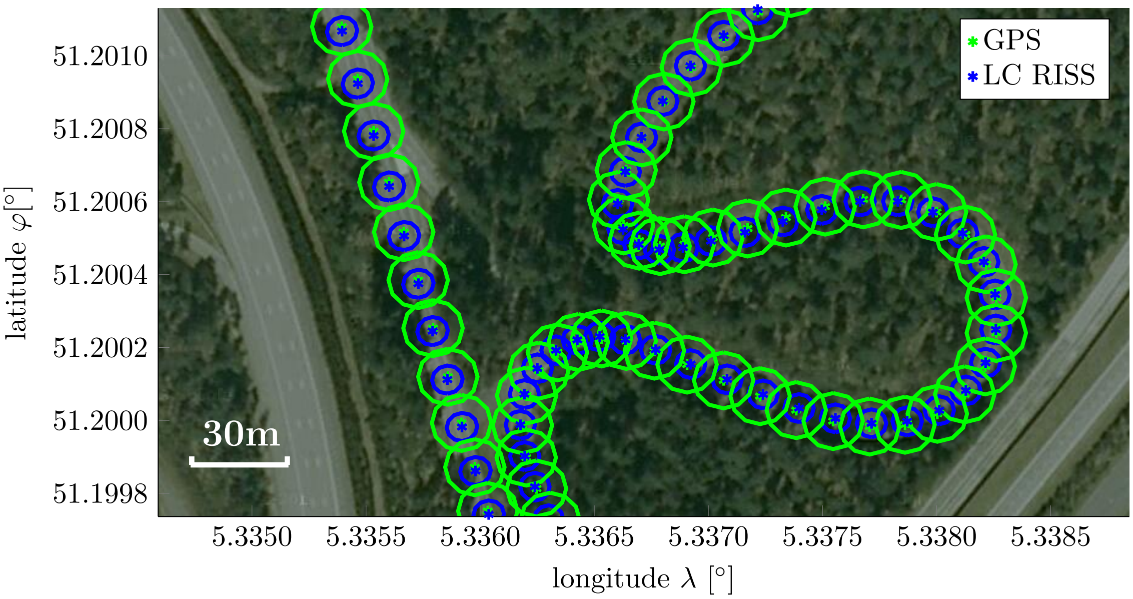

Figure 6.10: *LC RISS* error ellipsis with 95% confidence interval

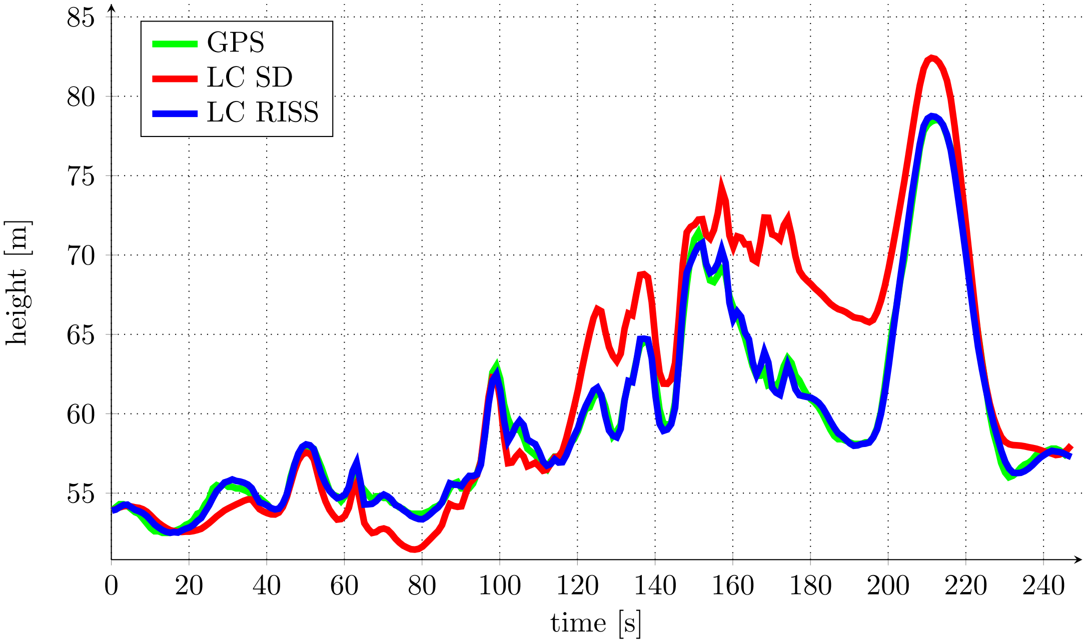

Figure 6.11: Comparison of *LC SD* and *LC RISS* heights

6.4 Estimated Velocity Errors

Comparing the performance of the two implemented methods, it is also of interest to compare the estimated errors. The velocity errors are chosen for comparison, because characteristics of the two integration methods can be demonstrated. To compare velocity errors of the ENU and NED frame, the up velocity errors Δv_k^{U+} are inverted.

In Figure 6.12 the estimated velocity errors are presented. The estimated *LC RISS* velocity errors are generally smaller than the *LC SD* velocity errors, for north and east direction. This is expected, since the RISS mechanization uses more accurate absolute velocity measurements, instead of acceleration measurements. The down velocity errors for the *LC RISS* approach are greater than the *LC SD* quantity. This could be caused by the strong varying pitch angle (see Figure 6.4) from the RISS mechanization, which is used for the calculation of the up velocity v_k^U (see Equation 3.47).

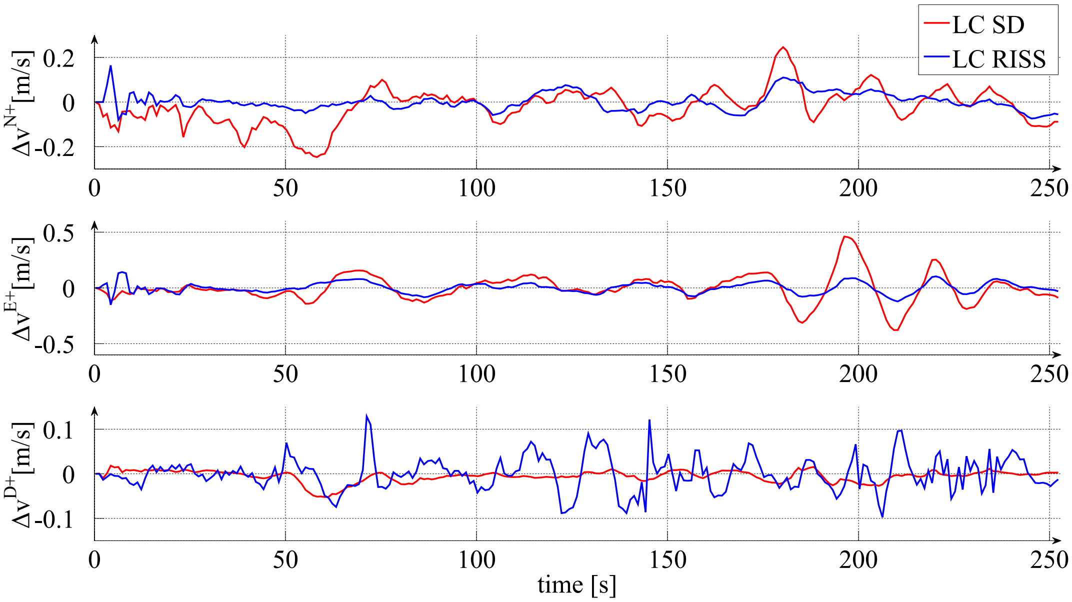

Figure 6.12: Comparison of estimated velocity errors

6.5 GPS outage

An interesting aspect is the behavior of the implemented systems during GPS outages. One main reason to do GPS/INS integration is to overcome GPS outages and to provide a continuous navigation solution. It is to mention that the GPS receiver provides a position at any time of the test measurement. Therefore no GPS outages exist in the measurement data. A GPS outage has to be simulated. A short 8 seconds and a long 30 seconds outage are simulated.

In Figure 6.13 the navigation solutions for the 8 seconds outage are displayed. Both solutions are almost equal and slightly off the road at the end the left turn, but still reasonable for navigation applications.

Figure 6.14 shows the navigation solutions during the long 30 seconds outage. The advantages of the RISS mechanization become clear now. The *LC SD* solution begins to drift after the first turn. On the straight track, at the beginning of the outage, both methods provide good solutions. The *LC RISS* is able to give a reliable navigation solution

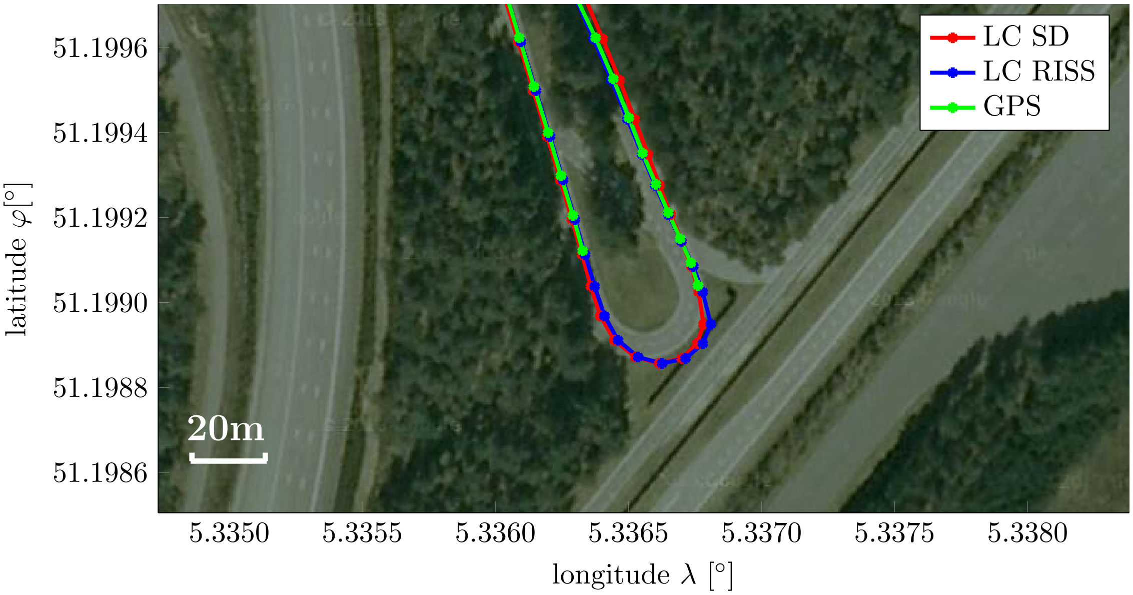

Figure 6.13: *LC SD* and *LC RISS* comparison during short GPS outage

during the whole outage, whereas the *LC SD* solution is only valid for about 16 seconds, before the drift becomes too great.

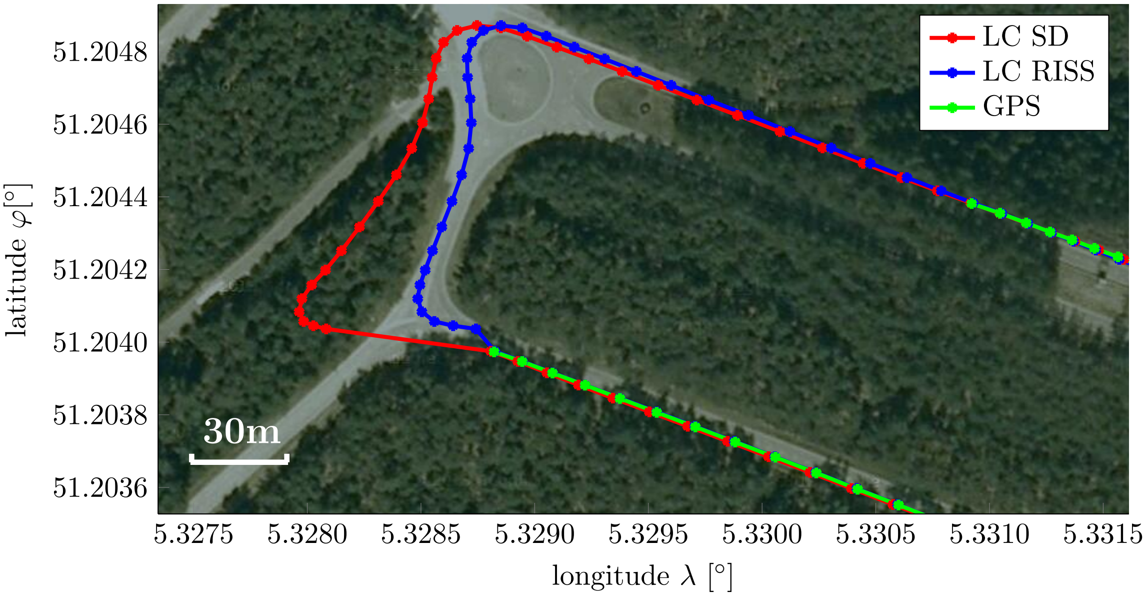

Figure 6.14: *LC SD* and *LC RISS* comparison during long GPS outage

To examine the behavior of both methods in more detail, the uncertainty of the estimates during an absent GPS signal is observed. Figure 6.15 and 6.16 show the error ellipsis plots during the outage. The uncertainty of the *LC SD* integration is growing faster and becomes much larger than the *LC RISS* method's. During the outage the fusion algorithm is always in prediction mode and the error covariance matrix $\boldsymbol{P}_k$ is not updated by a new measurement (see Figure 5.2).

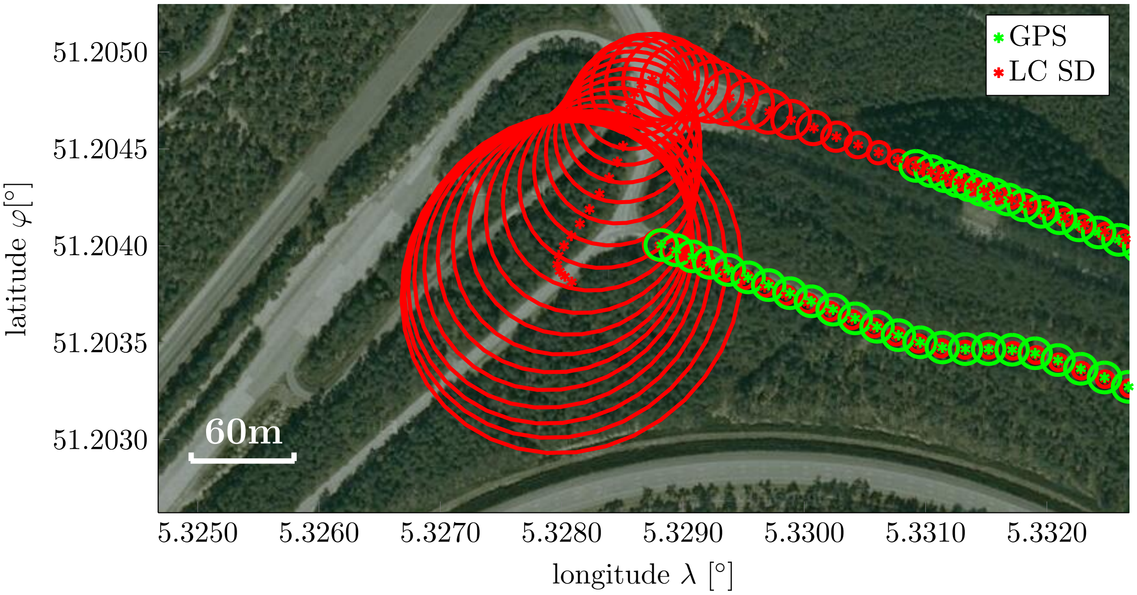

Figure 6.15: *LC SD* error ellipsis with 95% confidence interval during long GPS outage

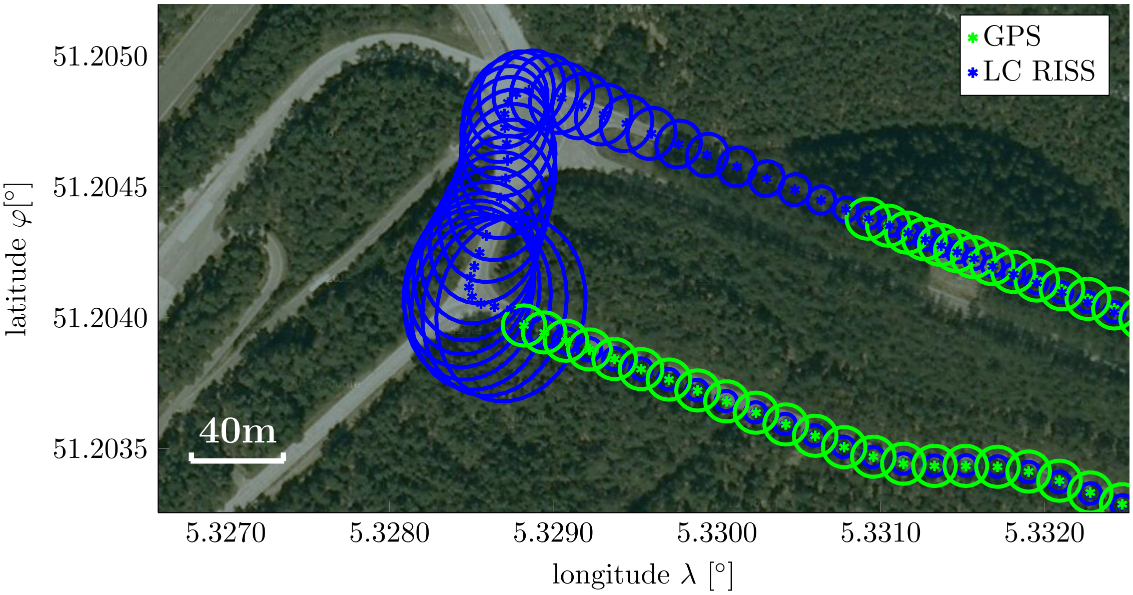

Figure 6.16: *LC RISS* error ellipsis with 95% confidence interval during long GPS outage

One idea to obtain a reliable LC SD solution during long GPS outages is to reduce the drift with velocity updates, provided by the wheel-speed sensors. The navigation solution for the same 30 seconds outage is shown in Figure 6.17. The drift is completely gone. Still the *LC RISS* solution is still more trustful.

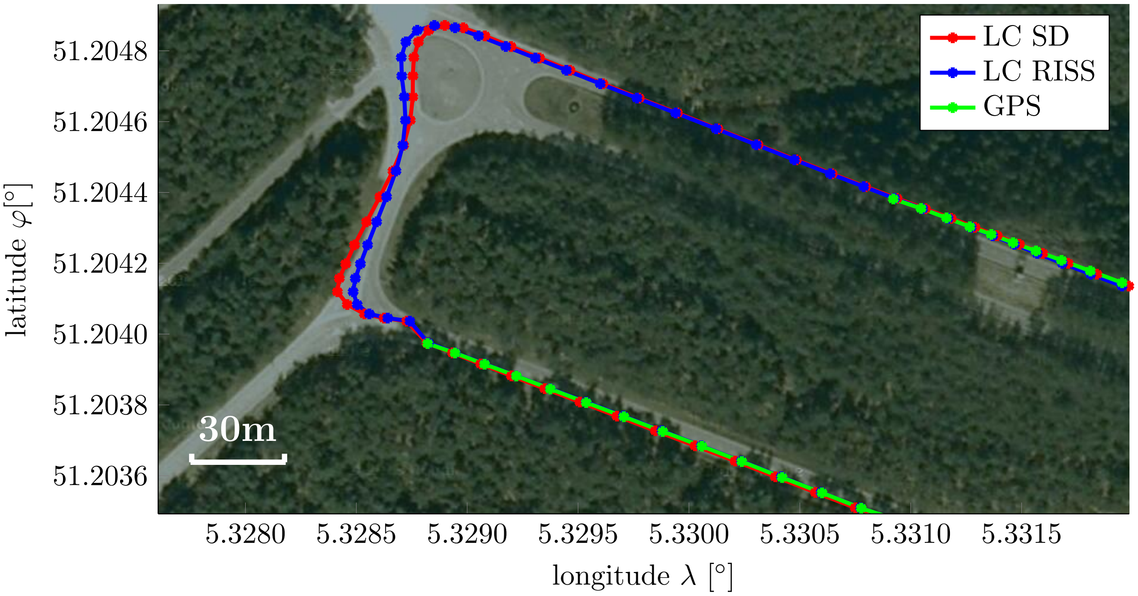

Figure 6.17: *LC SD* and *LC RISS* (with additional odometer velocity update) comparison during long GPS outage

The error ellipsis plot for the *LC S*D velocity update method is displayed in Figure 6.18. Compared to Figure 6.15, the uncertainty is strongly reduced through this method.

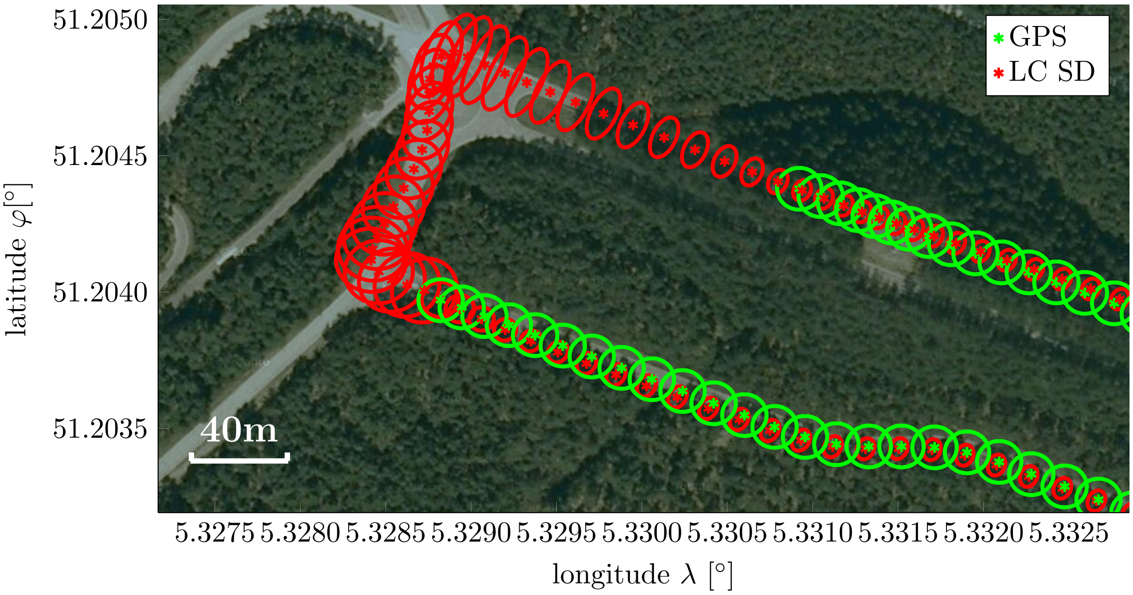

Figure 6.18: *LC SD* error ellipsis with 95% confidence interval during long GPS outage, with additional odometer velocity update

6.6 Discussion

In this Section the observations of the previous Sections in this Chapter are further discussed. Other characteristics of the implemented systems that are not shown in the previous Chapter are discussed as well.

The comparison of the SDINS and the RISS mechanization without any GPS aiding in Section 6.1 shows the fundamental differences of the two approaches. The stated advantages of the RISS model (see Subsection 3.2.3) are confirmed through simulation. The 3D RISS navigation solution drifts much less over time. At the end of a 4 minute long simulation, the drift caused an 2.75 kilometer position error in the plane, compared to the GPS measurement. The RISS error is only about 100 meters. The height error is about 600 meters for the SDINS, but only about 1 meter for the RISS. A nearly drift-less height estimation, like in Figure 6.2 with RISS mechanization, is not possible with all test data sets. A drift over time is present in some cases, but still much smaller compared to the RISS mechanization with external pitch from the AHRS sensor. The superior performance of the RISS mechanization will be an advantage during GPS outages, which will be discussed later. Different higher order integration methods are tested to compute the mechanization process. The simulation shows, that there is nearly no difference between higher order integration methods and a simple Euler method (Equation 3.37). The Euler method is therefore sufficient for a simulation with 50Hz IMU measurement data.

The analysis of the different attitude estimations in Section 6.2 leads to an unexpected result. The heading provided by the AHRS sensor is not reliable. Based on the GPS track, a plausibility check of the heading, respectively the yaw angle, can be performed. The AHRS angle is too large and does not match with the road trajectory. This behavior is also observed in Figure 6.3, where the difference accumulates to approximately 40°, compared to the LC headings. The yaw angles of both implemented methods and the GPS heading are almost equal and provide a good estimation. The same behavior is true for the pitch and roll angles. The AHRS pitch seems to be not correct as well, like Figure 6.6 indicates. It is to mention that an incorrect AHRS attitude estimation is present in all test measurements.

The reason why the AHRS Euler angles are not correct, could be the internal magnetometer fusion algorithm, which should actually provide an accurate attitude estimation. The sensor was mounted inside the fully electrical powered test vehicle, where external magnetic fields may distort the earth's magnetic field. A magnetometer fusion algorithm is not suitable for such an environment. Further investigations are beyond the scope of this thesis.

The performance comparison of both implemented systems with the GPS measurement in terms of accuracy is not feasible, since no highly accurate reference measurement is available. A visual comparison is done by comparing the simulated tracks with a satellite plot overlay. Figure 6.7 and 6.8 shows a selection of different simulated track sections. At some parts of the track the simulation results are more reliable than the GPS measurement and on other parts the GPS position seems more plausible. At straight tracks or turns, it may happen that the LC position estimates are off the road. The *LC RISS* provides in nearly all simulations the better absolute position estimation, due to the advantages of the proposed RISS mechanization. It is to mention that the absolute position measurements from the low-cost GPS receiver, is good from the beginning. The signal was never blocked according to the provided GPS status signal. The track trajectories are always reasonable and only in a few cases slightly beside the road, based on comparison with satellite plots. This is why the standard deviation for the GPS measurement is chosen $\sigma_{\varphi,\lambda} = 3.5\text{m}$ and $\sigma_h = 5.5\text{m}$ for the height measurement. The results from the simulated *LC SD* and *LC RISS* could be further improved by assuming less measurement noise or assuming increased system noise. Too small or too large noise terms are not representing realistic conditions and can not be justified.

The *LC SD* height estimation presented in Figure 6.11 suffers from a drifting behavior, despite GPS aiding. The reason could be an uncompensated bias in the inertial sensors, causing an incorrect transformation into the navigation frame and resulting in a false gravity compensation. The bias-drift is modeled and compensated like discussed in Chapter 4, but a constant bias sensor error can not be identified. A correct bias compensation of the non-statistical part of the bias error, is only possible with a sensor calibration prior the measurement. The drift in the height estimation for the *LC SD* is reduced by subtracting a constant z-acceleration, identified experimentally. The noise variance for the z-accelerometer is also increased, more than for the other sensors (see Table C.1).

It is worth mentioning that velocity aiding, additionally to the GPS position aiding, do not increase the absolute position accuracy. For all simulations presented in Chapter 6, only position aiding is used, except for the long *LC SD* GPS outage with noted velocity update (see Figure 6.17). There are several reasons why a velocity update does not increase the accuracy, in some cases even worsen the result. For the *LC RISS* a GPS velocity update is not recommended, because the RISS mechanization is based on odometer velocity measurements, which are already more accurate than the GPS velocity. For *LC SD* two velocity signal sources, one from the GPS receiver and the other from the wheel-speed sensors, are available. The wheel-speed measurement is favored because of increased accuracy and GPS velocity is not available during GPS outages anyway. Because in both cases only the absolute velocity is measured, rather that the NED velocities, a non-linear transformation has to be performed (see Appendix A). An approximation for the uncertainty propagation and the usage of estimated Euler angles, leads to an incorrect velocity update. However, a velocity update, like it is presented in Section 6.5, is successfully used to overcome long GPS outages.

The increased reliability of GPS/INS integrated systems during GPS outages are presented in Section 6.5. It is shown that for short outages of about 8 seconds, both implemented methods can still provide a reliable navigation solution. For the simulated 30 second outage, the *LC SD* is not able to provide a reasonable absolute position. During the outage, both systems are in prediction mode, which means that the navigation solution corresponds to the SDINS, or the RISS mechanization navigation solution respectively (see Section 6.1). The behavior of both mechanization methods is already discussed. A better *LC RISS* navigation solution, even during long GPS outages is expected, like it is verified in Figure 6.13.

In addition, some effort was spent on obtaining a reliable navigation solution for long GPS outages for the *LC SD* method. This is achieved with odometer velocity updates. Wheel-speed measurements are available, even during GPS outages. They are used to perform velocity updates, instead of GPS position updates. Figure 6.17 proves that due to this approach, it is possible to get a trustworthy solution again. The *LC SD* method, with odometer velocity updates during GPS outages, utilizes information of all available sensors.

Various simulations of GPS outages showed that even the LC RISS method is not always able to provide a good position estimate during GPS outages. For example, if the outage occurs at a time when the yaw angle is badly estimated, the solution becomes less dependable over time as well. It is to mention that a GPS outage effects the height estimation in the same way, as for the track. The height estimation during an outage for *LC RISS* is still reliable, whereas the *LC SD* height tends to drift and for long outages it seems to diverge.

The visualization of the estimated position uncertainty is done by plotting error ellipsis 95% confidence interval. Figure 6.9 and 6.10 indicate that the uncertainty is decreased, compared to the GPS measurements. The decreased estimated ESKF uncertainty for both LC systems is almost equal. Without any GPS outages, the mean standard deviation of the settled ESKF estimate, is for the *LC SD* about $\sigma_{\varphi,\lambda} \approx 1.4$m, whereas for the *LC RISS* it is about $\sigma_\varphi \approx 1.4$m, $\sigma_\lambda \approx 1.44$m.

The uncertainty for GPS outages is presented by Figure 6.15 and 6.16. The comparison shows that the uncertainty of *LC SD* technique grows faster and becomes therefore much larger than for the *LC RISS*. This corresponds to the *LC SD* drifting behavior, which leads to a bad navigation solution. The uncertainty during an outage is reduced by using odometer velocity updates (see Figure 6.18). Then the uncertainty is comparable to the *LC RISS* method.

Chapter 7

Conclusion and Outlook

In this thesis two different GPS/INS integration methods are derived and tested via numerical simulation. The integration architecture is a loosely-coupled system in indirect configuration (see Section 2.2). The fusion method is based on Kalman filtering (Section 5.1). The two implemented methods operate in the discrete time domain, based on the recursive KF scheme. Therefore the algorithms are suitable for real time applications. Due to the indirect configuration the errors of the inertial navigation system are estimated and used for correction to get full estimated states. The KF corresponding to this kind of estimation is called ESKF. With this integration technique a fusion algorithm is designed that considers the multi-rate property of the used sensors (see Section 5.4). With the implemented method it is even possible to consider several sensors with more than two different sampling rates, as long as the sampling rates are integer multiples of each other. The method is therefore not restricted to the 50Hz/1Hz inertial sensors/GPS receiver sampling rate setup of this thesis. The algorithms are suitable for arbitrary sensor configurations which provides high flexibility. The fused data is provided by inertial sensors, wheel-speed sensors and a GPS receiver. The loosely-coupled system based on the full IMU strapdown approach utilizes inertial sensor measurements, GPS position measurements and even odometer velocity measurements to overcome GPS outages without drifting. The loosely-coupled reduced inertial sensor system uses odometer and GPS position measurements.

The basis for the GPS/INS integration are the two mechanization processes. The first one is the strapdown algorithm, using inertial measurements to calculate the 3D attitude, velocity and position. The second one is the reduced inertial sensor system mechanization, based on odometer and a reduced set of inertial sensors to also compute 3D attitude, velocity and position (see Chapter 3). To correctly describe the errors of the INS, an inertial sensor error analysis is performed, with tools as AV plot and autocorrelation function (see Chapter 4). The algorithms are derived by defining the error equations formulating the INS errors. The computation steps of the GPS/INS integration algorithms are then explained in detail in Chapter 5. The simulation results show the different behavior of the *LC SD* and the *LC RISS* system. As a result it is found out that the integration methods provide a better attitude estimation as the AHRS sensor. Without highly accurate reference trajectory, no reliable statement about the absolute position accuracy can be made. However, another advantage of GPS/INS integration is shown:

the reliability of the system is increased during GPS outages. With additional odometer velocity updates, even the *LC SD* system is able to overcome long absent GPS signals.

For the use in real time applications the properties of the CAN-Bus network have to be further investigated. The time delay between the different sensor measurements can cause major problems for GPS/INS integration. Further research is also necessary to identify and correct the bias offset of the inertial sensors. Even with GPS aiding a complete compensation of the drift is not always possible. Based on this thesis advanced integration approaches are conceivable. The algorithms could be used in combination with digital map information, realizing a map aiding system. A different approach is the use of steering angle measurements for heading calculations. The heading can be applied as additional aiding source for the KF or to derive a completely new vehicle dynamics model, based on steering angle and odometer measurements.

Appendix A

Inertial Navigation

This Chapter provides additional relations used in this thesis, which are useful to understand the derivations in Chapter 3 and Chapter 5.

The vector cross product of $\boldsymbol{a} = [a_x, a_y, a_z]^T$ and $\boldsymbol{b} = [b_x, b_y, b_z]^T$ can be written as a matrix/vector multiplication.

$$\boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{A}\boldsymbol{b} \tag{A.1}$$

where $\boldsymbol{A}$ is the skew symmetric matrix

$$\boldsymbol{A} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{A.2}$$

The notation in (A.3) is used to indicate a skew symmetric matrix.

$$\boldsymbol{A} = [\boldsymbol{a}\times] = skew(\boldsymbol{a}) \tag{A.3}$$

Equations (A.4 - A.8) are used when operating with rotation rates. The skew symmetric matrix notation holds for rotation rates as well.

$$\boldsymbol{\omega}_{ie}^{n} = -\boldsymbol{\omega}_{ei}^{n} \tag{A.4}$$

$$\boldsymbol{\omega}_{in}^{n} = \boldsymbol{\omega}_{ie}^{n} + \boldsymbol{\omega}_{en}^{n} \tag{A.5}$$

$$\boldsymbol{\Omega}_{ie}^{n} = [\boldsymbol{\omega}_{ie}^{n}\times] \quad , \quad \boldsymbol{\Omega}_{en}^{n} = [\boldsymbol{\omega}_{en}^{n}\times] \tag{A.6}$$

$$\boldsymbol{\Omega}_{ie}^{n} = -\boldsymbol{\Omega}_{ei}^{n} \tag{A.7}$$

$$\boldsymbol{\Omega}_{in}^{n} = \boldsymbol{\Omega}_{ie}^{n} + \boldsymbol{\Omega}_{en}^{n} \tag{A.8}$$

The DCM is an orthogonal matrix and therefore the following relations hold.

$$\boldsymbol{C}_b^n = \boldsymbol{C}_n^{b,-1} = \boldsymbol{C}_n^{b,T} \tag{A.9}$$

$$\boldsymbol{C}_b^n \boldsymbol{C}_n^b = \boldsymbol{I} \tag{A.10}$$

A quaternion multiplication is done according to Equation (A.11).

$$\boldsymbol{q}_1 \bullet \boldsymbol{q}_2 = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \bullet \begin{bmatrix} e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix} \begin{bmatrix} e \\ f \\ g \\ h \end{bmatrix} \tag{A.11}$$

An alternative formulation to Equation (3.14) is also possible. Here the rotation rates are written in matrix notation instead of quaternions.

$$\dot{\boldsymbol{q}}_b^n = \frac{1}{2}\overline{\boldsymbol{\Omega}}(\boldsymbol{\omega_{nb}^b})\boldsymbol{q}_b^n \tag{A.12}$$

where

$$\overline{\boldsymbol{\Omega}}(\boldsymbol{\omega_{nb}^b}) = \begin{bmatrix} -\boldsymbol{\Omega}_{nb,(3\times3)}^b & \vdots & \boldsymbol{\omega}_{nb,(3\times1)}^b \\ \cdots & \cdots & \cdots \\ -\boldsymbol{\omega}_{nb,(1\times3)}^{b,T} & \vdots & 0 \end{bmatrix} = \begin{bmatrix} 0 & \omega_{nb,z}^b & -\omega_{nb,y}^b & \omega_{nb,x}^b \\ -\omega_{nb,z}^b & 0 & -\omega_{nb,x}^b & \omega_{nb,y}^b \\ \omega_{nb,y}^b & -\omega_{nb,x}^b & 0 & \omega_{nb,z}^b \\ -\omega_{nb,x}^b & -\omega_{nb,y}^b & -\omega_{nb,z}^b & 0 \end{bmatrix} \tag{A.13}$$

The ENU velocities for the RISS mechanization in Equation (3.47) are obtained with the transformation in (A.14). The graphical representation is given in Figure A.1.

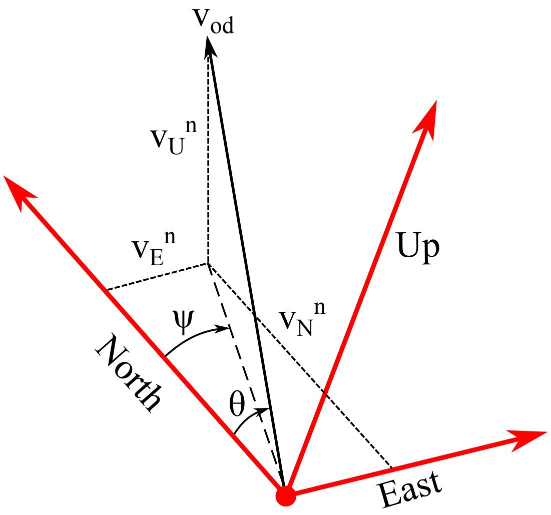

Figure A.1: Absolute velocity measurement in the ENU navigation frame

$$\begin{bmatrix} v_E^n \\ v_N^n \\ v_U^n \end{bmatrix} = v_{od} \begin{bmatrix} cos\theta sin\psi \\ cos\theta cos\psi \\ sin\theta \end{bmatrix} \tag{A.14}$$

Appendix B

GPS/INS Integration

B.1 Time continuous to time discrete

The time continuous system of form (B.1) can be converted into a time discrete system (B.2).

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t) + \boldsymbol{G}(t)\boldsymbol{w}(t) \tag{B.1}$$

$$\boldsymbol{x}_k = \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{B}_{k-1}\boldsymbol{u}_{k-1} + \boldsymbol{G}_{k-1}\boldsymbol{w}_{k-1} \tag{B.2}$$

The system dynamic matrix can be approximated with a abort Taylor series expansion after the linear term (Eq.B.3).

$$\boldsymbol{F}_k = e^{\boldsymbol{F}\Delta t} \approx \boldsymbol{I} + \boldsymbol{F}\Delta t \tag{B.3}$$

For the control vector coupling matrix $\boldsymbol{B}_k$ there are different approximations available (Wendel, 2011). The approximations are the same for the noise distribution matrix $\boldsymbol{G}_k$.

$$\boldsymbol{B}_k \approx \frac{1}{2}(\boldsymbol{I} + \boldsymbol{F}_k)\boldsymbol{B}\Delta t \tag{B.4}$$

$$\boldsymbol{B}_k \approx \boldsymbol{F}_k\boldsymbol{B}\Delta t \tag{B.5}$$

$$\boldsymbol{B}_k \approx \boldsymbol{B}\Delta t \tag{B.6}$$

The system noise covariance matrix in continuous time is defined in Equation (B.7). The discrete system noise with corresponding discrete covariance matrix is calculated by Equation (B.8-B.9).

$$E[\boldsymbol{w}(t)\boldsymbol{w}(\tau)^T] = \boldsymbol{Q}\delta(t-\tau) \tag{B.7}$$

$$\boldsymbol{w}_k = \frac{1}{\Delta t}\int_{t_k}^{t_{k+1}} \boldsymbol{w}(\tau)d\tau \tag{B.8}$$

$$\boldsymbol{Q}_k = E[\boldsymbol{w}_k\boldsymbol{w}_k^T] = \frac{\boldsymbol{Q}}{\Delta t} \tag{B.9}$$

B.2 Loosely-Coupled Strapdown Integration

With Equations (B.10-B.17) all matrix elements of the error dynamic system matrix from Equation (5.51) are defined.

$$\boldsymbol{F}_{11} = \begin{bmatrix} 0 & 0 & \frac{v^n_{eb,N}}{R_n-h} \\ \frac{v^n_{eb,E} tan\varphi}{R_n-h} & 0 & \frac{v^n_{eb,E}}{R_e-h} \\ 0 & 0 & 0 \end{bmatrix} \tag{B.10}$$

$$\boldsymbol{F}_{21} = [\boldsymbol{v}^n_{eb}\times]\left(2\left[\frac{\partial \boldsymbol{\omega}^n_{ie}}{\partial(\varphi,\lambda,h)}\right] + \left[\frac{\partial \boldsymbol{\omega}^n_{en}}{\partial(\varphi,\lambda,h)}\right]\right)\left[\frac{\partial(\varphi,\lambda,h)}{\partial(x_N,x_E,x_D)}\right] \tag{B.11}$$

$$\boldsymbol{F}_{22} = -\left(2\boldsymbol{\Omega}^n_{ie} + \boldsymbol{\Omega}^n_{en}\right) + [\boldsymbol{v}^n_{eb}\times]\left[\frac{\partial \boldsymbol{\omega}^n_{en}}{\partial \boldsymbol{v}^n_{eb}}\right] \tag{B.12}$$

$$\boldsymbol{F}_{23} = -\left[\boldsymbol{C}^{\widehat{n}}_b \boldsymbol{f}^b_{ib}\times\right] \tag{B.13}$$

$$\boldsymbol{F}_{31} = -\left(\left[\frac{\partial \boldsymbol{\omega}^n_{ie}}{\partial(\varphi,\lambda,h)}\right] + \left[\frac{\partial \boldsymbol{\omega}^n_{en}}{\partial(\varphi,\lambda,h)}\right]\right)\left[\frac{\partial(\varphi,\lambda,h)}{\partial(x_N,x_E,x_D)}\right] \tag{B.14}$$

$$\boldsymbol{F}_{32} = -\left[\frac{\partial \boldsymbol{\omega}^n_{en}}{\partial \boldsymbol{v}^n_{eb}}\right] \tag{B.15}$$

$$\boldsymbol{F}_{33} = -\boldsymbol{\Omega}^n_{in} = -[\boldsymbol{\omega}^n_{in}\times] = -[(\boldsymbol{\omega}^n_{ie} + \boldsymbol{\omega}^n_{en})\times] \tag{B.16}$$

$$\boldsymbol{F}_{44} = \begin{bmatrix} -\beta_{ax} & 0 & 0 \\ 0 & -\beta_{ay} & 0 \\ 0 & 0 & -\beta_{az} \end{bmatrix} \tag{B.17}$$

The missing partial vector derivative matrices to finally compute the error dynamic matrix $\boldsymbol{F}$ are given by Equations (B.18-B.21).

$$\frac{\partial(\varphi,\lambda,h)}{\partial(x_N,x_E,x_D)} = \begin{bmatrix} \frac{1}{R_n-h} & 0 & 0 \\ 0 & \frac{1}{(R_e-h)cos\varphi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.18}$$

$$\frac{\partial \boldsymbol{\omega}^n_{ie}}{\partial(\varphi,\lambda,h)} = \begin{bmatrix} -\omega_E sin\varphi & 0 & 0 \\ 0 & 0 & 0 \\ -\omega_E cos\varphi & 0 & 0 \end{bmatrix} \tag{B.19}$$

$$\frac{\partial \boldsymbol{\omega}^n_{en}}{\partial(\varphi,\lambda,h)} \approx \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{v^n_{eb,E}}{(R_e-h)cos^2\varphi} & 0 & 0 \end{bmatrix} \tag{B.20}$$

$$\frac{\partial \boldsymbol{\omega}^n_{en}}{\partial \boldsymbol{v}^n_{eb}} = \begin{bmatrix} 0 & \frac{1}{R_e-h} & 0 \\ -\frac{1}{R_n-h} & 0 & 0 \\ 0 & -\frac{tan\varphi}{R_e-h} & 0 \end{bmatrix} \tag{B.21}$$

Appendix C

Used Parameters

Table C.1 lists the actually used error coefficients for the simulation. The parameters are increased by the factor 2.5, compared to the identified values, because under real conditions the noise can be assumed larger. The parameters for z-axis accelerometer are increased up to a factor of 7.5 (for N_{az}), to reduce the drift in height simulation (see Subsection 6.3.2). The parameters are found by trial and error.

	velocity random walk	bias instability	acceleration random walk
	N_a $[m/s/\sqrt{s}]$	B_a $[m/s^2]$	K_a $[m/s/s^{3/2}]$
accel x	2.625×10^{-3}	1×10^{-3}	1.24×10^{-4}
accel y	2.45×10^{-3}	6.9×10^{-4}	8.4×10^{-5}
accel z	7.875×10^{-3}	2.1×10^{-3}	3.36×10^{-4}
	angle random walk	bias instability	rate random walk
	N_g $[rad/\sqrt{s}]$	B_g $[rad/s]$	K_g $[rad/s^{3/2}]$
gyro x	1.425×10^{-3}	-	-
gyro y	1.275×10^{-3}	-	-
gyro z	1.2425×10^{-3}	-	-

Table C.1: Error coefficients used for simulation

Appendix D

List of abbreviations

ADAS	Advanced Diver Assistance System
AHRS	Attitude Heading Reference System
AI	Artificial Intelligence
AV	Allan Variance
CAN	Controller Area Network
DCM	Direction Cosine Matrix
DGPS	Differential Global Positioning System
DOF	Degrees of Freedom
DR	Dead Reckoning
EKF	Extended Kalman Filter
ENU	East-North-Up
ESKF	Error State Kalman Filter
GM	Gauss-Markov
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISA	Inertial Sensor Assembly
KF	Kalman Filter
LC	Loosely Coupled
LDW	Lane Departure Warning
LIDAR	Light Detection and Ranging
LLH	Latitude, Longitude, Height
MEMS	Micro Electro Mechanical System
NED	North-East-Down
RISS	Reduced Inertial Sensor System
RSS	Received Signal Strength
RW	Random Walk
SBAS	Satellite Based Augmentation System
SD	Strap-Down
SDINS	Strap-Down Inertial Sensor System
TC	Tightly Coupled
WADGPS	Wide Area Differential Global Positioning System
WGS84	World Geodetic System 1984
WN	White Noise

Bibliography

Ahmad, U. et al. (2006). “In-building Localization using Neural Networks”. In: *IEEE International Conference on Engineering of Intelligent Systems*. Islamabad, pp. 1–6. DOI: 10.1109/ICEIS.2006.1703135.

Allen, W.J (2011). “Use of vision sensors and lane maps to aid gps-ins navigation”. Auburn University. Master Thesis. Auburn University. URL: http://etd.auburn.edu/bitstream/handle/10415/2807/thesis_john_allen.pdf?sequence=2 (visited on 06/07/2015).

Asadian, A., B. Moshiri, and A.K. Sedigh (2005). “A novel data fusion approach in an integrated gps/ins system using adaptive fuzzy particle filter”. In: *5th International Conference on Technology and Automation (ICTA)*, pp. 15–16.

Atlatec (2015). *Vision Based Mapping and Localization*. Web page. URL: http://atlatec.de/ (visited on 07/07/2015).

Avinash, P. et al. (2010). “Coarse In-Building Localization with Smartphones”. In: *Mobile Computing, Applications and Services* 35. Springer, pp. 343–354. DOI: 10.1007/978-3-642-12607-9_25.

Bauer, Clemens (2014). “Indoor Pedestrian Routing on Mobile Devices”. Graz University of Technology. Master Thesis. Graz University of Technology.

Boutayeb, M., H. Rafaralahy, and M. Darouach (1997). “Convergence analysis of the extended Kalman filter used as an observer for nonlinear deterministic discrete-time systems”. In: *IEEE Transactions on Automatic Control* 42.4, pp. 581–586. DOI: 10.1109/9.566674.

Czerwinski, Fabian (2010). *allan v3.0*. MATLAB Cenral. URL: http://www.mathworks.com/matlabcentral/fileexchange/26659-allan-v3-0 (visited on 03/08/2015).

Dutzler, Roland (2013). “Smartphone based Indoor-Navigation”. Graz University of Technology. Master Thesis. Graz University of Technology.

Estrada, C., J. Neira, and J.D Tardos (2005). “Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments”. In: *IEEE Transactions on Robotics* 21.4, pp. 588–596. DOI: 10.1109/TITS.2009.2026317.

Freescale, Semiconductor (2015). *Allan Variance: Noise Analysis for Gyroscopes*. Web Page. Application Note. URL: http://cache.freescale.com/files/sensors/doc/app_note/AN5087.pdf (visited on 02/08/2015).

Georgy, J. et al. (2010). “Low-Cost Three-Dimensinal Navigation Solution for RISS/GPS Integration Using Mixture Particle Filter”. In: *IEEE Transactions on Vehicular Technology* 59.2, pp. 599–615. DOI: 10.1109/TVT.2009.2034267.

Goswami, Subrata (2012). *Indoor Location Technologies*. Springer.

Grewal, Mohinder S. and Angus P. Andrews (2008). *KALMAN FILTERING: Theory and Practice Using MATLAB*. 3rd ed. ISBN 978-0-470-17366-4. John Wiley and Sons, Inc.

Gustafsson, F. (2012). *Statistical Sensor Fusion*. 2nd ed. Lund. Studentlitteratur.

Hoffman-Wellenhof, B., K. Legat, and M. Wieser (2003). *navigation: principles of positioning and guidance*. Springer.

Hoffman-Wellenhof, B. and H. Moritz (2006). *Physical Geodesy*. 2nd ed. Springer Vienna. DOI: 10.1007/978-3-211-33545-1.

Hofmann-Wellenhof, B., H. Lichtenegger, and E. Wasle (2007). *GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more*. Springer Vienna. ISBN: 9783211730171.

IEEE (1997). "IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros". In: *IEEE Std 952-1997 (R2008)*. DOI: 10.1109/IEEESTD.1998.86153.

ISO/IEC (2008). *Uncertainty of measurement – Part 3: Guide to the expression of uncertainty in measurement (GUM:1995)*. ISO/IEC Guide 98-3:2008.

Jabbour, M., P. Bonnifait, and V. Cherfaoui (2008). "Road tracking for multi-hypothesis localization on navigable maps". In: *IEEE Intelligent Vehicles Symposium*, pp. 138–143. DOI: 10.1109/IVS.2008.4621286.

Jang, J.S.R., C.T. Sun, and E. Mizutani (1997). *Neuro-fuzzy and soft computing; a computational approach to learning and machine intelligence*. Prentice Hall, Upper Saddle River.

Kandel, A. (1991). *Fuzzy Expert Systems*. Taylor & Francis. ISBN: 9780849342974.

Kaplan, E. and C. Hegarty (2005). *Understanding GPS: Principles and Applications*. 2nd ed. Artech House mobile communications series. Artech House. ISBN: 9781580538954. URL: https://books.google.at/books?id=-sPXPu0W7ggC.

Kim, Phil (2011). *Kalman Filter for Beginners: with MATLAB Examples*. ISBN: 978-1463648350. CreateSpace Independent Publishing Platform.

Krzikalla, R. et al. (2013). "Mehr Sicherheit durch Positionsbestimmung mit Satelliten und Landmarken". In: *Vernetztes Automobil, ATZ/MTZ Fachbuch, Springer*, pp. 20–26. DOI: 10.1007/978-3-658-04019-2_4.

Kuipers, J.B. (1999). *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. New Jersey. Princeton University Press.

Lachapelle, G. (2004). "GNSS Indoor Location Technologies". In: *Journal of Global Positioning Systems* 3.1-2, pp. 3–11.

Magnusson, N. and T. Odenman (2012). "Improving absolute position estimates of an automotive vehicle using GPS in sensor fusion". Chalmers University of Technology, Goeteborg, Schweden. Master Thesis. Chalmers University of Technology.

Mansfeld, W. (2013). *Satellitenortung und Navigation*. Vol. 2. ISBN: 9783663113287. Vieweg + Teubner Verlag.

Maurer, M. et al. (2015). *Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte*. Springer. DOI: 10.1007/978-3-662-45854-9.

Munguia, R. (2014). "A GPS-aided Inertial Navigation System in Direct Configuration". In: *Journal of Applied Research and Technology* 12, pp. 803–814.

NAVIlock (2015). *Navilock NL-302U USB GPS reciever*. Web page. URL: http://www.navilock.de/produkte/G_61422/merkmale.html (visited on 05/07/2015).

NCEI (2015). *Magnetic Declination Estimator*. Web Page. National Centers for Environmental Information. URL: http://www.ngdc.noaa.gov/geomag-web/#declination (visited on 08/18/2015).

Noureldin, A., U. Iqbal, and A.F. Okou (2008). "An Integrated Reduced Inertial Sensor System - RISS/GPS for Land Vehicle". In: *Position, Location and Navigation Symposium, 2008 IEEE/ION*. Monterey, CA, pp. 1014–1021. DOI: 10.1109/PLANS.2008.4570075.

Noureldin, A., T.B. Karamat, and J. Georgy (2012). *Fundamentals of INS, GPS and Their Integration*. Springer. ISBN: 9783642304651.

Prasad, R. and M. Ruggieri (2005). *Applied satellite navigation - using GPS, GALILEO and augmentation systems*. ISBN: 1-58053-814-2. Artech House.

Quinchia, A.G. et al. (2013). "A Comparison between Different Error Modeling of MEMS Applied to GPS/INS Integrated Systems". In: *Sensors* 13.8. ISSN 1424-8220, pp. 9549–9588. DOI: 10.3390/s130809549.

Sasiadek, J.Z. (2002). "Sensor Fusion". In: *Annual Reviews in Control* 26.2, pp. 203–228. DOI: http://dx.doi.org/10.1016/S1367-5788(02)00045-7.

SBG systems (2011). *IG-500A Sub-miniature AHRS User Manual*. URL: http://www.sbg-systems.com/products/ig500a-miniature-ahrs (visited on 05/07/2015).

— (2015). *Apogee-N: High Accuracy INS/GNSS*. Web page. URL: http://www.sbg-systems.com/apogee-sensors/apogee-n-high-accuracy-inertial-navigation-system (visited on 06/07/2015).

Schleicher, D. et al. (2009). "Real-Time Hierarchical Outdoor SLAM Based on Stereovision and GPS Fusion". In: *IEEE Transactions on Intelligent Transportation Systems* 10.3, pp. 440–452. DOI: 10.1109/TITS.2009.2026317.

Schmackers, J. and A. Glasmachers (2011). "Landmark based fast positioning for sensor data fusion: receiver design and measurement results". In: *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 25–30. DOI: 10.1109/ITSC.2011.6082808.

El-Sheimy, N., K. Chiang, and A. Noureldin (2006). "The Utilization of Artificial Neural Networks for Multisensor System Integration in Navigation and Positioning Instruments". In: *IEEE Transactions on Instrumentation and Measurement* 55.5. DOI: 10.1109/TIM.2006.881033.

El-Sheimy, N., H. Hou, and X. Niu (2008). "Analysis and Modeling of Inertial Sensors Using Allan Variance". In: *IEEE Transactions on Instrumentation and Measurement* 57, pp. 140–149. DOI: 10.1109/TIM.2007.908635.

Siebenpfeiffer, W., ed. (2014). *Vernetztes Automobil: Sicherheit - Car IT - Konzepte*. ATZ/MTZ Fachbuch. Springer. DOI: 10.1007/978-3-658-04019-2.

Skog, I. and P. Handel (2009). "In-Car Positioning and Navigation Technologies - A Survey". In: *Intelligent Transportation Systems, IEEE Transactions on* 10.1, pp. 4–21. ISSN: 1524-9050. DOI: 10.1109/TITS.2008.2011712.

Slavov, T. and P. Petkov (2010). "Stochastic Modeling of MEMS Inertial Sensors". In: *Cybernetic and Information Technologies* 10.2. Sofia, pp. 31–40.

Sola, Joan (2015). *Quaternion kinematics for the error-state KF*. Tech.Rep., LAAS-CNRS. URL: https://hal.archives-ouvertes.fr/hal-01122406/document (visited on 03/07/2015).

Titterton, D. and J.L. Weston (2004). *Strapdown Inertial Navigation Technology*. 2nd ed. Electromagnetics and Radar Series. Institution of Engineering and Technology. ISBN: 9780863413582.

Tuna, G. et al. (2012). "Evaluations of different Simultaneous Localization and Mapping (SLAM) algorithms". In: *38th Annual Conference on IEEE Industrial Electronics Society (IECON)*. Montreal, QC, pp. 2693–2698. DOI: 10.1109/IECON.2012.6389151.

Wang, J. and Y. Gao (2007). "The aiding of MEMS INS/GPS integration using artficial intelligence for land vehicle navigation". In: *IAENG International Journal of Computer Science* 33.1, pp. 61–67.

Wendel, J. (2011). *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation*. 2nd ed. Oldenbourg Verlag Muenchen.

Wendel, J. and G. Trommer (2001). "Direct Kalman Filtering of GPS/INS for Aerospace Applications". In: *International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation (KIS)*, pp. 144–149.

Werner, Martin (2014). *Indoor Location-Based Services*. Springer. DOI: 10.1007/978-3-319-10699-1.