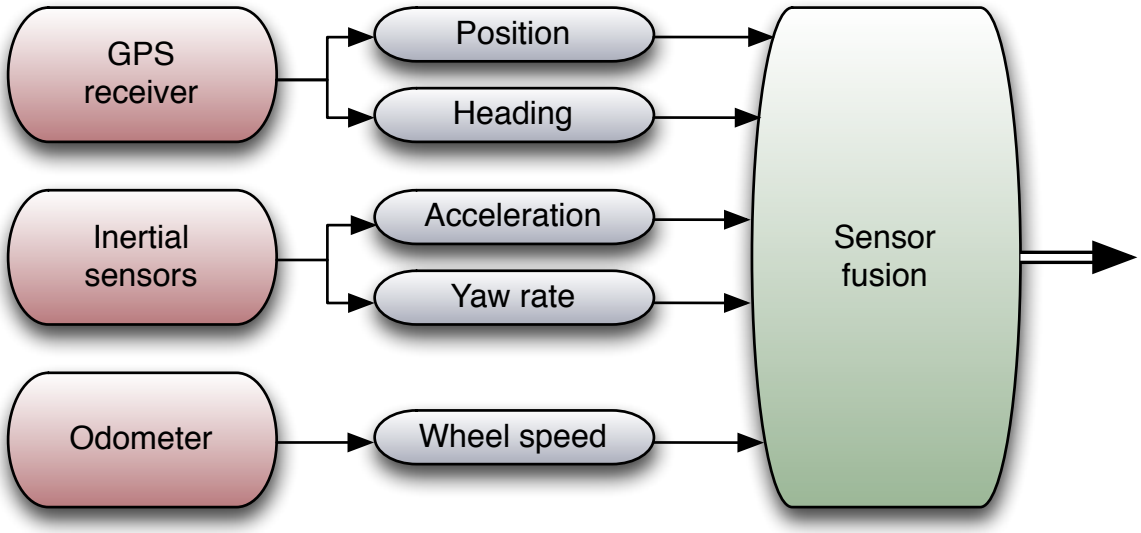
CHALMERS



Improving absolute position estimates of an automotive vehicle using GPS in sensor fusion

Master's Thesis in Signal Processing

NIKLAS MAGNUSSON
TOBIAS ODENMAN

Department of Signals and Systems
Division of Signal Processing and Biomedical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2012
Master's Thesis 2012:EX048

MASTER'S THESIS 2012:EX048

Improving absolute position estimates of an automotive vehicle using GPS in sensor fusion

Master's Thesis in Signal Processing

NIKLAS MAGNUSSON
TOBIAS ODENMAN

Department of Signals and Systems
Division of Signal Processing and Biomedical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2012

Improving absolute position estimates of an automotive vehicle using GPS in sensor fusion
NIKLAS MAGNUSSON
TOBIAS ODENMAN

©NIKLAS MAGNUSSON, TOBIAS ODENMAN, 2012

Master's Thesis 2012:EX048
Department of Signals and Systems
Division of Signal Processing and Biomedical Engineering
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
Sensor fusion using GPS together with IMU and odometer sensor input.

Chalmers Reproservice
Göteborg, Sweden 2012

Improving absolute position estimates of an automotive vehicle using GPS in sensor fusion
Master's Thesis in Signal Processing
NIKLAS MAGNUSSON
TOBIAS ODENMAN
Department of Signals and Systems
Division of Signal Processing and Biomedical Engineering
Chalmers University of Technology

Abstract

The objective with sensor fusion is to combine measurements from different sensors to improve the quality of information such that the combined information becomes more valuable than the information from each individual sensor.

According to our study of previous research, and the sensors available in the test vehicle provided by the Volvo Car Corporation, we created a sensor fusion model based on the extended Kalman filter and measurement inputs from a low-grade Global Positioning System (GPS) receiver, inertial measurement unit (IMU) sensors and odometer, to improve absolute position estimation. The following sensors were available in the test vehicle: radar, camera, accelerometers, gyroscope, odometers and both a low and a high-grade GPS receiver. Volvo Car Corporation does not integrate GPS receivers to improve vehicle functionality, but instead using them as an external device to solely assist the driver in navigation. We acquired sensor data from the test vehicle by driving in different environments such as city, highway, forest roads and on the test track at Hällered. This data was used as input to our non real-time sensor fusion model that was evaluated by simulations in MATLAB. Our results were compared to the output of a highly accurate system that utilizes a real-time kinematic (RTK)-GPS receiver and a six degree-of-freedom IMU, and to the position measurements of the low-grade GPS receiver.

Simulation results indicated improved accuracy and robustness in some cases, in particular when GPS measurements were less reliable such as in urban environments. On the contrary the Kalman filter output was less accurate when GPS measurements were reliable. These ambiguous results are due to several issues, including the lack of accelerometer data in three dimensions and inaccurate time stamps provided by the low-grade GPS receiver. Determining GPS measurement quality was proven to be difficult. Latency between time of measure of the internal vehicle sensors, and transmission of their output on Controller Area Network (CAN), was also considered as an important issue. Because our model approximates a two-dimensional space, changes in altitude will cause bias errors in the Kalman filter output.

Many of the issues regarding absolute position estimation of an automotive vehicle is covered within this thesis, which could serve as a basis for further development such as real-time implementation in a vehicle.

Keywords: Sensor fusion, Absolute positioning, Kalman filter, GPS, IMU, Dead reckoning, Loose coupling, Kinematic model

Contents

Abstract I

Contents III

Preface V

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Purpose . . . 2
- 1.3 Project delimitations . . . 2
- 1.4 Report layout . . . 2

2 Navigation systems 4
- 2.1 Global navigation satellite system . . . 4
- 2.2 Inertial navigation system . . . 4

3 Sensor theory 5
- 3.1 Global Positioning System . . . 5
  - 3.1.1 Limitations of the GPS . . . 6
  - 3.1.2 Differential and Real Time Kinematic GPS . . . 9
- 3.2 Inertial measurement unit . . . 9
- 3.3 Odometer . . . 10
- 3.4 Camera . . . 10
- 3.5 Radar . . . 12

4 Sensor fusion theory 13
- 4.1 Kalman filter . . . 14
  - 4.1.1 Implementing a discrete Kalman filter . . . 16
  - 4.1.2 Extended Kalman filter . . . 17

5 Existing research 19
- 5.1 Summary of reviewed papers . . . 19
- 5.2 Methods to improve absolute position estimates . . . 21
  - 5.2.1 GPS/INS integration . . . 21
  - 5.2.2 Lane tracking . . . 22
  - 5.2.3 Map matching . . . 22
  - 5.2.4 Traffic sign localization . . . 22
  - 5.2.5 Learning algorithms . . . 23

6 Data acquisition 23
- 6.1 Test and development data . . . 24
- 6.2 Our data collection . . . 25
  - 6.2.1 Data collection scenarios . . . 27
  - 6.2.2 Sensor characteristics analysis . . . 27

7 Proposed sensor fusion 29
- 7.1 Kalman filter model . . . 29
  - 7.1.1 Process model . . . 30
  - 7.1.2 Measurement model . . . 31
  - 7.1.3 Filter parameters . . . 32

7.1.4 On-road detection . . . 34
7.1.5 Gyroscope bias compensation . . . 34
7.2 Data pre-processing . . . 34
7.2.1 Heading angle manipulation . . . 35
7.2.2 Coordinate transformations . . . 36
7.2.3 Data synchronization . . . 39
7.3 Simulation . . . 41
7.3.1 Error ellipses . . . 41

8 Simulation results **42**
8.1 City . . . 42
8.2 Forest road . . . 48
8.3 Highway . . . 49
8.4 Test track . . . 51
8.5 Sample time and synchronization of signals . . . 52
8.6 Map system differences . . . 53
8.7 Determining GPS measurement quality . . . 53

9 Discussion **55**
9.1 Sensors and signals . . . 55
9.1.1 GPS receiver . . . 55
9.1.2 Inertial measurement unit and wheel speed . . . 57
9.2 Data synchronization . . . 57
9.3 Sensor fusion model . . . 57
9.4 Position reference issue . . . 59
9.5 Real-time implementation challenges . . . 59

10 Conclusion **60**

A Matlab code: Error ellipses **67**

B Matlab code: Heading angle adjustment **68**

C Linearized Kalman Filter equations **69**
C.1 Linearized Process and Measurement model of the Kalman Filter . . . 69
C.2 Linearized augmented matrices for extra bias state . . . 70

D IMU noise and bias detection and parametrization **72**

Preface

In this thesis methods to improve absolute position estimate robustness and accuracy of a vehicle using GPS and sensor fusion were studied. A non real-time sensor fusion model was developed and simulated using MATLAB.

This master thesis project was carried out from January 2012 to May 2012 at the Department of Signals and Systems, Division of Signal Processing and Biomedical Engineering and Division of Active Safety, Chalmers University of Technology, Sweden, with Dr. Lennart Svensson as examiner and Henrik Lind as academic supervisor.

Aknowledgements

We would like to thank the department of active safety at Volvo Car Corporation for their support, and Dr. Lars Hammarstrand (Danielsson) at the department of Signals and Systems, Chalmers University of Technology, for helping us with the sensor fusion model. Without their help this project would not have been possible.

Göteborg May 2012
Niklas Magnusson, Tobias Odenman

List of abbreviations

ABS	-	Anti-lock braking system
ARW	-	Angle random walk
BCM	-	Brake control module
CAN	-	Controller Area Network
DEM	-	Digital Elevation Maps
DGPS	-	Differential global positioning system
DOP	-	Dilution of precision
ECEF	-	Earth-centered earth-fixed
EKF	-	Extended Kalman Filter
ENU	-	East, north and up
FIR	-	Far infra-red
GNSS	-	Global navigation satellite systems
GPS	-	Global positioning system
HDOP	-	Horizontal dilution of precision
HS-CAN	-	High-Speed Controller Area Network
IMU	-	Inertial measurement units
INS	-	Inertial navigation systems
LDW	-	Lane departure warning
LIDAR	-	Light Detection And Ranging
LLA	-	Longitude, latitude and altitude
LMSE	-	Least mean squared error
MEMS	-	Microelectromechanical systems
MCS	-	Master control station
MOEMS	-	Micro-opto-electro-mechanical system
NIR	-	Near infra-red
NMEA	-	National Marine Electronics Association
RHCP	-	Right-Handed Circularly Polarized
ROI	-	Region of interest
RTK-GPS	-	Real time kinematic global positioning system
SLAM	-	Simultaneous Localization And Mapping
SNR	-	Signal-to-Noise Ratio
TOF	-	Time-Of-Flight
TSR	-	Traffic sign recognition
UTC	-	Coordinated Universal Time
VDOP	-	Vertical dilution of precision
VRW	-	Velocity random walk

1 Introduction

Vehicle and road safety has been an important field of research during the last decades. Passive safety such as car body and seat belts have been undergoing a lot of development, resulting in mitigated traffic accidents. With increased use of electrical components in the automotive industry such as sensors to perceive the surroundings and the vehicle itself, focus is now also put on active safety.

The field of active safety includes well-known functions such as the anti-lock braking system (ABS), but also newly introduced functions such as collision warning and avoidance. Many active safety functions use relative position information, however by including absolute position data, such as the global position coordinates provided by a GPS receiver, into those functions some of their limitations could be repressed. Acquiring information about the absolute position of a vehicle can be of use in other fields than safety, e.g. it could enable a more efficient transportation system as a whole by autonomously coordinating or monitoring a traffic flow.

To improve position estimations, and at the same time maintain a low cost, it is important to combine information from different low-cost sensors using sensor fusion technology instead of using highly accurate but expensive GPS receivers, which would still suffer from limitations such as the inability of acquiring measurements in e.g. a tunnel. Sensor fusion is a field of technology that is undergoing expansion within the automotive industry, and with increased computational power and advancements in research, new possibilities arises.

1.1 Background

In the modern automotive industry there are several types of sensors that are considered as standard within present and future active safety research, e.g., radar, cameras, inertial measurement units (IMU), odometers and Global Positioning System (GPS) receivers. However Volvo Car Corporation do not integrate GPS receivers to improve vehicle functionality, but instead using them as an external device to solely assist the driver in navigation. By combining data provided from different sensors in a sensor fusion model, a refined output can be made that could aid in mitigating limitations that exists in todays vehicle functions. E.g. it could be easier to interpret important information shared between cars when their absolute positions are known, especially when sender and receiver are not in sight of each other, thus not able to directly measure their relative positions.

The general aim of research within the automotive industry is to provide sufficiently good performance at low cost. GPS receivers commonly used in commercial vehicles usually have an absolute position accuracy of 5 meters (1σ), provided optimal conditions, and an update rate of 1 Hz. Such accuracy and update rate is tolerable in many applications, however in active safety implementations, where peoples lives could be at stake, a more robust and accurate method to provide absolute position estimates is required. A low-grade GPS receiver, fused with IMU and odometer measurements is a common solution for improving absolute position estimations in automotive vehicles, which is because the sensors required usually already exist in modern vehicles. Such a solution could therefore function as a basis from which further methods to improve absolute position estimates can be derived.

Some research in the absolute position estimation field make use of camera and radar to improve estimates even further, where features in the surroundings can be matched with a database such as a map. Multi-variable sensor fusion can be complex to achieve, signals used for fusion often have different update rate and can be on various formats, and when using Controller Area Network (CAN) signals for sensor fusion there is a problem with

varying update rate. In order to advance in the field of sensor fusion to improve absolute position estimation, it is important to provide, not only information about the aspects that contribute to a working implementation, but also to address related problems and challenges.

1.2 Purpose

One of the purposes of this thesis is to study different methods of using sensor fusion within absolute position estimation, and to investigate different problems that may arise and/or solutions that may be required, by performing a literature review.

From this study we will create a sensor fusion model from scratch that is simple and possible to develop further by adding extra states, measurements or functions.

The aim of this model is to improve absolute position estimation, with focus on robustness, of an automotive vehicle by using sensors obtainable in a present Volvo test vehicle.

The idea is to create a sensor fusion model that is not working in real-time, but that instead can serve as a basis of test and evaluation before implementing a real-time system in a vehicle

1.3 Project delimitations

A focus throughout the entire thesis is absolute position estimation.

Articles concerning absolute position estimation of an automotive vehicle that incorporates more or less the sensors obtainable in a Volvo test vehicle, will be addressed in the literature review.

Our sensor fusion model will be built from scratch, simple, and be possible to expand by adding states and/or functions.

A non-real-time system will be created, and because Volvo Car Corporation uses MATLAB the sensor fusion model will be made in that programming language.

Our model is to be validated given data from different environments with a sufficiently wide enough spectrum of dynamics.

Data will be collected by using a Volvo test vehicle.

1.4 Report layout

Sections 2 and 3 in this thesis provide theory about navigation systems, and sensors that are implemented in modern Volvo cars, respectively. The addressed sensors are GPS, IMU, odometer, camera and radar. The thesis continues by describing theory about sensor fusion (section 4), which is the technology of combining different sensors to create a refined output.

A summary of different articles and theory concerning methods for improving position estimation is described in section 5. The general conclusions made in this study of previous research served as a guideline for the development of our proposed sensor fusion model described in section 7.

Sections 6 to 8 are a part of the iterative process of creating a simple, yet robust sensor fusion model using GPS, INS and odometer integration in an extended Kalman filter. This involves acquisition of data (section 6), definition of our sensor fusion model (section 7), pre-processing of data such as coordinate system transformations and data synchronization (section 7.2), and also a selection of simulation results presented in section 8. In section 9

a discussion regarding simulation results as well as other findings and issues is made, and conclusions of this thesis can be found in section 10.

2 Navigation systems

The Merriam-Webster dictionary explains the word *navigation* as:

"the science of getting ships, aircraft, or spacecraft from place to place; especially: the method of determining position, course, and distance traveled." [3]

When people think of the word navigation, items such as a compass and map often come to mind, and perhaps also navigation by stars, such as the northern star Polaris. Through advancement in technology navigation has become more automated in vehicles, it is even possible with many cellular phones that has a built-in global navigation satellite systems (GNSS) receiver. Two types of navigation that are common in the automotive industry will be addressed in this thesis, and these are GNSS and inertial navigation systems (INS). The former rely on outer sources, i.e. satellites, to enable navigation, while the latter only requires the ability to measure inertial forces.

2.1 Global navigation satellite system

GNSS is a navigation method based on signals transmitted from satellites that are in orbit around the earth. The signals are received by GNSS receivers and decoded into, e.g., position, time, velocity and heading.

The most common GNSS is the GPS, which is offered globally for free by the United States Military [63], but other nations have their own financed GNSS. Russia (former Soviet Union) has GLONASS, China is developing their COMPASS, and the European Union is working on a project called Galileo. All of these systems are intended to provide world-wide global coverage [73]. These different systems all uses internal reference time that is fundamental to enable absolute position measurements through satellite signals. GNSS receivers utilizes different time standards, such as UTC and GPS-time, where receivers using UTC must be synchronized to account for leap seconds [46]. Time synchronization is explained further in section 7.2.3.

Today the difference between these different GNSS systems is the number of satellites in orbit. GPS has 24 active satellites, which is enough to have global coverage, while GLONASS with only 18 active satellites does not. The upcoming Galileo system with 27 satellites will be globally available and is intended to have better coverage around the polar regions than GPS. Noticeable for these systems is that Galileo will be the first civilian developed navigation system under civilian control [13]. Some GNSS receiver units have multiple built-in GNSS receivers, e.g. GPS and GLONASS, to improve satellite coverage [73].

All GNSSs require visibility of at least four satellites to acquire 3D position measurements. Thus GNSS performance is highly dependent on the environment in which the receiver is in, such as urban canyons or rural areas. Further information about GPS, of which much is equal to that of all GNSS, is presented in section 3.1.

2.2 Inertial navigation system

Most of the early development of INS was made during the Cold War, where scientists and engineers worked on improving rocket guidance using gyroscopes [25]. While the first technical gyroscope was created in 1817 by Johann Gottfried Friedrich von Bohnenberger, it took until 1920s to achieve a commercial accelerometer product, which was created by McCullom and Peters using Wheatstone bridges [40]. In land vehicles such as automobiles INS are often used together with GNSS in order to provide a more accurate absolute position and orientation estimate [25]. INSs use inertial measurement units (IMU) to

measure inertial forces of the object. By knowing the initial position, measurements of the changes in inertial forces created when the object is moving is used to continuously calculate a new position.

3 Sensor theory

To receive information about the surroundings in the traffic and the dynamics of the vehicle, different types of sensors are required. This section will provide information regarding sensors through the perspective of absolute position estimation. Sensor that are addressed in this section, GPS, IMU, odometer, camera and radar, are available in the test vehicle, provided by Volvo Car Corporation, that will be used in our data collection (section 6.2).

3.1 Global Positioning System

Plans of a global satellite based navigation system dates back to the 1970s. The navigation system was to serve the US military and its allies, however in the early 1980s the project transformed into a civil-military utility [63]. The GPS system was launched in 1993 and became fully operational in 1995. "Selective Ability", which enabled the US military to achieve more accurate position measurements than other recipients, was discontinued in 2000, thus allowing full GPS access worldwide to all recipients and enabling an accuracy below 10 meters RMS [28]. According to information provided by the US department of defense, GPS receivers can achieve a position measurement accuracy of 3 meters horizontally, and 5 meters vertically, 95 % of the time [26]. However this is highly dependent on the surroundings of the receiver, see section 3.1.1.

GPS receivers can also estimate velocity by analyzing Doppler measurements and correct them according to satellite movement, which otherwise would cause measurement errors. However velocity measurements may not be accurate enough for some applications [20], thus in automotive vehicles wheel odometers could be used to provide velocity information instead (section 3.3).

With a constellation of 24 operational satellites working in six different orbits, four satellites on each orbit, the GPS system can achieve coverage all over the world, which implies visibility of between four to ten satellites continuously. To have the GPS system working, accurate time stamps have to be embedded together with information about the satellite position in order for the receiver to estimate its position. This accuracy in time and position of the satellites are made possible by four atomic clocks available on each satellite [20].

Different time standards are used in different GPS receivers, including Coordinated Universal Time (UTC) and GPS time in which GPS-time is linear, and UTC is not. UTC instead uses leap seconds, which implies that one second is added to the time scale at approximately every 18'th month [22]. As of April 2012 GPS-time is 15 seconds behind of UTC [46], [2].

All satellites require continuous time and position synchronization in order to ensure position accuracy, a synchronization that is monitored and controlled from a master control station (MCS) in Colorado Springs, USA. Although GPS satellites have both upload and download possibilities via the S- and L-band respectively, users such as mobile phones with GPS receivers and even military vehicles and aircrafts are only allowed to receive information [78]. It is only the MCSs that can use the S-band for uploading, e.g. to perform satellite corrections. Because of this one-way communication an absolute position calculated by a receiver is not automatically known to other subjects [20].

The GPS receiver uses the downloaded satellite data to calculate the distance the signal has travelled, often called pseudorange, by estimating the Time-Of-Flight (TOF), i.e. the time between the signal was sent and received, and knowing the velocity of the transmitted signal. If pseudoranges to four or more satellites can be calculated, position measurements of the receiver can be achieved using triangulation [20], [78]. In addition to calculating the pseudorange using TOF, pseudoranges can be calculated by determining the phase of the carrier wave signal [79].

National Marine Electronics Association (NMEA) has developed a protocol, providing information in data sentences, for data transmission which most GPS receivers utilize. From these sentences different measurements can be extracted such as position, velocity, heading and several signal quality measurements [69].

For applications that have high accuracy requirements, such as active safety, a GPS receiver cannot solely provide sufficient absolute position estimation, which is due to several complications.

3.1.1 Limitations of the GPS

In order to enable position measurements using GPS, information from four or more satellites are required to pinpoint a location. If the number of visible satellites are less than four, the receiver will not be able to estimate its position. If the number of satellites is above four then every additional satellite information received can result in increased position measurement accuracy. The way that the measurement accuracy increases by the number of visible satellites is because of redundancy, which implies that the system becomes more robust to disturbances in the signals received from the individual satellites (see also dilution of precision section 3.1.1).

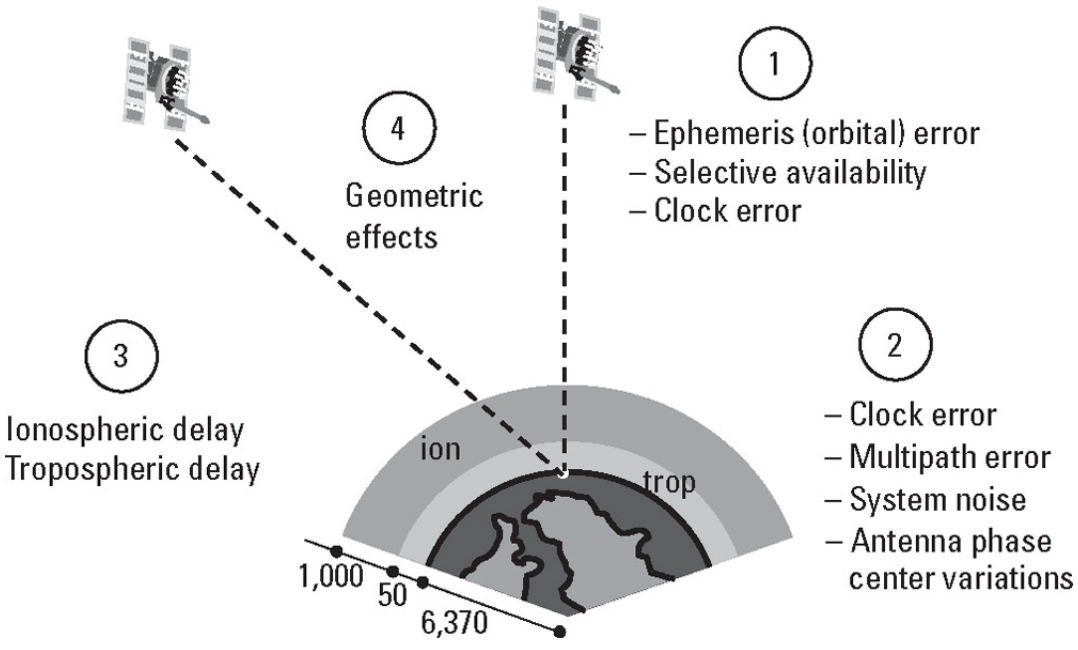


Figure 3.1: Common GPS errors and biases [20].

Errors in the receivers ability to estimate a proper position originates from either the transmitted satellite, during signal propagation or in the receiver. There are numerous of physical influences acting on the GPS system and its ability to determine the position of a receiver.

There are many errors and biases that affect the GPS system from satellite to receiver. Clock error, orbital error, relativistic effects, ionospheric and tropospheric delay, multipath error, antenna phase center, and instrumental biases and noise are the most common [20], [78], see figure 3.1. However only the major ones are comprised in this master thesis.

Satellite errors and relativistic effects Although the GPS satellites each have four atomic clocks onboard, there will always be a small timing error. Position of each satellite has to be known, orbital errors occur when a satellite is moving out of orbit, thus the position needs to be corrected.

Relativity is affecting the GPS system in several ways; satellite orbits are effected by relativistic perturbations due to the gravitational field of the earth, satellite signals are influenced by the earths gravitation in that sense that space-time curvatures are created, satellite clock frequency is influenced by the the gravitational field creating time differences, and fourth and last the receiver clock is also effected by a frequency shift [32]. However the receiver clock will have an error of 10 ns per three hour cycle, which can be regarded as small [32], and in fact this timing problem is corrected when synchronization is made from the MCS (section 3.1).

Atmospheric effects It is only in vacuum that GPS signals can travel at the speed of light, so when the transmitted signals enters the atmosphere, particularly the ionosphere and troposphere, a delay will be introduced [20]. However the delay caused in the ionosphere can be measured and corrected (eliminated) [51], [78]. One of the major disturbances in the GPS system include tropospheric delay, or tropospheric refraction. It occurs when the GPS signal enters the lowest part of the earths atmosphere, called troposphere, where the signal gets refracted. This distortion is difficult to estimate because it depends on temperature, air pressure and humidity, which decreases the accuracy of the measurements [78].

Multipath errors Multipath errors are due to reception of one or more reflections of the same GPS signal, caused by the surroundings, thus corrupting the distance calculations in the receiver unit [8]. Reflections are often caused by buildings, trucks or other large objects, and the error could grow up to 100 m according to [32]. There are methods to attenuate multipath errors, including the use of Right-Handed Circularly Polarized (RHCP) antennas [78] and another method that uses the Signal-to-Noise Ratio (SNR) [8]. A third method is to use a so-called choke-ring antenna, which reflects signals coming from shallow angles that clearly have been reflected once or more during its propagation [34]. However the best way to avoid multipath errors is to keep the receiver at a location where the risk of reflections are minimized.

Receiver errors and noise The antenna phase center is the time when the GPS signal is received, and does not often coincide with the actual physical center of the receiver antenna, which can cause pseudorange calculation errors [32]. Depending on the quality of the electronics of the receiver measurement noise varies, but many GPS receivers executes a self-test on startup to minimize these effects [20].

Blocked signals In addition to the errors and biases mentioned above a major issue regarding GPS-based positioning remains. Although large objects, buildings and mountains can cause multipath errors, another problem is that they also can cause signal blocking. Signal blocking is most common in urban environments (or urban canyons as they often are

referred to) [14]. If fewer than four satellites are visible to the receiver, position estimation by solely using an unaided GPS will not be possible.

Dilution of precision The accuracy of GPS position measurements can be evaluated using information such as standard deviations, signal-to-noise ratio and dilution of precision (DOP). DOP is depending on the current geometry of satellite constellation [45]. The geometry relative the receiver changes and affects the precision of the measurements [71]. There are different types of DOP signals, e.g, vertical (VDOP) and horizontal dilution (HDOP). Depending on the GPS receiver, different messages can be acquired [38].

Figure 3.2 demonstrates an example of good and bad DOP. A low DOP value (good DOP) often results in a higher precision of the position measurements [45], and DOP values are by definition ≥ 1.

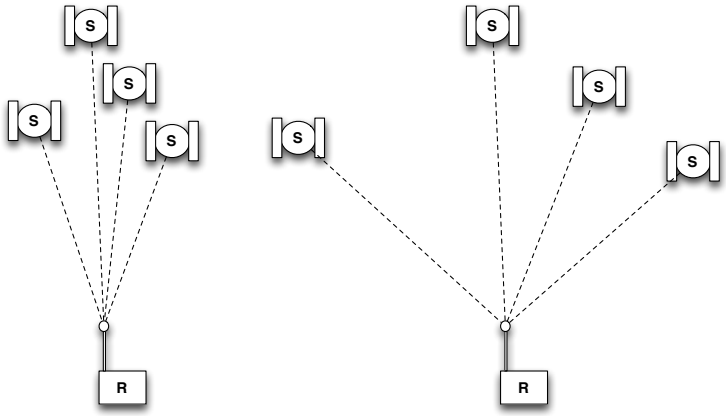


Figure 3.2: Geometric dilution of precision. Bad and good DOP in the left and right-most figure, respectively. Satellites are marked with S, and receiver with R.

The effect of DOP is explained further in Figure 3.3, where satellite errors are included [45]. Red intersections mark the area of possible positions. For the sake of simplicity, only two satellites are included in the figure.

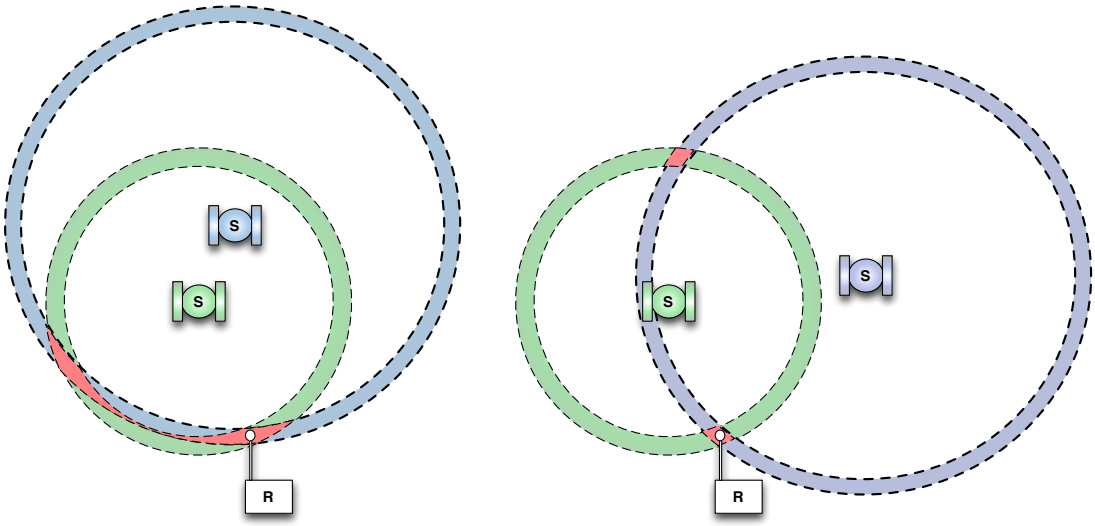


Figure 3.3: Satellites (S) and receiver (R). Higher DOP value in the left-most figure.

To conclude, the DOP decreases with an increased spread of the satellites in view. Note that Figure 3.3 depicts a two-dimensional example, which requires three satellites to narrow the uncertainty region (marked with red) to one zone. If there is a narrow view of the sky DOP can be expected to be high.

3.1.2 Differential and Real Time Kinematic GPS

Numerous methods to reduce the effects of errors and limitations of GPS have been developed over the years. Two popular tools or methods are the differential GPS (DGPS) and the real time kinematic GPS (RTK-GPS), which both make use of information from a fixed 'base' station.

The base station has its position known with high accuracy, and by continuously measuring the absolute position with a GPS receiver, and calculating the difference between these, a correction is acquired [20]. Both DGPS and RTK-GPS use information from the 'base' to correct the measurements in the receiver unit, also called 'rover'. These methods mainly allow correction for atmospheric propagation errors, if the same satellites are visible by both the 'base' and the 'rover' [44], [79]. The main difference between DGPS and RTK-GPS is that DGPS sends the calculated pseudorange correction to a rover, and RTK-GPS sends a carrier-phase correction [20]. The main drawback of DGPS and RTK-GPS is that the receiver, or 'rover', has to be within a certain range of the 'base' in order to receive corrections.

3.2 Inertial measurement unit

Inertial navigation systems (INSs) incorporates inertial measurement units (IMUs) that combines inertial sensors such as linear accelerometers and rate gyroscopes, whose measurements are relative to inertial space, into one platform [60]. Gyroscopes can be used for measuring both rotational velocity, called 'rate gyroscopes', as well as a rotational angle, called 'displacement gyroscopes'. Accelerometers are used to measure acceleration, which can be exemplified by displacement measurements of a mass-spring system [25], [60]. By knowing angular rate and or acceleration of an object, its change in position can be calculated by integrating the signal over time [72].

IMU sensors can use gimbal or strapdown technology, where the latter one is more compact and common but requires more advanced signal processing to match the performance of a gimbal IMU. Apart from incorporating IMUs, an INS also consist of one or more navigation computers that calculates the gravitational acceleration [25].

IMUs are commonly based on Microelectromechanical systems (MEMS), which are cheaper at the cost of decreased performance. The accuracy of MEMS gyroscopes and accelerometers increase all the time, and the accuracy of a MEMS gyroscope can be as good as $0.01\,\mathrm{deg}/h$, and around $1mg$ (g as in gravitational acceleration) for MEMS accelerometers [72]. Gyroscopes and accelerometers have common error sources such as drift, or random walk, which is due to noise, and also bias errors [77]. Errors acting upon gyroscopes and accelerometers can be divided into four categories; fixed or repeatable terms, temperature induced variations, switch-on to switch-on variations, and in-run variations such as random walks. Fixed or repeatable errors can be predicted, thus compensated for, temperature differences can also be corrected through calibration while the other two types of error are more difficult to compensate for [72]. It is the gyroscope accuracy that limits the performance of most INSs [77], [25]. Table 3.1 below summarizes the advantages and disadvantages of IMUs for navigational purposes.

Table 3.1: Advantages and disadvantages of IMUs for navigational purposes [25], [77], [72]

Advantages	Does not depend on external support or weather and lighting.
	It is a natural component in sensor fusion for position estimation and navigation due to outputs such as velocity and acceleration.
	Does not interfere with other sensors that requires antennas or receivers.
	Cannot be subject to external disturbances (jamming).
Disadvantages	The position and attitude estimate diverges over time, i.e., errors increase over time
	Costs are higher than for example GPS receivers, because of longer calibration time and maintenance costs, which are due to higher fail rate than GPS receivers.

Through the development of smaller IC circuits, MEMS inertial sensors are now common in the automotive industry. MEMS sensors are based on silicon alloys, and the technology to make them cheaper, smaller and more accurate is getting better and better [40]. Today it it not possible to have sub-meter accuracy in position, using MEMS based INS, for more than one minute [25]. However, in the future micro-opto-electro-mechanical system (MOEMS) sensors could improve performance even further [72].

3.3 Odometer

A wheel odometer, sometimes referred to as an odograph, is an instrument that measures the distance travelled by counting the number of revolutions [4]. Miscalculations can be caused by slippage, and the odometer accuracy varies depending on environment [62] such as temperature and air pressure in the tires.

3.4 Camera

The average age of the world's population is increasing, and as an effect of this the average driver's ability of seeing in the dark has decreased. To increase road safety and aid drivers at night many vehicles implements night vision cameras. Because road safety is of major concern in the automotive industry, much research is done in this field.

The aim is to improve the ability to autonomously detect pedestrians and other critical objects on the road such as traffic signs and road lines, but also to reduce the cost of such systems [48]. Later research also focus on using the already implemented night vision cameras into other types of active safety systems, e.g., measuring distance to other vehicles or track near environment features to increase accuracy of the position estimate created by sensor fusion with a GPS receiver [48].

Cameras can operate in different frequency spectra. Night vision cameras that are common in vehicles often use either a near infra-red (NIR) or far infra-red (FIR) frequency spectra. By capturing the reflected radiation of frequencies close to the spectrum visible to a human eye, the NIR system yields an output image resembling a grey scale camera. The FIR system has a frequency spectrum in a higher range than NIR and is dependable on thermal radiation of the visible objects to create an image. Since NIR systems are available at a low cost, they are commonly used in vehicles [74].

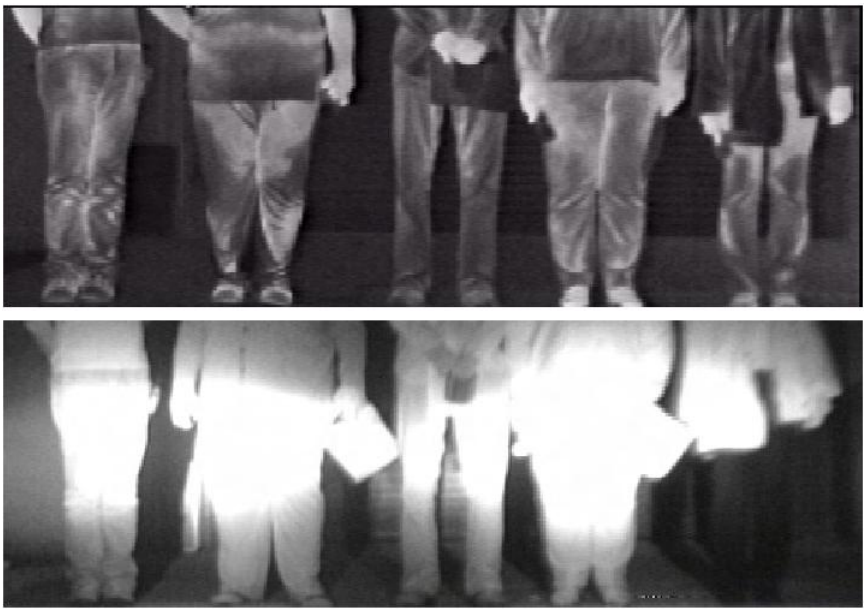


Figure 3.4: An image of pedestrians in FIR (top) and NIR (bottom) [74].

A monocular black and white camera can be found in the Volvo test vehicle that was used during our data collection (section 6), where both range and range rates can be estimated. The estimation accuracy of range and range rate decrease with increased distance to target. [70].

Near infra-red cameras

As depicted in Figure 3.5, NIR is closer to the visible spectrum than FIR, and images provided from a NIR system can be regarded as having a natural look that therefore are easy to interpret [74]. In order to extract an image using NIR systems illumination is needed, hence NIR is often described as an active system [37]. Not only can NIR systems cause glare between each other, but other illumination sources such as traffic and street lights and oncoming traffic lighting may also cause glare and blooming [74]. However when functioning properly, NIR systems can provide main beam range vision without causing dazzle to oncoming traffic [37].

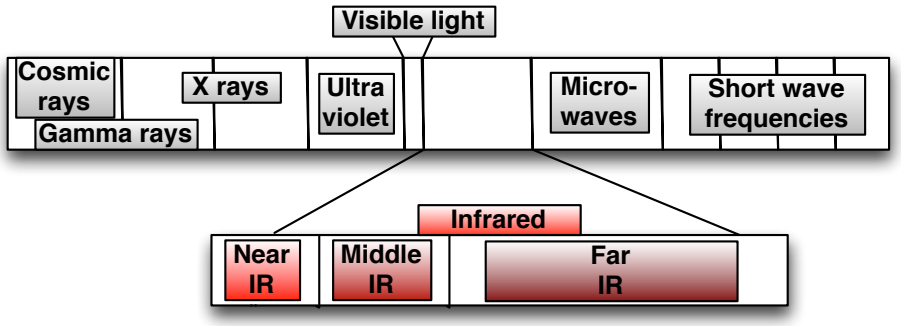


Figure 3.5: Spectrum depicting different wavelengths, note the relation between visible light and infrared waves.

Far infra-red cameras

While the image provided by a NIR camera is considered to be natural looking and easy to interpret, the resulting FIR camera image is considered as difficult to interpret due to its unnatural look [74]. Because FIR makes use of the thermal radiation, where black bodies at a temperature of 300 K have the highest radiation energy in this spectrum, pedestrians, animals and automotive vehicles in use appear very clear [74], [37]. When it comes to estimating an absolute position of a vehicle with the aid of cameras, tracking road features such as lane markings is a common approach. However it is difficult for a FIR camera to detect lane markings.

Usage of FIR systems has to be done under consideration of a number of circumstances including outdoor temperature, wind, humidity and precipitation [74]. An advantage of the passiveness of the FIR system is that it cannot cause glare towards other systems or drivers. Within the military FIR has been traditionally used as a method of detecting targets, and NIR has been used to improve visibility where there has been good conditions for illumination. FIR cameras are more expensive than NIR cameras [74].

3.5 Radar

Radar is a well-known system determining position and velocity of objects. In 1904 Christian Huelsmeyer created the first 'telemobiloscope' as an anti-collision device for ships and the big break-thru for radar technology came in the 1930s at the upcoming of World war two. Countries at that time wanted to detect approaching enemies and Great Britain became a leading developer of radars, with radar antennas positioned around their coastline [23] [10].

Radar systems use the fundamental properties of electromagnetic waves to evaluate the time difference between a transmitted signal and its reflection, or echo, which implies that if a transmitted signal hits an object, the signal is reflected back to the radar receiver antenna. By measuring TOF, the distance to an object can be calculated [80].

According to Peyton [80] a radar system is subject to three different problems. The first is *resolution*, which represents the ability of detecting and distinguishing objects from each other, which can be difficult if multiple objects are close simultaneously. With higher bandwidth the resolution in range improves but with lower bandwidth there is a better resolution in frequency.

The second problem is *detection* and how to determine an object from noise and clutter. Noise is often easy to remove by improving the signal to noise ratio using filtering or better equipment. Clutter is defined as unwanted echo caused by, e.g., weather, unwanted objects, and the ground. To remove clutter advanced signal design and processing methods are often required [64].

The third problem concerning radars systems is *measurement*, i.e., what to measure. As described above, range and velocity are standard radar measurements. In modern radar technology more advanced radars incorporate methods to measure size, shape and type of objects such as vehicles, pedestrians and buildings [80].

If a target detected by radar is moving, the frequency of the transmitted signal will have changed upon reception, also known as 'Doppler frequency' [49]. A negative change in frequency represents a negative velocity of the object relative the transmitter, which means that the detected target is approaching the radar system (Figure 3.6).



Figure 3.6: Doppler effect for moving objects where d and d' are the wavelengths of the transmitted and received signal, respectively.

Radars are often used in automotive vehicles for active safety implementations and are often combined with cameras to achieve better estimations. The benefit of radars is the long range resolution and the ability to detect objects. The camera is then used for identify objects and also follow them with its wide angular resolution [43].

4 Sensor fusion theory

Successful sensor fusion can be observed everywhere, and a good example can be derived from the animal kingdom [30]. Animals have always used a combination of different sensors and they all use them differently. No animal would survive with only vision, but, e.g., with the combined use of hearing and the ability to smell, animals can sense potential danger and act according to their interpretation. Many animals compensate their lack of vision and hearing with a combination of other types of sensors, which has close resemblance to industrial sensor fusion [30].

The objective with sensor fusion is to combine measurements from different sensors to improve the quality of information such that the combined information becomes more valuable than the information would be from each individual sensor [52]. This implies several independent measurements, but not always the use of different sensors. Sensor fusion can be divided into the following categories; *fusion across sensors*, *fusion across attributes*, *fusion across domains* and *fusion across time* [75].

Fusion across sensors is when several sensors measure the same attribute, e.g. the velocity of a vehicle, where the combination of several sensors can increase the robustness of the measured data. The category *fusion across attributes* involves several sensors that measure different attributes, e.g. acceleration, position and angular velocity of a vehicle. This type of sensor fusion increases the value of data by combining information from different sources, thus acquire data regarding different dynamics and aspects of an object.

Next category is *fusion across domains* where several sensors measure the same attribute but at different range, e.g. cold and warm ranged temperature sensors. Such sensor setup can be important when, e.g temperature measurements are made in a temperature-varying environment, where a single sensor might have a limited range.

The final category of sensor fusion is *fusion across time* where the same attribute is measured but at different times, which could imply increased reliability of a measurement.

Another description of sensor fusion is that it could be divided into following three parts; *complementary*, *competitive* and *cooperative* [21]. *Complementary* fusion is described as implementation of independent sensors, where each sensor measure different parts of a system to obtain the information needed for the system as a whole. In *competitive* fusion different sensors measure the same attribute to make the system more robust. Finally

cooperative fusion is used when a combination of sensors obtain information, which would not be possible for a single sensor [21].

4.1 Kalman filter

A frequently used sensor fusion algorithm is the Kalman filter, which is a recursive filtering method for discrete data that was developed by Rudolph Kalman and published in 1960 [39]. The Kalman filter enables estimation of past, present and future states of linear systems by using measurements in a fashion that minimizes the least mean squared error (LMSE) [76].

Many systems are nonlinear and in those cases the Extended Kalman Filter (EKF), a filter where the system is linearized around a working point, can be used [66]. The Kalman filter is an algorithm that is widely used in sensor fusion applications, partially due to its computational efficiency [52]. Following equations describe the mathematical basis of the Kalman filter. Theory and equations of the Kalman filter in this section were provided from [76], [66] and [52].

The states, $x \in \mathbb{R}^n$, are a part of a linear discrete-time process described in Eqn. 4.1 below.

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{4.1}$$

Where

Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ relates the state $\mathbf{x}$ from time step $k-1$ to k, and can be either constant or changing in time. The matrix $\mathbf{A}$ is often referred to as the process model of the Kalman filter.

Matrix $\mathbf{B} \in \mathbb{R}^{n\times l}$ relates the control input $\mathbf{u}$ to the states from time step $k-1$ to k, and can be either constant or changing in time.

Vector $\mathbf{u} \in \mathbb{R}^l$ is the control input, which is optional.

Vector $\mathbf{w} \in \mathbb{R}^n$ is a random variable representing the process noise of the system, which is assumed to be white and normally distributed, $\mathbf{w} \sim N(0, \mathbf{Q})$, where $\mathbf{Q}$ is the process noise covariance matrix that can be either constant or changing in time.

The states $\mathbf{x}$ are effected by measurements $\mathbf{z} \in \mathbb{R}^m$ (see Eqn. 4.2) that, e.g., can be data from sensor outputs.

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{4.2}$$

Where

Matrix $\mathbf{H} \in \mathbb{R}^{m\times n}$ relates the state $\mathbf{x}$ from time step from k to measurements $\mathbf{z}$ at time step k, and can be either constant or changing in time. The matrix $\mathbf{H}$ is often referred to as the measurement model of the Kalman filter.

Vector $\mathbf{v} \in \mathbb{R}^n$ is a random variable representing the measurement noise of the system, which is assumed to be white and normally distributed, $\mathbf{v} \sim N(0, \mathbf{R})$, where $\mathbf{R}$ is the measurement noise covariance matrix that can be either constant or changing in time.

The Kalman filter first calculates a primary *a priori* estimate (superscript "-"), given process information prior to step k - $\hat{\mathbf{x}}_k^-$, and then a final *a posteriori* estimate of the state, given measurement $\mathbf{z}_k$ at time step k - $\hat{\mathbf{x}}_k$. The aim of the Kalman filter is to estimate the state $\mathbf{x}$ given measurement $\mathbf{z}$ ($\mathbf{x}$ conditioned $\mathbf{z}$), in which Bayes' theorem can be applied,

$$p(\mathbf{x}_k \mid \mathbf{z}_k) \sim N(E[\mathbf{x}_k], E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]) = N(\hat{\mathbf{x}}_k, P_k), \tag{4.3}$$

where

$E[\mathbf{x}_k]$, the first moment of the state distribution, is the expected value of the state $\mathbf{x}$ at time step k, also defined as the *a posteriori* state estimate $\hat{\mathbf{x}}_k$.

$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]$, the second moment of the state distribution, is the covariance of the state estimate error $e_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$, also defined as the *a posteriori* estimate error covariance P_k.

To calculate the *a posteriori* state estimate the *a priori* state estimate information is needed, where the relation is defined as

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - H\hat{\mathbf{x}}_k^-), \tag{4.4}$$

where

the term $\mathbf{z}_k - H\hat{\mathbf{x}}_k^-$, also called the residual, reflects the difference between prediction and measurement.

matrix $\mathbf{K}_k \in \mathbb{R}^{n \times m}$, known as the Kalman gain, acts as a weighting of the residual in the *a posteriori* state estimate calculation. The Kalman gain K_k is calculated for every time step k using the covariance matrix of the *a priori* state estimate error P_k^-.

The *a priori* state estimate error covariance matrix can be defined as

$$P_k^- = E[(e_k^-)(e_k^-)^T] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)] \tag{4.5}$$

Since the error of the *a posteriori* state estimate is to be minimized, which depends on the *a priori* state estimate error covariance P_k^-, the Kalman gain K_k has to be selected to achieve this. By performing some substitutions and expectations according to the equations above, taking the derivative of the trace with respect to K_k and putting it equal to zero, the Kalman gain K_k can be derived, see Eqn. 4.6.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{4.6}$$

If the measurement error covariance matrix R decreases, the Kalman gain will increase and thus the Kalman filter will "trust" the measurements to a wider extent. When the *a priori* estimate error covariance matrix P_k^- decreases the Kalman gain will decrease and thus the Kalman filter will rely on the *a priori* state estimate to a wider extent.

Hence, the Kalman filter can place weights indicating how it relies on the measurements and the internal process model, respectively, in a continuous fashion by setting a specific value on K_k. This enables the Kalman filter to still produce valid state estimates at instances when the measurements are of lesser quality, given that the process model is accurate.

4.1.1 Implementing a discrete Kalman filter

There are different versions of the Kalman filter, with different application areas, however a standard version for linear systems is the discrete Kalman filter. Depicted in figure 4.1 are the five steps of the Kalman filter operations cycle, which require initial estimates $\hat{\mathbf{x}}_{k-1}$ and P_{k-1}.

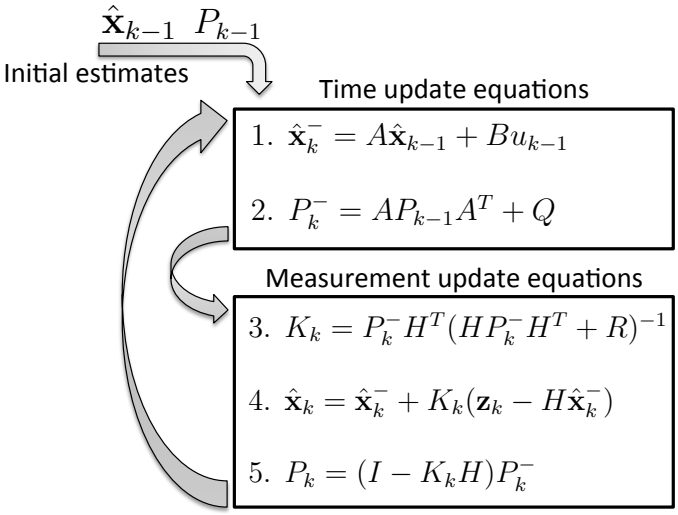


Figure 4.1: Linear Kalman filter operations cycle, I is the identity matrix.

Sometimes the *time update* equations are referred to as *the prediction stage*, while the *measurement update* equations constitutes *the correction stage*.

As stated previously the initial estimates, $\hat{\mathbf{x}}_{k-1}$ and P_{k-1}, are required in order for the Kalman filter to start its iterative process, which can be acquired through user input or from some external source. When analyzing steps one to five in Figure 4.1 there are a few matrices that require parametrization prior to executing the Kalman filter iterative process;

state transformation matrix A,

the *optional* control input transformation matrix B,

measurement noise covariance matrix R,

process noise covariance matrix Q,

and the state-to-measurment transformation matrix H.

Elements in the A-matrix is chosen such that it includes, for every state in $\mathbf{x}$, both the previous state $\mathbf{x}_{k-1}$ and alternatively also the derivatives of that state, which together forms the current state $\mathbf{x}_k$. Depending on the system you apply the Kalman filter on, control input $\mathbf{u}$ is existing or not. In sensor fusion where there is no direct control action applied, e.g. in form of an actuator, there is no need to consider $\mathbf{u}$ and thus neither matrix B. In general, a passive system, which acquire sensor inputs but does not physically act upon the information given, does not have any control input $\mathbf{u}$ and thus no B-matrix. However an active system, which sense the surroundings and react upon them, will require a control input $\mathbf{u}$ and thus the B-matrix has to be defined in accordance to the system behavior.

Measurement data collection has to be made in order to estimate the parameters of the measurement noise covariance matrix R, which is usually done off-line. In cases where, e.g., there are variations in the environment where sensor data is acquired, R_k can be defined as a variable, which is updated iteratively depending on the situation either autonomously or through user input.

A more difficult task is to estimate the process noise covariance Q [76], since the process often is not fully observable but is instead approximated using a mathematical model. In other words, a reliable (but often complex) process model requires less weightings in the Q-matrix than a simple model containing a lot of approximations to reduce both complexity and or computational power.

When it comes to the *state-to-measurment* transformation matrix H you have to know the relation between state and measurement. Consider an example where the velocity of a bicycle is defined as a state within a Kalman filter, and that an odometer is mounted on the front fork of the vehicle counting the number of revolutions of the wheel. By knowing the physical dimensions of the wheel, the measurement of wheel count can be translated into velocity. In the same way velocity can be reversibly transformed back into wheel revolutions, i.e. a *state-to-measurement* transformation.

4.1.2 Extended Kalman filter

As described in the beginning of section 4.1, the Kalman filter requires a linear system. To deal with cases regarding non-linear systems, such as the one this thesis addresses, a modified Kalman filter has to be used. In this thesis the emphasis has ben put on the well-known non-linear *extended Kalman filter* (EKF), in accordance with the study of previous research made in the field of improved absolute position through sensor fusion (section 5).

By linearizing about the estimate of the current mean and covariance, given the Taylor series, Jacobian matrices are obtained that substitute the linear transformation matrices in the Kalman filter equations [61]. However, there is a drawback with using the extended Kalman filter, apart from the approximation using linearization, which is that random variables are no longer normally distributed when subject to nonlinear transformations [5]. When the Jacobians have been calculated the same Kalman filter cycle equations, presented in Figure 4.1, are used.

Using the same notation for the state vector $\mathbf{x}$, the control input $\mathbf{u}$ and the process noise $\mathbf{w}$, as the linear Kalman filter, following non-linear process difference equation is defined as

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}). \tag{4.7}$$

Since we make no use of the control input $\mathbf{u}$ in this thesis, it will be omitted in further equations and discussions. The measurement $\mathbf{z}$ is defined as

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k). \tag{4.8}$$

Because these equations are non-linear, they have to be re-written on a linear form [76], [24],

$$\mathbf{x}_k \approx \tilde{\mathbf{x}}_k + F(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \Gamma\mathbf{w}_{k-1}, \tag{4.9}$$

where

variable $\tilde{\mathbf{x}}_k = f(\mathbf{x}_{k-1}, 0)$, since the expected value of $\mathbf{w}$ is zero,

matrix F, sometimes referred to as the process model, is the Jacobian of the non-linear process function f with respect to $\mathbf{x}$ according to Eqn. 4.10,

variable $\hat{\mathbf{x}}_{k-1}$ is the a posteriori estimate at time step $k-1$,

matrix Γ is the Jacobian of the non-linear process function f with respect to $\mathbf{w}$ according to Eqn. 4.11.

$$F_{i,j} = \left.\frac{\partial f_i}{\partial \mathbf{x}_j}\right|_{(\hat{\mathbf{x}}_{k-1},\ 0)} \tag{4.10}$$

$$\Gamma_{i,j} = \left.\frac{\partial f_i}{\partial \mathbf{w}_j}\right|_{(\hat{\mathbf{x}}_{k-1},\ 0)} \tag{4.11}$$

The non-linear measurement can be approximated in the following manner,

$$\mathbf{z}_k \approx \tilde{\mathbf{z}}_k + H(\mathbf{x}_k - \hat{\mathbf{x}}_k) + \Phi\mathbf{v}_k, \tag{4.12}$$

where

variable $\tilde{\mathbf{z}}_k = h(\tilde{\mathbf{x}}_k, 0)$, since the expected value of $\mathbf{v}$ is zero,

matrix H, sometimes referred to as the measurement model, is the Jacobian of the non-linear measurement function h with respect to $\mathbf{x}$ according to Eqn. 4.13,

matrix Φ is the Jacobian of the non-linear measurement function h with respect to $\mathbf{v}$ according to Eqn. 4.14.

$$H_{i,j} = \left.\frac{\partial h_i}{\partial \mathbf{x}_j}\right|_{(\tilde{\mathbf{x}}_k,\ 0)} \tag{4.13}$$

$$\Phi_{i,j} = \left.\frac{\partial h_i}{\partial \mathbf{v}_j}\right|_{(\tilde{\mathbf{x}}_k,\ 0)} \tag{4.14}$$

All four Jacobians calculated above are time varying, but for the sake of simplifying reading the equations, time step subscript k has been left out of Eqns. 4.10, 4.11, 4.13 and 4.14. Figure 4.2 summarize the equations required to perform the extended Kalman filter cycle, which is similar to the linear Kalman filter cycle (see Figure 4.1).

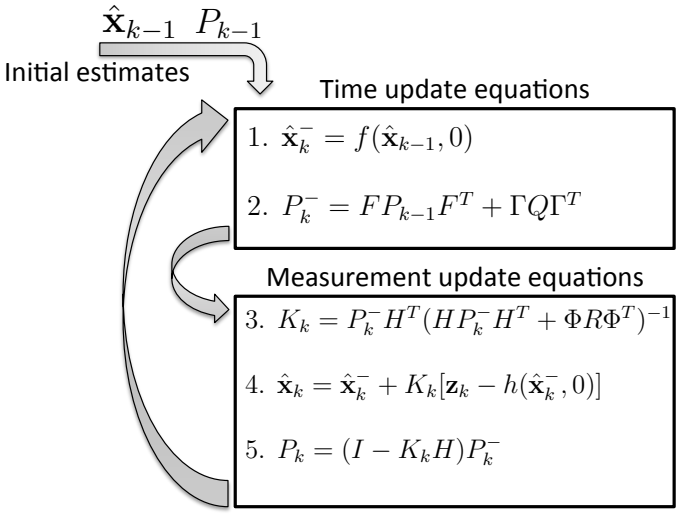


Figure 4.2: Extended Kalman filter operations cycle, I is the identity matrix.

Definitions of K, Q, P and R are described in section 4.1, which is about the linear Kalman filter. Note once again that the equations have the control input u omitted, and that the stochastic variables $\mathbf{w}$ and $\mathbf{v}$ are put to zero when calculating the a priori state estimate $\hat{\mathbf{x}}^-$ and the a posteriori state estimate $\hat{\mathbf{x}}$, depicted in Figure 4.2.

5 Existing research

In order to get a better understanding of how the sensors can be used, fused and incorporated in different methods to improve position estimation of a vehicle, a literature review was carried out. A selection of ten articles, published between 2001 and 2011, were studied that concerned absolute position estimation of automotive vehicles. In this section the most important results and conclusions, as well as recommendations for future work, from these articles will be presented. Note that much of the actual theory acquired from this literature review is included in the sensor theory and sensor fusion theory section (section 3 and 4). The conclusions of this study of previous research functioned as a guideline for the development of our sensor fusion model (section 7).

5.1 Summary of reviewed papers

There is often great focus on sensor fusion between GPS receivers and inertial sensors within absolute position estimation. To maintain good accuracy although using low-cost off-the-shelf sensors is a major focus in most of the articles, and to verify the final performance of the solution, comparison with expensive and highly accurate systems are made.

For example it is common to create a rather cheap GPS/IMU system using Kalman filtering and compare it with an RTK GPS/tactical grade IMU solution. In [28] an absolute position estimate accuracy of 3-4 meters, and a yaw angle estimate accuracy of 3.5 degrees was achieved with one sigma confidence interval using this method. By also using a camera to track far field terrain features, such as curves in the horizon, and match them according to USGS Digital Elevation Maps (DEM), a sub-degree accuracy for orientation was achieved [28]. For information about the GPS/IMU integration see section 5.2.1.

There can be advantages in extracting near field features as well, something that was also investigated using a monocular camera to track road markers and traffic signs along a highway that were matched with a map used as reference (section 5.2.4). This information was fused with measurements from an IMU, and simulations showed sub-degree orientation accuracy and a position accuracy of $25cm$ [28]. To reduce computational complexity during image processing, a region of interest (ROI) was defined when extracting both near and far field features. Traffic sign detection was also used in [62] together with lane assignment, however instead of having the camera measuring the distance it only measures the angle to the sign, an angle that is used do align a laser that is used for distance measurements.

[28] suggested that a vision system combined with map matching should be integrated with GPS/IMU to improve the estimate accuracy, i.e. a combination of the methods incorporated in that article. This type of sensor fusion was carried out in [6], indicating that absolute position estimation is possible even though only two satellites are visible, however results varied between 1 and 10 meters in precision depending what satellite pair that was used. For more information about map matching see section 5.2.3.

Since there are mainly two types of cameras that are used in position estimation, different articles argues about which of them is to be preferred. One vehicle manufacturer suggest NIR over FIR, but according to article [74] the decision to choose between NIR and FIR is foremost based on the perception of how the system should be used. The difference in interpretation difficulty between the systems could be a cause of specifically preferring

NIR over FIR. Because NIR has advantages where FIR has disadvantages, and vice versa, a suggestion is made that they should serve as a complement to each other. Applying both technologies could also render higher redundancy to the estimated output, in this case position estimation. However if the main application of the camera is to increase position estimate accuracy through identification of lane markings, NIR cameras are to be regarded as most suitable.

It was suggested that a vehicle model should be made to take the vehicle dynamics into account [28], but [55] and [9] instead suggests implementation of pattern recognizing algorithm to improve dead reckoning, navigation where only the initial position is known, by classification of vehicle motions such as left and right turns. Other results, conclusions and suggestions mentioned in [28] is listed below.

The GPS provides inaccurate velocity measurements for low speeds (under 8 km/h).

Relative position between IMU and camera must be known when fusing their outputs.

Accuracy of the GPS/IMU fusion estimates were limited by the availability of GPS satellites and the accuracy of the velocity measurements.

Suggestion is made to use a down-faced camera that is dedicated to detect and measure the distance to road markers, also suggested in [6].

Color images could improve the efficiency of the vision system, however this will lead to increased computational effort.

Having the GPS receiver and IMU on the same platform, as well as using a common clock, will make signal processing easier [33]. One way to reduce the drift in IMU sensors is to fuse the sensor output with magnetometer measurements, which allow a reduction in position error from 150 meters to 5 meters obtained after a one minute session [25], it also showed an improvement in the orientation estimates in article [33].

In [55] GPS has been fused with gyroscope, odometer and combined in map matching to get a better absolute position estimate, a solution that has been implemented into Audi, Open and Alfa Romeo automobiles. But still results show that the availability of GPS is the main limitation.

When it comes to sensor fusion, extended Kalman filters, or just Kalman filters, are used throughout almost all of the research, though how to actually perform the fusion is a more difficult issue. Some research including [9] use loosely coupling methods to fuse sensor data between GPS and INS, but states that a tighter coupling method could result in better accuracy of the position estimate. Using an ultra-tight coupling method, which integrates the sensors at tracking loop level of the GPS, [54] successfully increased the SNR of the carrier phase by $14dB$. The drawbacks of ultra-tight coupling is the level of difficulty in creating it, also resulting in increased computational complexity, when compared to loosely coupling. Further information about GPS/INS fusion is presented in section 5.2.1

To improve the performance of the sensor fusion framework even further, rule or fuzzy-based fusion can be applied onto an existing kalman filter fusion framework [50]. This simply implies that the behavior of the kalman filter will adapt according to some predefined rules through additional thresholds and weights. This can either be done in a discrete fashion, where continuous change in the data does not change the outcome of a specific rule, known as rule-based fusion, or in a continuous fashion, known as fuzzy-based fusion. An example where rules could be applied is, e.g., when the vehicle is stationary (non-moving), where increased weights on the GPS position measurements would allow the position estimate to converge faster.

5.2 Methods to improve absolute position estimates

In this section a few of the frequently used methods to improve absolute position estimates are addressed. These methods were described in the articles referred to in section 5 and make use of GNSS receivers, IMU, radar, camera but also other sensors and combines different methods into one final absolute position estimation system. These methods were selected due to their applicability in automotive vehicles, where some car models already have implemented all or most of the sensors mentioned. The studied articles point towards several different implementations for position estimation, however the main focus, or starting point, is often the integration between GPS/INS. The different methods that are presented comes in rising level of complexity. GPS/INS integration is often used in research today and is even implemented in some modern vehicles, while learning algorithms can provide implementations that could be available in the future.

5.2.1 GPS/INS integration

Fusing GPS and INS data is considered a basis towards improving absolute position estimations, in that sense that most research incorporate this solution alone, or as a subset of a final sensor-fusion system. There are a lot of methods to integrate GPS data with data from an INS, and research often discuss various levels of integration such as *loosely*, *tightly* and even *ultra-tightly*. Having a *loosely* integration (see Figure 5.1) implies that the GPS data used for fusion is extracted at the latest stage of the processing within the GPS receiver, i.e. the calculated position, velocity and heading (sometimes referred to as attitude) [9]. A *tightly* coupled GPS/INS-fusion is when the GPS receiver data is extracted in a rawer format, providing the pseudorange, carrier phase and Doppler measurements instead of position, velocity and attitude as in Figure 5.1 [9], [54]. In [54] an *ultra-tight* coupling is suggested, which requires data-fetching as early as in the tracking loops, and having pseudorange, carrier phase and Doppler measurements, to feed back Doppler estimations to the GPS receiver. The loosely coupled approach gain simplicity, but on the other hand it also has the disadvantage of decreased performance and robustness compared to the tightly coupled methods [19].

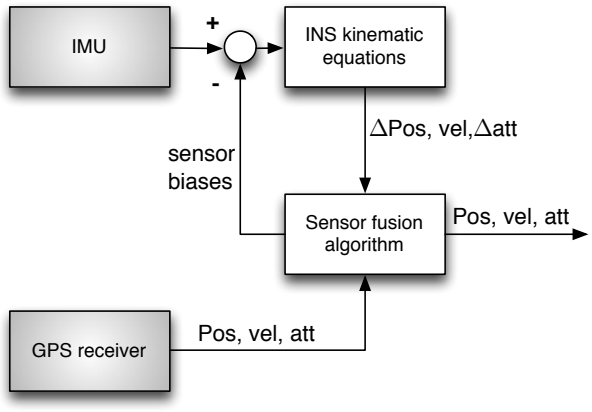


Figure 5.1: Loose coupling between GPS and INS.

It is important to know that coordinate transformations are often required prior to execute sensor fusion in general, as is the case with GPS/INS-fusion. Further details about the coordinate transformations required for GPS/INS-integration is explained in section 7.2.2.

5.2.2 Lane tracking

Research on vision based sensors for automotive vehicles has largely increased during the last years [42]. The aim of using road lines for positioning is to determine the lateral position of the vehicle relative the road by matching the position of lane markings to a map (section 5.2.4). Many manufacturers work today with Lane departure warning (LDW) which often uses cameras or laser systems, such as Light Detection and Ranging (LIDAR), to detect road lanes and functions as a warning system when a vehicle comes close to a road line [6].

Lane detection using vision sensors was starting to develop in 1990s and since then, algorithms for detecting lines has been well established. Also prices for cameras has decreased. A disadvantage of using cameras is that the measurements are weather dependent [6]. If there is for example snow on the road, lane markings are difficult to detect. LIDAR is weather independent but is more expensive [6]

Another problem with using cameras for lane tracking is that colored road lines can be difficult to track since their corresponding grey scale output is similar to that of the road when using a NIR camera [6]. It could also be difficult in heavy traffic since part of a line can be hidden by other vehicles [42]. To analyze road lines with these types of problems, a good algorithm is required. With more computational power, more complex algorithms can be used in real time. For example different type of learning algorithms can be used to increase the robustness of lane tracking using cameras [65].

5.2.3 Map matching

Assuming that the vehicle is placed on a road absolute position estimate accuracy can be improved by using map matching. Map matching is a mathematical association method that selects the most likely road inhabited by the vehicle given an absolute position, and adjusts that position according to the coordinates of the selected road [36]. This adjustment can cause problems when roads are close to each other, and in those cases jumps in the position estimate can occur if the algorithm change the road that is most likely [55]. These jumps and faulty estimates can be decreased by using a combination of sensors, where a weighted output from a sensor fusion system can be matched to a likely position on a map.

Map matching is not a standalone method and requires measurement from sensors which can be different depending on implementation, and it is also dependent on accurate map-data. One article suggest a sensor fusion algorithm with IMU and odometer together with map matching [55]. Another popular sensor fusion is the integration between GPS and IMU together with map matching. Other methods that can be used in combination with map matching is, e.g., Simultaneous Localization And Mapping (SLAM), lane tracking, and traffic sign localization [62], [28].

5.2.4 Traffic sign localization

Recent studies use cameras to detect traffic signs, so-called traffic sign recognition (TSR), and even measure their position relative the vehicle. Laser sensors can however measure angle and distance from the vehicle to the detected object with higher accuracy than a camera. With information of where the traffic sign is located, the absolute position of the vehicle can be determined [62]. This requires accurate information regarding the position of the traffic sign, either through a map or by having the traffic sign transmitting information about its absolute position to nearby vehicles.

Traffic sign recognition and localization requires high computational power. This is mainly because that high update rates are required, that several objects can be detected

at the same time, that objects can be partially hidden, or that objects are difficult to detect due to background disturbances. Future research regarding this method is to also analyze objects other than road signs [27]. Such objects could for example be crossroads or traffic lights.

5.2.5 Learning algorithms

The main idea of learning algorithms is to make use of data acquired by the system in a similar or same situation. There were two learning algorithm techniques comprised in the literature review, SLAM and pattern recognition of vehicle dynamics. SLAM is a method for estimating a position using knowledge from past visits using camera and GPS. The combination of these sensors yield a more accurate position estimate than a GPS receiver alone, especially in urban environments where the accuracy of a GPS receiver is less reliable [60]. The approach is to aid the system by learning and recognizing its surrounding environment. This is accomplished by combining images from a camera with position information provided by a GPS receiver. The algorithm matches images with the current position estimate and compares them to a database. By using this method iteratively the position estimate gets more robust and accurate when a specific position is being repeatedly visited. SLAM is appropriate in bus travels or when commuting, where the travelled path is often repeated [60].

Another learning technique is a type of pattern recognition, which uses inertial sensors instead of camera to recognize the vehicle dynamics. The recognized sensor data pattern from the IMU is mapped into a specific position with the help of a GPS receiver. As with SLAM, this method has increased usability if the road has been visited before, in that sense that a pattern is recognized and can be matched to a specific location [9].

6 Data acquisition

During the time frame of this master's thesis, we collected data using a test vehicle provided by Volvo Car Corporation. Before performing this data collection, we had the possibility to analyze previously collected data. This data set was short and simple, which functioned as a good starting point for testing and evaluating the sensor fusion model we created. By thoroughly analyzing this data, it was easier to know what parameters and aspects to focus on at the time of our own data collection. More information about the previously collected data is given in section 6.1.

When we had acquired sufficient amount of knowledge regarding the collection of sensor data, we performed another data collection using a test vehicle provided by Volvo Car Corporation. In this data collection we acquired measurements from a low-grade GPS receiver (BU-353), an expensive RTK-GPS (RT-3002), as well as odometer and inertial sensors. Our data collection is further described in section 6.2.

Three main data sources were used; vehicle sensor data provided through CAN, expensive differential GPS receiver, and in our data collection also a commercial GPS receiver. There are several CAN networks available in the vehicle, e.g. High-Speed CAN (HS-CAN) and sensor-CAN, where the difference between these two is that HS-CAN provides filtered signals generated by the brake control module (BCM) originated from raw sensor output provided on sensor-CAN.

6.1 Test and development data

This data was collected by our fellow student Martin Ståhl during the summer of 2011, in which he collected data from an expensive DGPS, together with data provided on the CAN bus such as velocity, yaw rate and longitudinal and lateral acceleration. Due to the lack of data from a commercial low-grade GPS receiver, and the limited amount of data in total, this data was used as input to verify and develop our MATLAB code, and also for us to acquire knowledge about important aspects of the data and data collection at an early stage in the project.

These 60 seconds of data was collected on a parking lot at Volvo Car Corporation, Torslanda (Sweden). A plot of the position measurements from the GPS receiver on top of Google Maps is depicted in Figure 6.1 below.

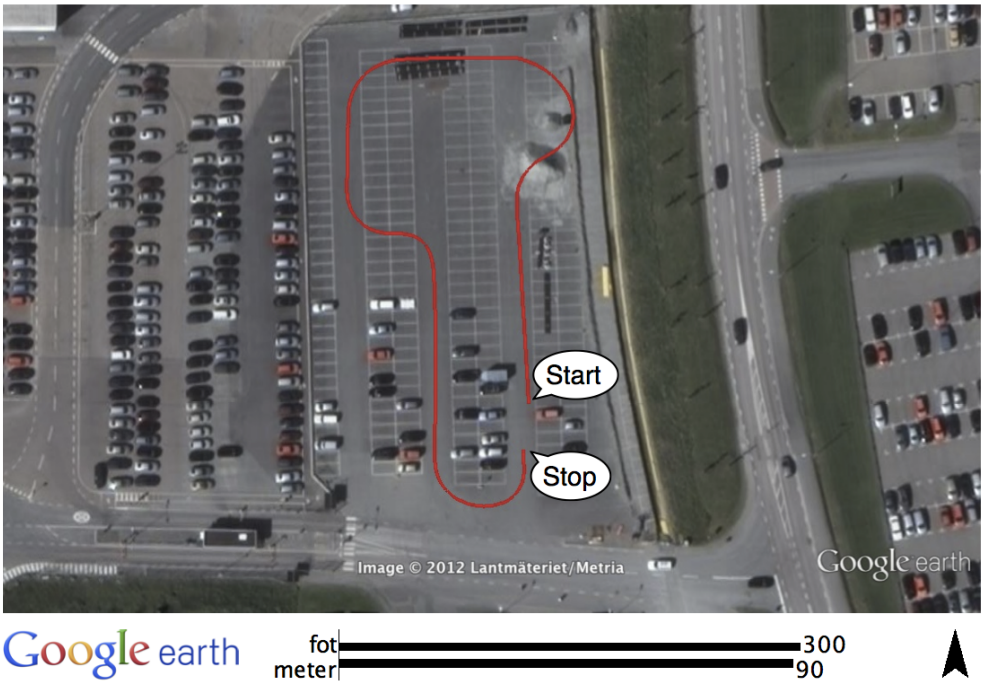


Figure 6.1: Trajectory of the 60 second data collection, plotted on top of Google Earth maps.

The trajectory involves a couple of turns, and ends up near the starting location, thus providing fluctuations in all the sensor outputs. Sensors that were used in this data collection run are provided in the list below.

- RTK-GPS receiver (Trimble SPS852) with the differential ability turned off, thus the measurement accuracy is deteriorated. Trimble SPS852 provides position updates at a rate of up to 20 Hz. For further specifications and information, visit [35].
- Gyroscope, renders yaw rate. A filtered signal on the HSCAN network called *YawRate-Actual* was used.
- Accelerometers, renders lateral and longitudinal relative the vehicle. Filtered signals on the HSCAN network called *LongAccOverGnd* and *LateralAcceleration* was used.
- Odometer, or wheel tick, to measure the velocity of the vehicle. A filtered signal on the HSCAN network called *VehicleSpeedOverGnd* was used.

Sensor specifications of the signals provided on HSCAN were not available, and even if they where the signals provided were filtered in an unknown fashion, making sensor specification almost irrelevant.

Since there are many factors effecting the GPS receiver performance, we did not find it important to include all the specific accuracy information from the provider (Trimble). However the specifications states that the Trimble SPS852 can provide a position estimate better than 5 m 3DRMS [35].

6.2 Our data collection

Because of the insufficiency of the data described in the previous section 6.1, we had to perform a new data collection. The main difference between these data collections is that we collected data from both an expensive RTK-GPS and a commercial low-grade GPS receiver, simultaneously together with inertial and velocity information given on the CAN bus, whereas the data collected by Ståhl did not include information from a low-grade GPS receiver.

This logging equipment was setup to record HSCAN and RT-3002 data, while the NMEA sentences transmitted from the BU-353 was logged separately on a laptop computer.

The logging system in the test vehicle was setup to save data in three-minute (180 seconds) segments, in order to maintain low file sizes which are easier for computers to handle. Note that the existence of obtaining raw sensor output was not known to us until after the data collection. Following list summarizes the sensor equipped onboard the vehicle.

- Reference system RT-3002, with a Kalman filter output rate of 100 Hz, giving 2 cm position accuracy of 1σ provided optimal conditions and differential corrections. Without any corrections position accuracy is 1.5 m circular error probability (50 % probability). Further specifications and information can be found in [56]. RT-3002 contains following measurement units:
  - RTK-GPS receiver.
  - Six-degree-of-freedom IMU containing three gyroscopes (yaw, pitch and roll) and three accelerometers (longitudinal and lateral relative vehicle, and down), providing inertial measurements in three dimensions.
- Low-grade commercial GPS receiver BU-353 working at 1 Hz, providing position measurement accuracy of 5 m 2D RMS, i.e. horizontal position only. Vertical position is available, however at low accuracy [47].
- Gyroscope, renders yaw rate. The signal that was used is called *yawRateActual*, was provided on HSCAN, and is unfiltered but offset compensated.
- Accelerometers, renders lateral and longitudinal acceleration relative the vehicle. Signal *lateralAcceleration* is unfiltered but offset compensated, the signal *longAcceleration* is filtered. Both signals were provided on the HSCAN.
- Odometer, or wheel tick, to measure the velocity of the vehicle. The filtered HSCAN signal *vehicleSpeedOverGnd* was used instead of raw odometer data, which is the result of a complex algorithm depending on driving situation.

Information regarding the sensor signals on HSCAN was provided by the Volvo Car Corporation, further specifications were not provided. The test vehicle we used was a Volvo V60 equipped with, except from the sensors in the list above, a forward-faced camera and radar.

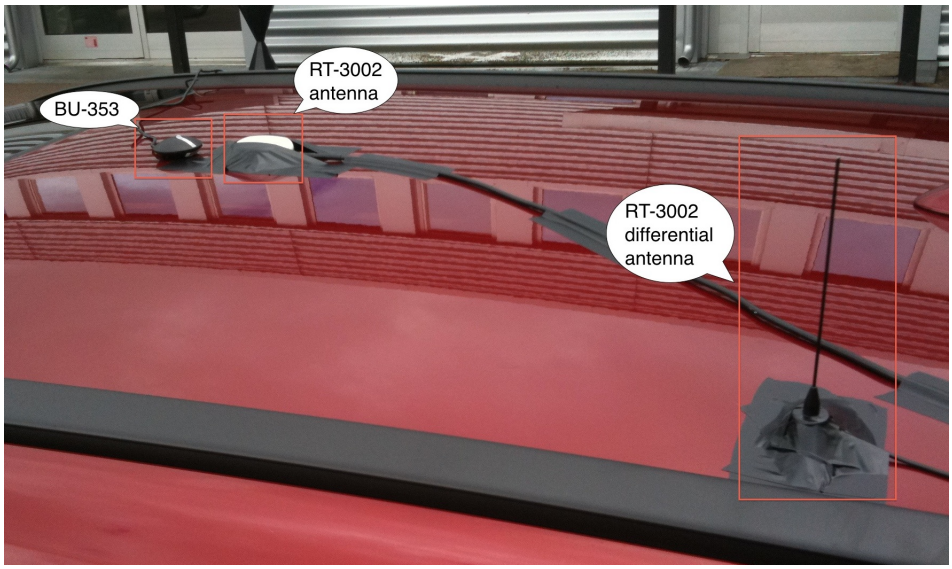


Figure 6.2: Antenna setup depicting the BU-353 unit, RT-3002 GPS and differential correction antenna placed on the roof of the vehicle.

Figure 6.2 displays the roof of the vehicle, showing the BU-353 GPS receiver unit, RT-3002 GPS receiver antenna and the antenna used for receiving differential corrections from a base station.

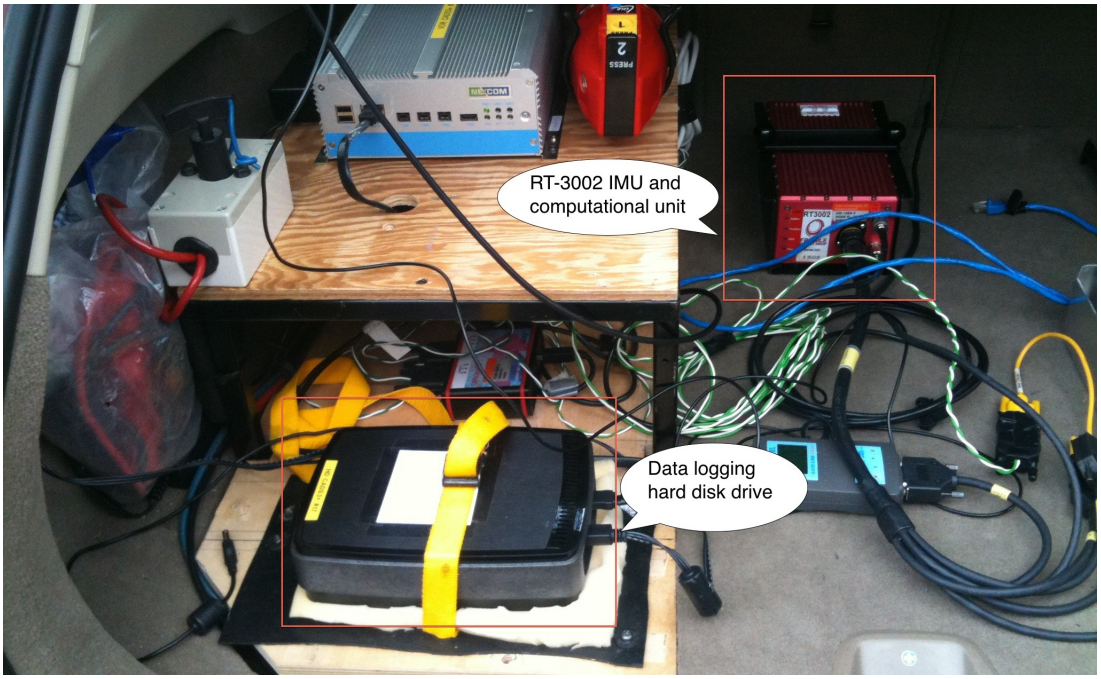


Figure 6.3: Setup in the trunk of the vehicle, depicting the hard disk drive used for logging data and the RT-3002 main unit including it's IMU and processing unit.

Logging equipment was placed in the trunk of the vehicle, presented in Figure 6.3. The hard disk drive was protected with a soft foam and strapped on to the vehicle to avoid damages.

6.2.1 Data collection scenarios

The aim of the data collection was to gather information in different road environments such that different aspects of the sensor fusion model could be evaluated. Our data collection ranged from inner city to small forest road environments. The scenarios selected from this data collection was based upon the optimality for GNSS navigation, and was thus sorted into the following four categories; city, forest road, highway and test track (Hällered).

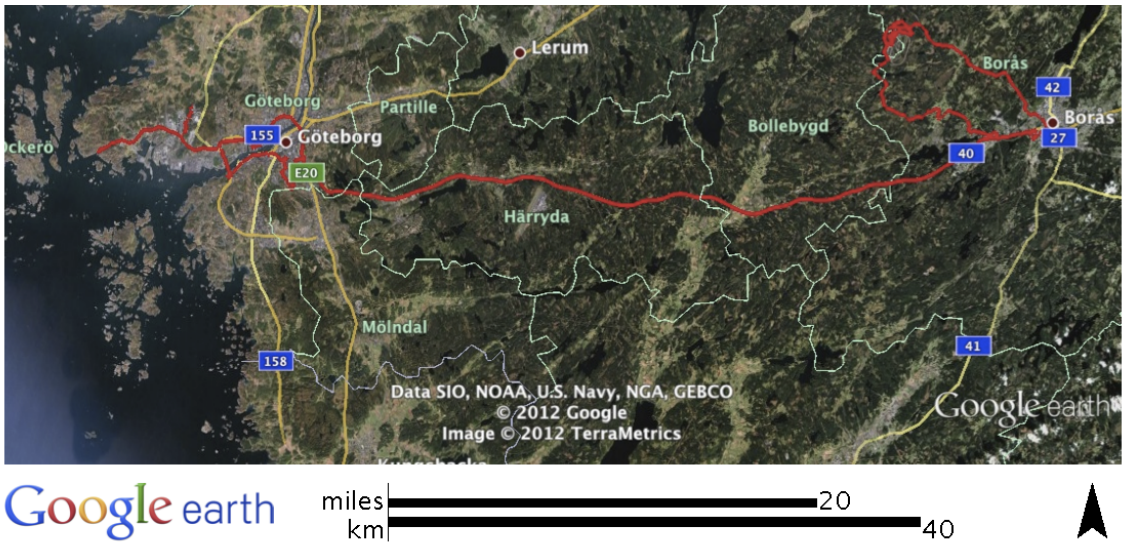


Figure 6.4: Trajectory for the total data collection



Figure 6.5: Images gathered from scenarios; highway, forest road and city (left, center and right, respectively).

Our logging expedition went from from Volvo Torslanda to Hällered outside Borås, where the RT-3002 could make use of differential corrections provided from a base station at the test track, and then back to Gothenburg's inner city (Figure 6.4). Figures showing the comparison of Kalman filter results with BU-353 measurements and RT-3002 position estimates in the different scenarios are presented in the simulation results section 8. Figure 6.5 show images from the front-faced camera of the vehicle at different scenarios.

6.2.2 Sensor characteristics analysis

In order to analyze the characteristics of the sensor signals on the CAN bus a specific data collection was carried out using the same test vehicle. Vehicle sensor data was initialized by driving around on a parking lot for a short period of time. After the system had been initiated the vehicle was parked and remained stationary for about four and a half hour, collecting CAN data. To avoid initial offsets in the accelerometer data, we attempted to

park the vehicle on a horizontally leveled spot. Following signals on the CAN bus were analyzed; YawRateActual, LateralAcceleration, LongAcceleration, VehicleSpeedOverGnd and CAN time (section 6.2). The results from this data collection can be seen in Figure 6.6, 6.7 and 6.8.

VehicleSpeedOverGnd was exactly zero when the vehicle was stationary, and the CAN time was extracted to provide with time information during the collection.

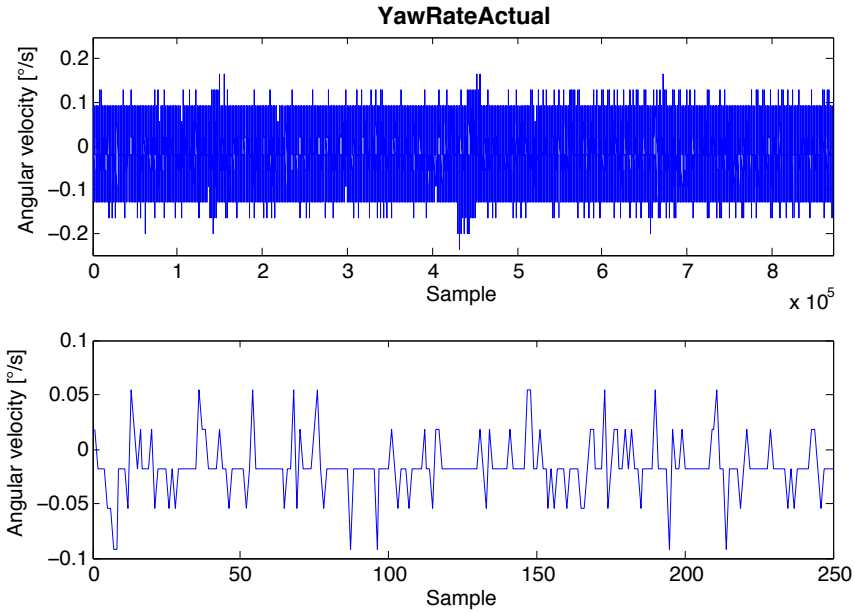


Figure 6.6: CAN bus signal *YawRateActual*. Top: whole stationary data collection. Bottom: a selection of 250 samples.

A plot of *YawRateActual* is depicted in Figure 6.6, where it can be evaluated that the signal is subject to noise and bias offset, but no drift. In conformity with *YawRateActual*, neither *LateralAcceleration* nor *LongAcceleration* was subject to drift during this data collection.

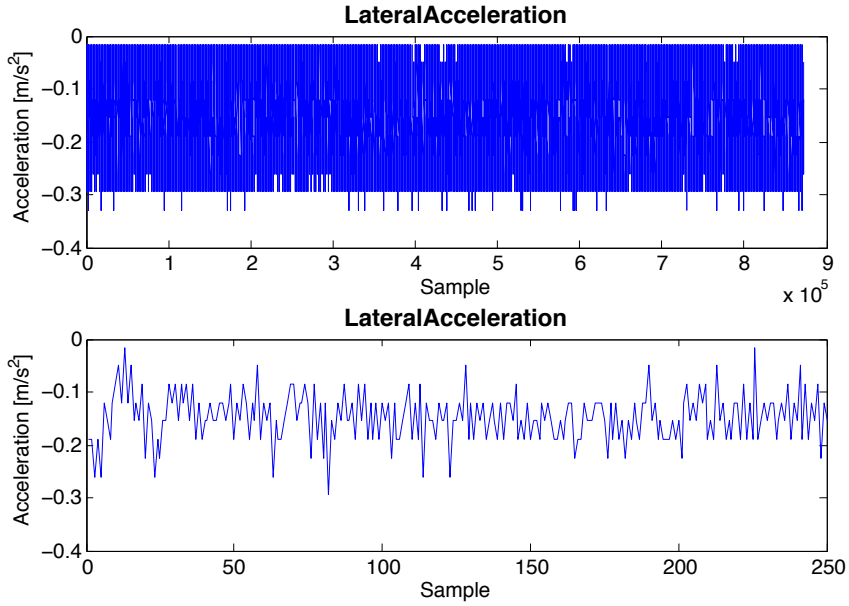


Figure 6.7: CAN bus signal *LateralAcceleration*. Top: whole stationary data collection. Bottom: a selection of 250 samples.

As can be seen in Figure 6.7 and 6.8 the *LateralAcceleration* signal is a lot more noisy than *LongAcceleration*, however the offset of *LongAcceleration* is smaller than that of *LateralAcceleration*. Note the spike in *LongAcceleration*, depicted in Figure 6.8.

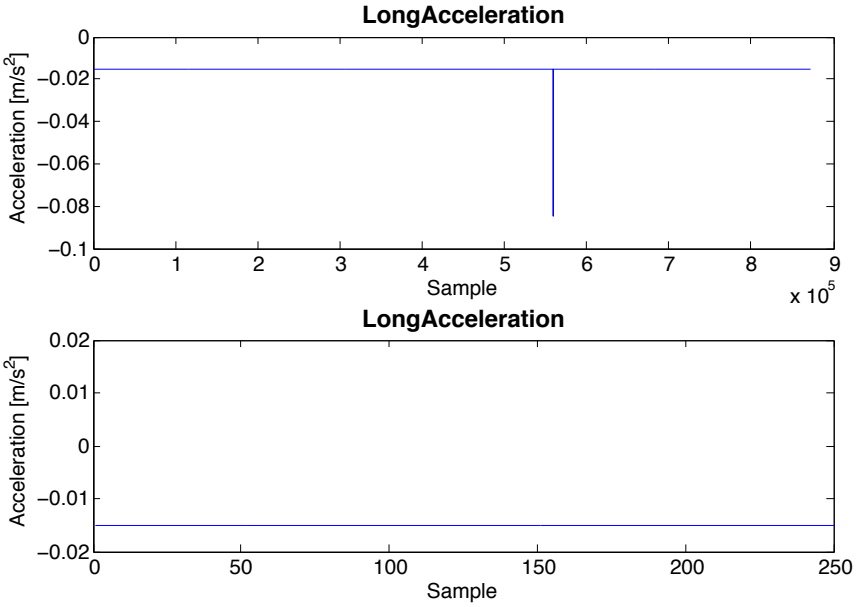


Figure 6.8: CAN bus signal *LongAcceleration*. Top: whole stationary data collection. Bottom: a selection of 250 samples.

By analyzing the signals and extracting their corresponding standard deviation, values for the elements in in the measurement covariance matrix R (Eqn. 7.11 in section 7.1.3) can be selected. By using MATLAB-command *std* to calculate the standard deviation of each signal, following result was achieved:

- *YawRateActual*: $\sigma_{\dot{\varphi}} = \sqrt{Var(\dot{\varphi})} \approx 0.029$,
- *LateralAcceleration*: $\sigma_{a_{lat}} = \sqrt{Var(a_{lat})} \approx 0.046$,
- *LongAcceleration*: $\sigma_{a_{long}} = \sqrt{Var(a_{long})} \approx 7.5 \cdot 10^{-5}$
- *VehicleSpeedOverGnd*: $\sigma_v = \sqrt{Var(v)} = 0$

7 Proposed sensor fusion

According to our study of previous research, and the sensors available in the test vehicle, we created a sensor fusion model based on the extended Kalman filter and measurement inputs from a low-grade GPS receiver, IMU sensors and odometer, to improve absolute position estimation.

The process model of the Kalman filter was based on kinematic relations between states, of which most of them where measurable. Because the low-grade GPS receiver could only provide position, velocity and heading measurements, rather than Doppler measurements, carrier phase and pseudoranges, a loosely coupled integration between GPS and INS was chosen.

7.1 Kalman filter model

In this section we explain the Kalman filter, with focus on states, process and measurement model, and ultimately the Kalman filter parameters.

In an attempt to have both simplicity and functionality, i.e. few states but good performance, following state vector x was formulated (Eqn. 7.8),

$$x = \begin{bmatrix} E\ [m] \\ N\ [m] \\ \varphi\ [rad] \\ v\ [m/s] \\ \dot{\varphi}\ [rad/s] \\ a\ [m/s^2] \end{bmatrix} \tag{7.1}$$

where $a = \dot{v}$ and the direction of a (acceleration) and v (velocity) is according to heading angle φ, see Figure 7.1. States E and N are local coordinates *East* and *North* (for more information see section 7.2.2).

The states were selected through an evaluation of the sensor data available, where directly measured states are easier to estimate, gaining better robustness. For a mathematical overview of the Kalman filter theory, see section 4.1.

In Figure 7.1 a vehicle is depicted in ENU-coordinates (section 7.2.2), where φ, v, a is the heading angle, velocity in heading direction, and acceleration in heading direction, respectively.

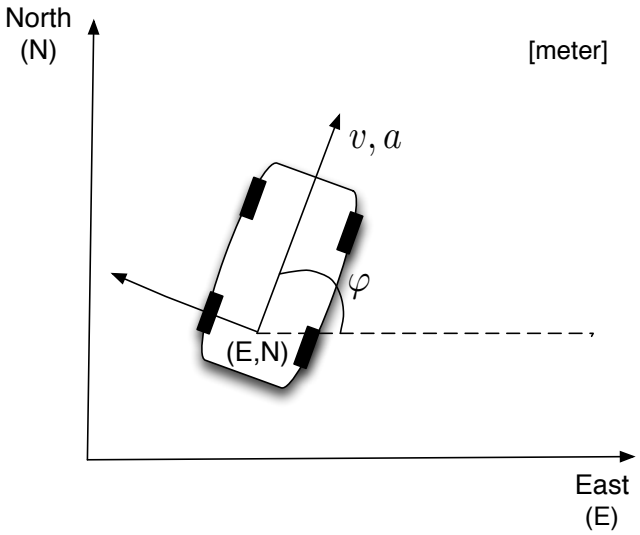


Figure 7.1: Local geodetic coordinate frame (ENU) where "Up" is not depicted. This is the sensor fusion (GPS/INS) coordinate frame.

On-road detection (section 7.1.4) and gyroscope bias compensation (section 7.1.5) are considered as expansions to the default Kalman filter.

7.1.1 Process model

As described in section 4.1 the process model describes the relation between the state estimates from previous time step $k-1$ to current time step k. The process model can be selected in various ways, even though the measurement inputs are the same. In our case we chose a kinematic representation of our states as process model. A kinematic process model describes the kinematic relations between the states x.

Eqn 7.2-7.7 describe the nonlinear kinematic relation between the states x defined in section 7.1, where T is the sampling time. Equations were taken from a licentiate thesis by Lars Danielsson [15].

$$
\begin{aligned}
E_k &= E_{k-1} + cos(\varphi_{k-1})(v_{k-1}T + a_{k-1}T^2/2 + \dot{a}_{k-1}T^3/3) - ... \\
&\quad sin(\varphi_{k-1})(v_{k-1}\dot{\varphi_{k-1}}T^2/2 + (v_{k-1}\ddot{\varphi_{k-1}} + 2a_{k-1}\dot{\varphi}_{k-1})T^3/6 + ... \\
&\quad (\dot{a}_{k-1}\dot{\varphi}_{k-1} + a_{k-1}\ddot{\varphi}_{k-1})T^4/8 + \dot{a}_{k-1}\ddot{\varphi}_{k-1}T^5/20)
\end{aligned} \tag{7.2}
$$

$$
\begin{aligned}
N_k &= N_{k-1} + sin(\varphi_{k-1})(v_{k-1}T + a_{k-1}T^2/2 + \dot{a}_{k-1}T^3/3) + ... \\
&\quad cos(\varphi_{k-1})(v_{k-1}\dot{\varphi_{k-1}}T^2/2 + (v_{k-1}\ddot{\varphi_{k-1}} + 2a_{k-1}\dot{\varphi}_{k-1})T^3/6 + ... \\
&\quad (\dot{a}_{k-1}\dot{\varphi}_{k-1} + a_{k-1}\ddot{\varphi}_{k-1})T^4/8 + \dot{a}_{k-1}\ddot{\varphi}_{k-1}T^5/20)
\end{aligned} \tag{7.3}
$$

$$
\varphi_k = \varphi_{k-1} + sign(v_{k-1})(\dot{\varphi}_{k-1}T + \ddot{\varphi}_{k-1}T^2/2) \tag{7.4}
$$

$$
v_k = v_{k-1} + a_{k-1}T + \dot{a}_{k-1}T^2/2 \tag{7.5}
$$

$$
\dot{\varphi}_k = (\dot{\varphi}_{k-1} + \ddot{\varphi}_{k-1}T) \tag{7.6}
$$

$$
a_k = a_{k-1} + \dot{a}_{k-1}T \tag{7.7}
$$

The nonlinearity depicted in Eqn 7.2 and 7.2 was a reason for us to choose a nonlinear applicable filter, in our case the extended Kalman filter, which requires linearization of the process and measurement model around a working point (section 4.1). The linearized process and measurement model can be found in Appendix C.1. Variables $\ddot{\varphi}$ and $\dot{a}$, i.e. yaw acceleration and jerk, are so-called *noise states*, which were defined as having zero mean, thus omitted in the linearization. However the yaw acceleration and jerk behavior is included in the process noise covariance matrix Q, Eqn. 7.13.

The sample time T is in fact time-varying (T_k), which was calculated by subtracting the CAN time of current HSCAN signal with the previous,

$$
T_k = T_k^{CAN} - T_{k-1}^{CAN}. \tag{7.8}
$$

Results regarding the sample time can be found in section 8.5.

7.1.2 Measurement model

Before describing the measurement model of the Kalman filter, the measurements has to be defined. Our sensor fusion model integrated GPS receiver, IMU and odometer data, where the GPS receiver provided latitudinal and longitudinal position, as well as velocity and heading, of which the two latter are called "true" measurements since they are not effected by slip (Figure 7.2).

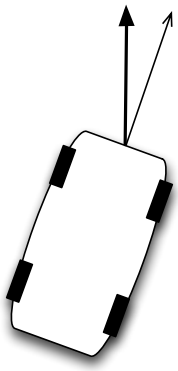


Figure 7.2: Example comparison between true heading (bold arrow) and car direction (thin arrow), these differ when vehicle is subject to slip.

In section 4.1 the measurement model, linearly represented by matrix H, is described as providing the relation between the state estimates and measurements at current time step k. The measurements are depicted in vector z,

$$z = \begin{bmatrix} E\ [m] \\ N\ [m] \\ \varphi\ [rad] \\ v\ [m/s] \\ \dot{\varphi}\ [rad/s] \\ a_{long}\ [m/s^2] \\ a_{lat}\ [m/s^2] \end{bmatrix}, \tag{7.9}$$

where a_{long} and a_{lat} are the lateral and longitudinal accelerations relative the vehicle. Most of the states in x can be directly measured (Eqn. 7.9) and thus has a direct linear relation, with the only exception of the lateral acceleration measurement that has the following nonlinear model (Eqn. 7.10).

$$a_{lat} = \dot{\varphi} \cdot v \tag{7.10}$$

The linear part of the measurement model H is presented in Eqn. C.19 in Appendix C.1.

GNSS and INS provide information on different coordinate systems, and measurement data and measurements z is on different formats and units, thus data pre-processing had to be performed. This data pre-processing is further described in section 7.2.

7.1.3 Filter parameters

When the foundation of the Kalman filter had been made, i.e. process and measurement model, there were some parameters left to define in order to have complete functional Kalman filter;

- measurement covariance matrix R,
- process noise covariance matrix Q,
- initial state estimate x,
- and initial state covariance matrix P.

To simplify the interpretation of the simulation results, default parameters were set and are presented in Eqn. 7.11 - 7.13 below. Some parameters were guesstimated, and others were derived through evaluation of sensor data, provided in section 6.2.2.

$$R = \begin{bmatrix} 25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (10 \cdot \pi/180)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (0.1 \cdot \pi/180)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.11}$$

Measurement variances (Eqn. 7.11) were derived in the following way;

- GPS position measurements, north and east after transformation from latitude and longitude (section 7.2.2), were derived from BU-353 specifications that stated a 5 meter standard deviation at 1σ confidence interval, thus having a variance of $5^2 = 25\ m^2$.
- Heading angle measurement variance was guesstimated by analyzing several plots of the BU-353 trajectory in comparison with the RT-3002, and a standard deviation of 10° was specified.
- Velocity from *vehicleSpeedOverGnd* did not show any drift or bias when logged (section 6.2.2), thus a small standard deviation was set, $0.01\ m$.
- Yaw rate measurements proved to be stable, as was depicted in section 6.2.2, Figure 6.6, therefore a low variance was set, $0.1\ ^\circ/s$.
- Acceleration (longitudinally and lateral relative vehicles nose) measurement variances differed between each other. Longitudinal acceleration measurements where less noisy than lateral (section 6.2.2), thus a variance of 0.01 and 1 was chosen.

Because the Kalman filter was by default initialized with the first measurements provided from the sensors, referred to as optimal initialization in the simulation results section 8, the initial state covariance matrix, Eqn. 7.12, was very similar to the measurement error covariance matrix, Eqn. 7.11.

$$P_{k=1} = \begin{bmatrix} 25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 & 0 & 0 \\ 0 & 0 & (10 \cdot \pi/180)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-4} & 0 & 0 \\ 0 & 0 & 0 & 0 & (0.1 \cdot \pi/180)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.12}$$

The process noise covariance matrix, Eqn. 7.13, address the effect of yaw acceleration and jerk upon the process model, and were evaluated by trial and error upon a series of data sets.

$$Q = \begin{bmatrix} 10(20 \cdot \pi/180)^2 & 0 \\ 0 & 1000 \end{bmatrix} \tag{7.13}$$

Initial state estimate $x_{k=1}$ is by default set equal to the first measurement from each sensor, or alternatively set to zero. If states are set to zero, the initial state covariance

matrix P is altered such that $Var(\varphi) = \pi^2$. The reason for increasing the heading variance in this case is because heading measurements are only updated every second, and the heading state has big influence on the Kalman filter output. The variance is set to π^2 since a standard deviation of π corresponds to an unknown heading (heading may range between 0 and 2π).

7.1.4 On-road detection

In an attempt to improve the filter output estimates, we wanted to have measure of GPS measurement quality. This was done by investigating if the provided position from the BU-353 GPS receiver gave us a point on a road or not. For the selected data segments we manually entered if each position measurement from the BU-353 was off or on the road according to Google Maps. If the position was not on the road, the GPS measurements at that time instance was temporarily given very high measurement error variance, in our case 10^8 (Eqn. 7.11).

7.1.5 Gyroscope bias compensation

As presented in section 6.2.2 the yaw rate signal provided on HSCAN is subjected to bias offset but no drift, when the vehicle was stationary (non moving). However as seen in our own simulations and in [29], sensor drift will increase when the sensor is subject to dynamics, e.g. turns.

As an attempt to compensate for the potential drift and bias, we added an extra state to the Kalman filter process model (section 7.1.1). This state is not measurable, and will thus only be estimated by the Kalman filter at every cycle. Eqn. 7.14 below describes the process model of this state, which will hereafter be referred to as "gyro bias" ($\dot{\varphi}^{bias}$).

$$\dot{\varphi}_k^{bias} = \dot{\varphi}_{k-1}^{bias} + w_{drift} \tag{7.14}$$

Where w_{drift} is represented by adding another process noise in the process noise covariance matrix Q (Eqn. 7.13),

$$Q = \begin{bmatrix} 10(20 \cdot \pi/180)^2 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & (10^{-4} \cdot \pi/180)^2 \end{bmatrix} \tag{7.15}$$

This also alters the yaw-rate process model $\dot{\varphi}$ as follows,

$$\dot{\varphi}_k = \dot{\varphi}_{k-1} + \ddot{\varphi}_{k-1} T + \dot{\varphi}_{k-1}^{bias}. \tag{7.16}$$

The augmented state space resulted in the following variances; $Var(\dot{\varphi}^{bias}) = (.0001\pi/180)^2$ in $P_{k=1}$ (Eqn. 7.12) and Q (Eqn. 7.13).

Other augmented matrices and vectors when using extra gyro bias state can be found in Appendix C.2.

7.2 Data pre-processing

When the sensor fusion model had been made, in accordance with the output from the data acquisition, the data had to be transformed into a frame that was compatible with the process and measurement model of the Kalman filter. Different measurements had to be related to each other in a common frame in order to be fused. In accordance with the measurement model of the Kalman filter, described in section 7.1.2, following measurement data was used;

- global longitudinal positon,
- global lateral positon,
- heading angle,
- velocity,
- yaw rate,
- longitudinal acceleration (relative vehicle),
- and lateral acceleration (relative vehicle),

where following measurements need conversion to fit into a common local navigation frame depicted in Figure 7.1; global longitudinal and lateral position, and heading angle.

Latitude and longitude position is on a global frame, thus the need for conversion to a local navigation frame. Heading angle given from the GPS receiver require manipulation such as conversion from compass to unit circle frame to enable sine and cosine transformation, and there is also the problem of the heading angle signal jumping between 0 and 360 degrees.

Apart from these conversions and transformations, data synchronization had to be made since measurements arrived at different time instances. Following subsections will describe the main pre-processing we performed to enable sensor fusion on a common framework.

7.2.1 Heading angle manipulation

There were three manipulation steps of the heading angle applied in this project. The first step concerned removal of heading angle measurements when velocity was below 5 km/h, due to increased noisy characteristics. Secondly the heading angle was converted from compass to unit circle format, and thirdly, jumps created in the signal when the heading angle moved between 0 and 360 degrees were removed. The latter two will be explained in further detail within this section.

Heading data extracted from a GPS receiver must be pre processed in order to work with sine and cosine functions, and to match according to our process model. This was because the heading from the GPS receiver was on compass format, and sine and cosine functions require data to be on unit circle format (Figure 7.3). Heading angle was measured in degrees and was first transformed to the more conventional format of radians (Eqn. 7.17).

$$heading_{rad}^{compass} = heading_{deg}^{compass} \cdot \frac{\pi}{180} \tag{7.17}$$

By laterally transposing the compass and rotate it 90 degrees ($\frac{\pi}{2}$) according to Eqn. 7.18, as can be derived from Figure 7.3, the heading was converted to unit circle format.

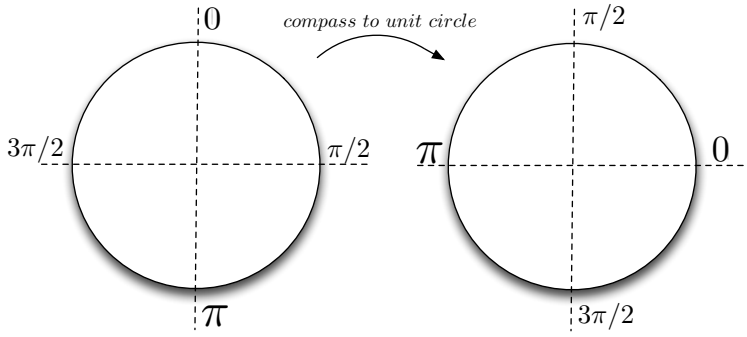


Figure 7.3: Transformation from compass (left) to unit circle (right) format.

$$heading_{rad}^{unitCircle} = -heading_{rad}^{compass} + \frac{\pi}{2} \tag{7.18}$$

The last step in the pre processing of the heading data was to remove jumps that occurs around a critical area of 2π, which is due to the compass format output ranging from *0 to 360 degrees*. If heading is increasing above this range the heading jumps from 2π to zero and if he heading instead is decreasing below this range the measured heading jumps from zero to 2π. When using sine and cosine functions, multiples of 2π does not matter. But since we are to estimate the heading angle using the Kalman filter, which does not consider 2π and 0 to be equal, jumps between these values will deteriorate the estimate.

These jumps are removed by counting the number of revolutions n, which changes when the value jumps either from 0 to 2π or vice versa. It is assumed that if the heading changes more than π in between two samples it has rotated one revolution according to the difference between the two samples, thus changing the variable n with $+1$ or -1 accordingly (Eqn. 7.19). When all the jumps in the signal has been detected, variable n can be used to correct the signal by removing these jumps.

$$heading_{rad,Continuous}^{unitCircle} = heading_{rad}^{unitCircle} + 2\pi n \tag{7.19}$$

MATLAB code for this transformation can be found in Appendix B.

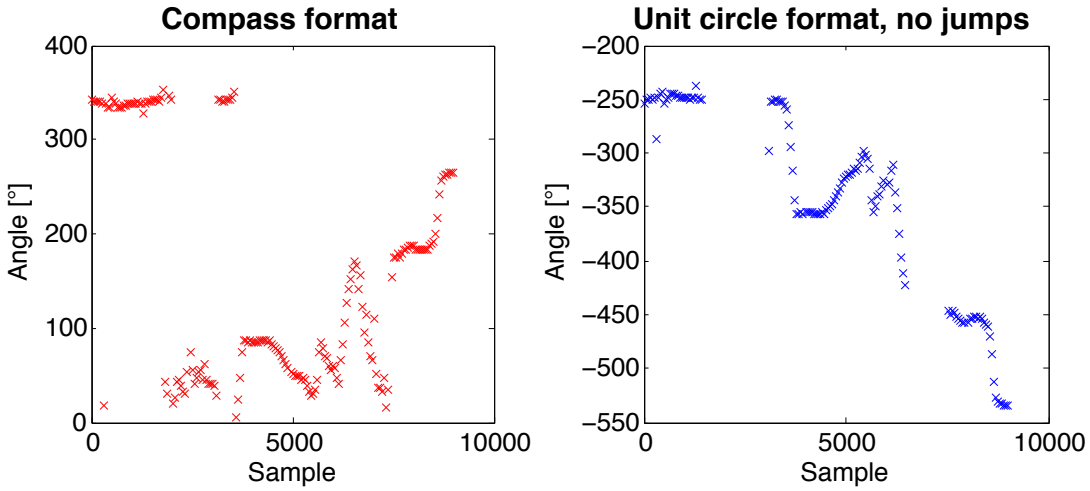


Figure 7.4: Compass format (left), and the resulting unit circle format where jumps are removed (right). Data collected at Aschebergsgatan, Gothenburg.

Figure 7.4 depicts the heading angle on compass format, explaining the problem with the jumps in the signal, and also the resulting heading output after manipulations. Note that some measurements are removed where the vehicle had a velocity of less than 5 km/h, as seen in the right-most figure of Figure 7.4. Compare Figure 7.4 with the vehicle position trajectory in the simulation results section 8.1 to get a better understanding of the movement of the vehicle.

7.2.2 Coordinate transformations

When handling different kinds of sensors, such as GPS and IMU that give global and body-related measurements, respectively, it is important to have the data on the same frame to enable fusion, in this case a local geodetic frame having data relative the coordinates east, north and up (ENU). Position data given from the BU-353 GPS-receiver, longitude,

latitude and altitude (LLA), are on a global coordinate system. However, position data from an INS-system can be described in the local geodetic frame (ENU) [11], [18].

Because of the difference in the coordinate systems of GPS and INS, a transformation is required to render sensor fusion. GPS positions can be presented in different formats such as *decimal degrees* and *degrees, minutes and seconds*. In order to provide fusion between GPS and INS a two-step transformation, from LLA to earth-centered earth-fixed (ECEF), and ECEF to ENU, was required [18].

Transformation from global to navigation frame Transformation between global GPS coordinate frame (LLA) to navigational INS coordinate frame (ENU) is a two-step solution, where LLA transforms to ECEF coordinates, and the ECEF coordinates are transformed to ENU coordinates. First of all the LLA data has to be on the decimal degree format. Then to transform from LLA to ECEF following equations are needed [18], [12], [68],

$$x = (R_n + h)\cos\varphi\cos\lambda \tag{7.20a}$$
$$y = (R_n + h)\cos\varphi\sin\lambda \tag{7.20b}$$
$$z = (R_n[1 - e^2] + h)\sin\varphi \tag{7.20c}$$

where λ, φ and h is the geodetic longitude, latitude and ellipsoid height, respectively, and e^2 is calculated using WGS 84 constants [53],[67]; the semi-major axis $a = 6378137.0$ meters, and the inverse flattening $1/f = 298.25722356$ (Eqn. 7.21).

$$e^2 = f(2 - f) \tag{7.21}$$

R_n describes the ellipsoid curvature [12],

$$R_n = \frac{a}{\sqrt{1 - e^2\sin^2\varphi}}. \tag{7.22}$$

In the WSG 84 standard, used in the GPS, Earth is approximated as an ellipsoid. In this transformation the height h is defined as the ellipsoid height, which is best explained in Figure 7.5 below.

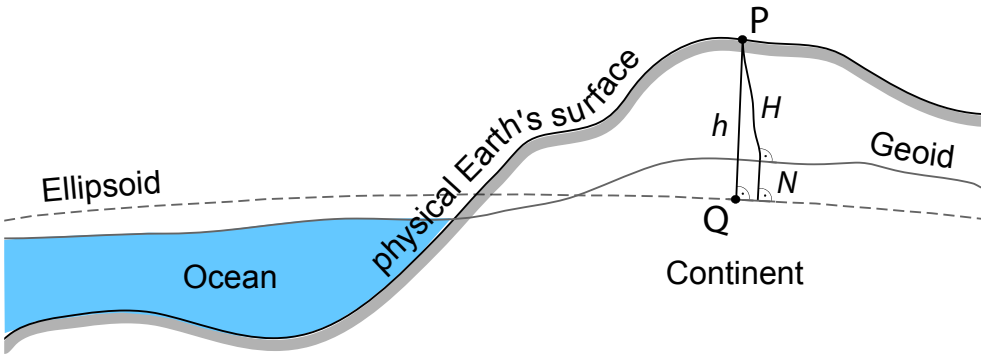


Figure 7.5: Description of ellipsoidal height h relative to physical Earth's surface, geoid, ocean level and ellipsoid approximation. Courtesy of Michael Kuhn, Curtin University of Technology.

It is not possible to transform directly from ECEF to ENU coordinates without introducing a reference point $A = [x_{ref}, y_{ref}, z_{ref}]^T$, which will define the origin of the local

geodetic frame [68]. Using the new ECEF coordinates x,y and z derived from Eqn. 7.20a-c, relative distances to the reference point A can be calculated as follows:

$$[\delta X, \delta Y, \delta Z]^T = [x, y, z]^T - A = [(x - x_{ref}), (y - y_{ref}), (z - z_{ref})]^T \tag{7.23}$$

The resulting relative coordinates $[\delta X, \delta Y, \delta Z]^T$ calculated in Eqn. 7.23 above will be used in the transformation to ENU coordinates,

$$[E, N, U]^T = R[\delta X, \delta Y, \delta Z]^T \tag{7.24}$$

where R in Eqn. 7.24 is the transformation matrix defined as,

$$R = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\varphi\cos\lambda & sin\varphi\sin\lambda & \cos\varphi \\ \cos\varphi\cos\lambda & \cos\varphi\sin\lambda & \sin\varphi \end{bmatrix}. \tag{7.25}$$

When the position retrieved from the GPS receiver has been transformed to the local ENU-coordinate system, sensor fusion can be applied. In case the ENU-coordinates are to be transformed back to global coordinates, which, e.g., is needed in order to plot position outputs from the sensor fusion model on top of maps such as *Google Maps*.

Transformation back from navigation to global frame Transforming from ENU back to LLA-coordinates might seem easy, just transform from ENU back to ECEF and then again from ECEF to LLA. But since R_n is required to calculate the height h and the geodetic latitude φ is required to calculate R_n, there is no closed form solution going from ECEF to LLA-coordinates [12]. Hence, an iterative solution is suggested where only a few iterations is required to get centimeter level accuracy. Transforming from ENU to ECEF is however more or less straight forward, using Eqns. 7.23, 7.24 and 7.25.

$$[x, y, z]^T = R^T[E, N, U]^T + A \tag{7.26}$$

Using the result given through Eqn. 7.26 the longitudinal coordinate λ can be directly transformed according to,

$$\lambda = \arctan(y/x) \tag{7.27}$$

Latitude and altitude coordinates can be derived through an iterative process, using initializing value φ_{temp},

$$\varphi_{temp} = \arctan(p/z) \tag{7.28}$$

where

$$p = \sqrt{x^2 + y^2}. \tag{7.29}$$

The initial latitude φ_{temp} from Eqn. 7.28 and p from Eqn. 7.29 are used to iteratively compute a better approximation of the true global latitude coordinate φ through the following equations [12];

$$R_n = \frac{a}{\sqrt{1 - e^2 \sin^2(\varphi_{temp})}} \tag{7.30a}$$

$$h = \frac{p}{\cos(\varphi_{temp})} - R_n \tag{7.30b}$$

$$\varphi_{temp} = \arctan\left(\frac{z}{p}\left[1 - e^2 \frac{R_n}{R_n + h}\right]^{-1}\right). \tag{7.30c}$$

After a few loops through Eqns. 7.30a-c the value of φ_{temp} can be assigned as the "true" latitude coordinate φ, and the ellipsoidal height h is then calculated as follows,

$$h = \frac{p}{\cos\varphi} - R_n. \tag{7.31}$$

It is important to note that processing these equations, e.g in MATLAB, results in answers in radians, while LLA-coordinates are usually expressed in degrees. Since the transformation from ECEF to LLA does not have a closed form solution, a lossless conversion is not possible. It is also well-known that iterative calculations are in general considered computationally demanding [18], therefore the number of iterations should be kept as low as possible.

7.2.3 Data synchronization

To perform data synchronization between the three main data sources; HSCAN, RT-3002 and BU-353, there are some things that needs to be taken into consideration.

Both the vehicle sensors and RT-3002 provided their data over two separate CAN busses using the same clock. Since the transmission of signals on CAN may vary depending on priorities in the network, it is difficult to estimate when sensor signals will be updated [16], but it is even more difficult to estimate the time difference between measurement and transmission on the bus. To be able to synchronize IMU and odometer signals with a varying update rate, with RT-3002 and BU-353 that has a different but fixed update rate, CAN times from the RT-3002 was matched with CAN times on the HSCAN, and the BU-353 was matched to the RT-3002 data through comparison of the UTC/GPS times.

Different GPS receivers may use different time standards. The BU-353 is using UTC that utilizes leap seconds, while RT-3002 uses GPS time that has a linear time scale. Leap second is a time correction being made, which adds one second at approximately every 18'th month to the UTC system [22] (see also section 2.1). The difference between these two standards is that GPS time is 15 seconds behind UTC (in April 2012) [2], which had to be encountered for when synchronizing RT-3002 and BU-353 data.

By selecting a data sequence provided from the CAN, which already had IMU, odometer and RT-3002 data synchronized, and extracting the GPS time of the first and last sample of that data sequence, corresponding BU-353 GPS data was extracted, taking the 15 leap seconds under consideration.

The first and last *GPS-time* stamp of a selected three-minute CAN and BU-353 data sets can be written on the following form; *hhmmss.xxx*, where *hh* are hours, *mm* minutes, *ss* seconds, and *xxx* milliseconds. Since the CAN bus signals used are updated at approximately 50 Hz, and the BU-353 is updated at 1 Hz, the data has to be augmented to a common frequency of 100 Hz (Figure 7.6).

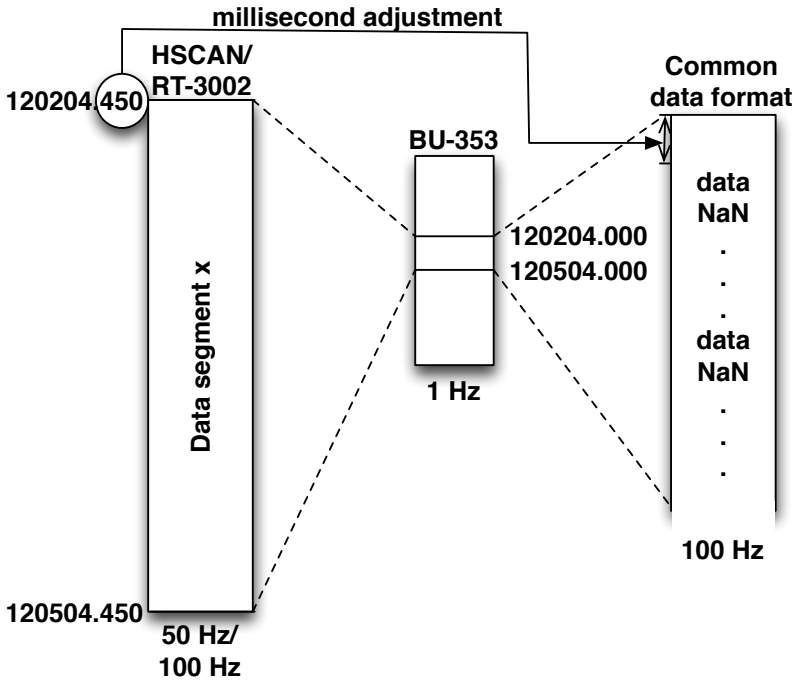


Figure 7.6: Synchronization to make BU-353 data (1 Hz) and HSCAN (50 Hz) on same format (100 Hz).

HSCAN data had to be shifted to match the indexes of the 100 Hz BU-353 data (millisecond adjustment in Figure 7.6).

RT-3002 data overlap As mentioned the system that logged HSCAN and RT-3002 data saved the data in three-minute segments. Combining two adjacent segments would thus result in a six-minute segment, however RT-3002 data will be overlapped, but HSCAN-signals and thus also CAN-time of the RT-3002 will not be overlapped. Therefore merging several RT-3002 data segments without taken this into consideration will result in a jump in the data, see Figure 7.7.

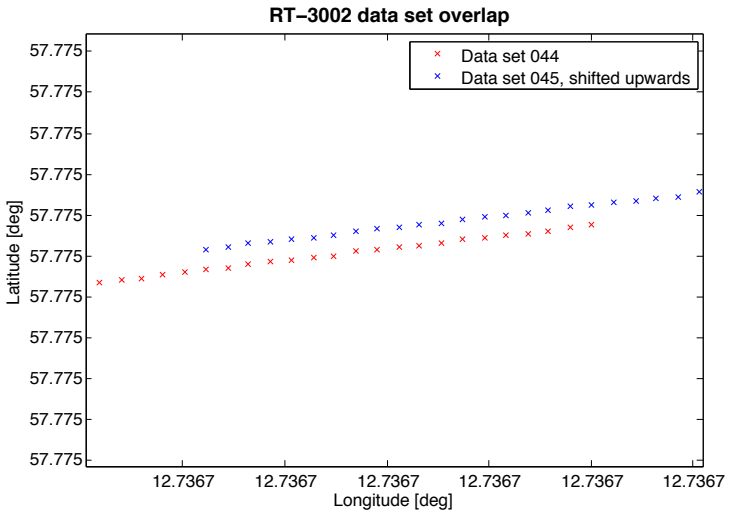


Figure 7.7: Position measurement overlap between two adjacent data segments, 044 and 045, the 045 data is shifted upwards in order to visualize the overlap.

Analysis of several adjacent data segments showed that the overlap varied between 170 and 190 ms. In our case where the RT-3002 position data will only function as a reference rather than input to the Kalman filter, and since we only consider the data segments separately, this overlap can be neglected.

However there is no overlap when comparing yaw rate data between RT-3002 and HSCAN, or when comparing heading output from two adjacent RT-3002 data segments, thus it is reasonable to assume that overlaps only occur for position estimates.

7.3 Simulation

In order to test our sensor fusion model without having to implement it into a vehicle, simulations were made. Performing simulations is an easy way to test and verify code, where errors and problems with functionality can be evaluated and corrected. Through simulations performed in MATLAB we have tested all the algorithms, such as the Kalman filter, but also coordinate transformations, heading conversions, and other parts of the code.

A summary of the code that was written is presented in the flowchart in Figure 7.8 below.

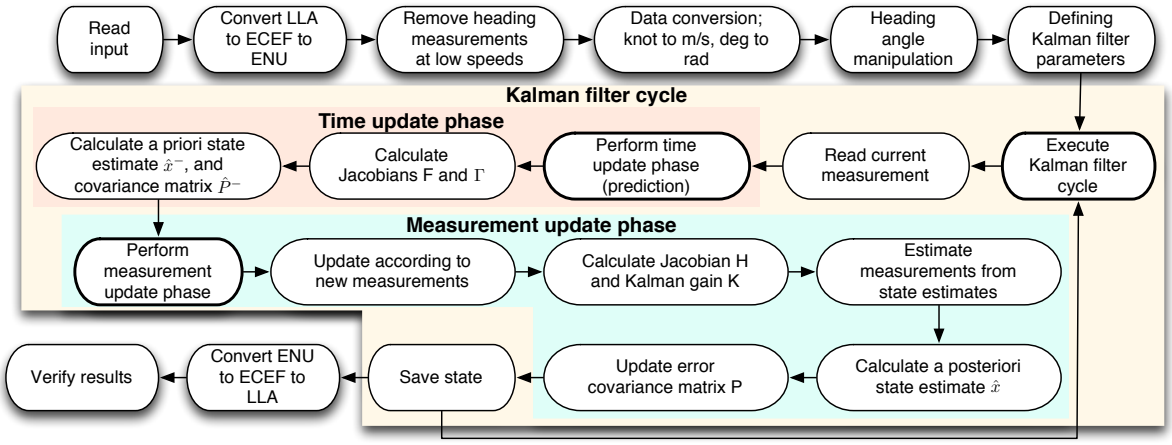


Figure 7.8: Summarized flowchart of our simulation process.

The simulation process contains four major segments; initialization, pre-processing, Kalman filter cycle execution, and verification of results through plots. In Figure 7.8 the Kalman filter cycle and its two main phases, time update and measurement update, are depicted.

7.3.1 Error ellipses

To provide information regarding the uncertainty of the Kalman filter estimates, evaluation of the state estimate covariance matrix P can be made (Figure 4.1). By evaluating the position estimate covariance matrix C,

$$C = \begin{bmatrix} Var(E) & cov(E,N) \\ cov(N,E) & Var(N) \end{bmatrix}, \tag{7.32}$$

where the diagonal elements corresponds to the variance of the positions in east and north direction, and the off-diagonal elements are the cross-covariance (Eqn. 7.32), error ellipses can be plotted to illustrate the possible error according to the Kalman filter. Examples of error ellipses are presented in Figure 8.6-8.7 in section 8.1, simulation results.

Position is given in east and north, thus 1σ in each direction corresponds to ~ 39 % confidence interval for two dimensions (two degrees of freedom) [59], [31], [58]. To acquire 95 % confidence interval $\sim 2\sigma$ for two dimensions, the error ellipse axes are multiplied by a factor of 2.447 [31], [58]. In the same sense, 50 % confidence interval is acquired through multiplication by a factor 1.1773 [58]. MATLAB-code for calculating and plotting error ellipses is given in Appendix A.

8 Simulation results

Data was logged in different scenarios as specified in section 6.2.1, namely city, forest, highway and test track (Hällered).

To validate various aspects of the filter and its performance, results rom the different scenarios are presented. The selection of scenarios were based on environments that would incorporate as much dynamics and aspects of interest regarding absolute position estimation.

In order to simplify the interpretation of the simulation results, default settings such as parameters and extra functions were specified (section 7.1). This is done in order to avoid repeating the values of most parameters and other default settings for every result, instead deviations from these default settings will be presented together with the corresponding result. Note that the RT-3002 is only considered as a position reference at the test track, where it can acquire corrections from a base station, however it is overall assumed to provide a more accurate absolute position than the BU-353.

8.1 City

This scenario represents a typical city environment with large buildings close to the road, and where the vehicle is subject to several tops and operates at a relatively low velocity. In Figure 8.1 the filter is used with default parameters and the additional OnRoad function enabled as well as the extra gyro bias state.

Note the jump in the green trajectory in the down left quadrant, and the jump in the turquoise trajectory in the upper right quadrant.

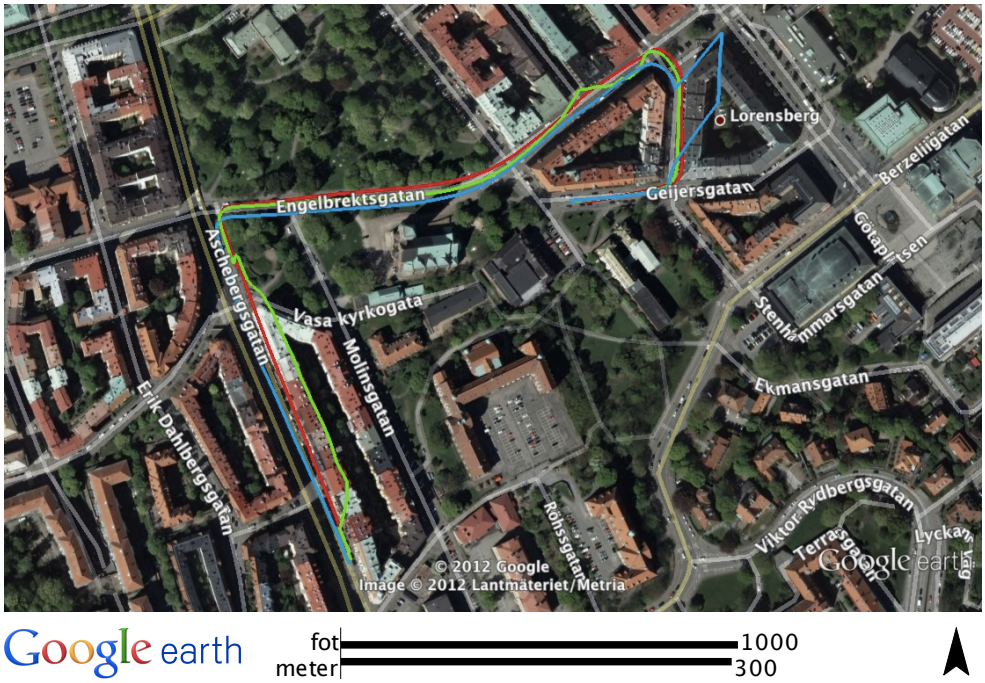


Figure 8.1: Overview of data collection scenario; from Achebergsgatan and onto Engelbrektsgatan in central Gothenburg. Turquoise, green and red represents the measurements from RT-3002, BU-353 and our Kalman filter output, respectively.

Figure 8.2 show the comparison between when extra gyro bias state is added and when its not used. There is no noticeable difference at the start of the run but after the turn in the top left corner a deviation between the two trajectories begins, a deviation that can be seen throughout the rest of the run. Overall the filter position output with the added gyro bias state is closer to the RT-3002 position trajectory.

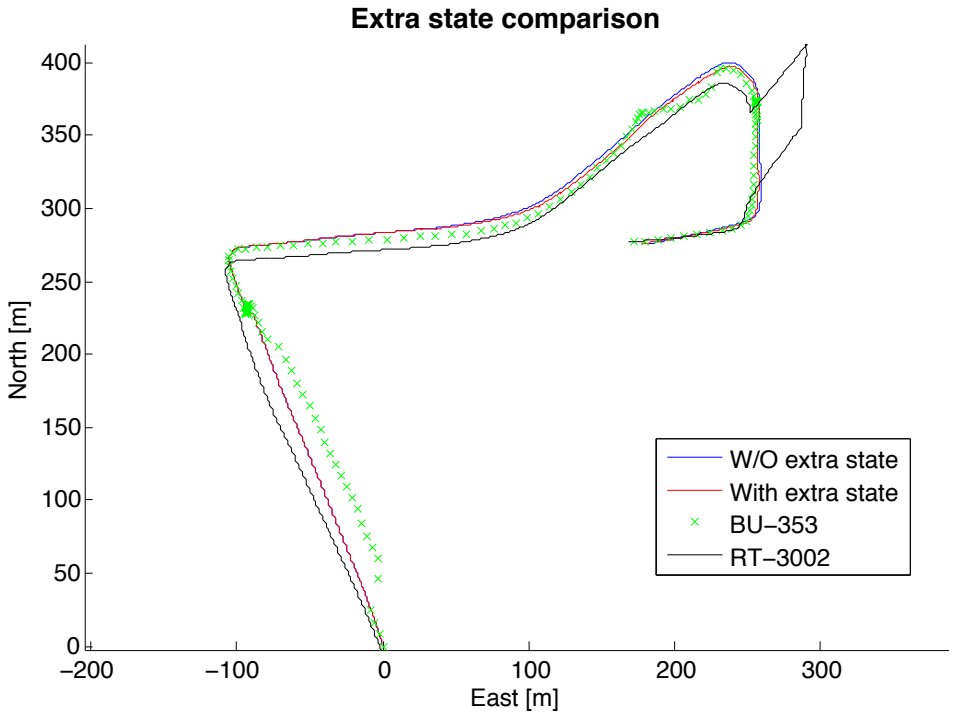


Figure 8.2: Comparison between using the extra gyro bias state in the Kalman filter, and not.

Figure 8.3 shows a close-up view of the top right part of Figure 8.2. In this figure the differences in position estimate between using extra bias state, or not is more clear.

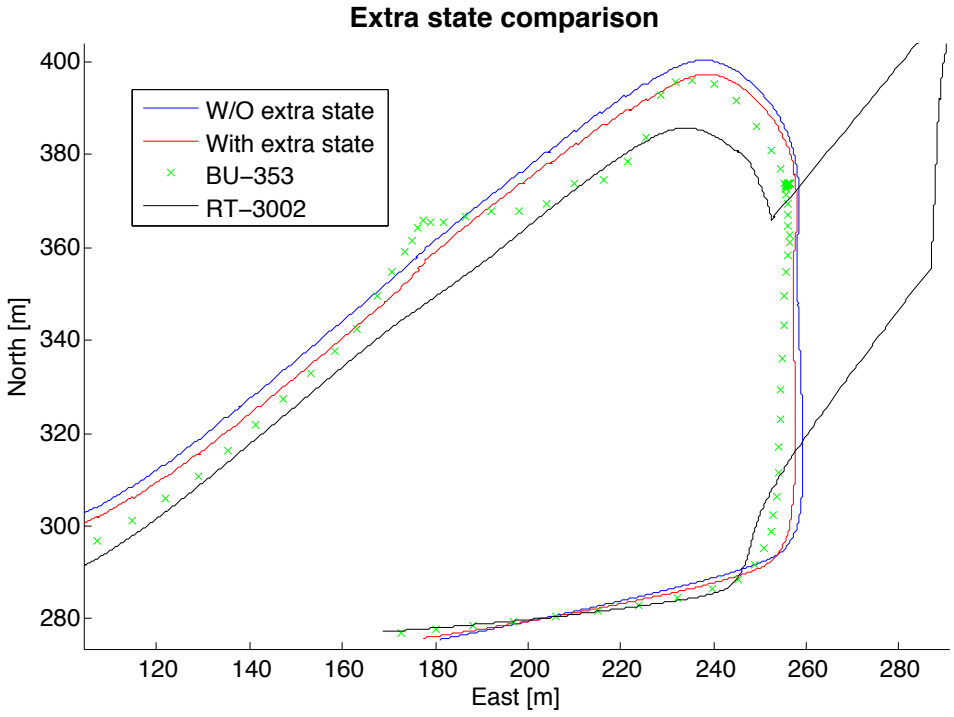


Figure 8.3: Close-up view of the top right quadrant of Figure 8.2.

The OnRoad function alters the filter parameters at two occasions in Figure 8.4; between coordinates $(E, N) \approx (-10, 40)$ and $(E, N) \approx (-100, 230)$, and also for a few measurement samples near $(E, N) \approx (180, 360)$.

As can be seen in Figure 8.4 the filter output is closer to the RT-3002 position estimates when the OnRoad function is enabled. When this function is not used the filter output trajectory follows the BU-353 measurements off the road and into nearby buildings. In the other occasion when OnRoad is enabled there is less change in the filter output compared to BU-353 measurements.

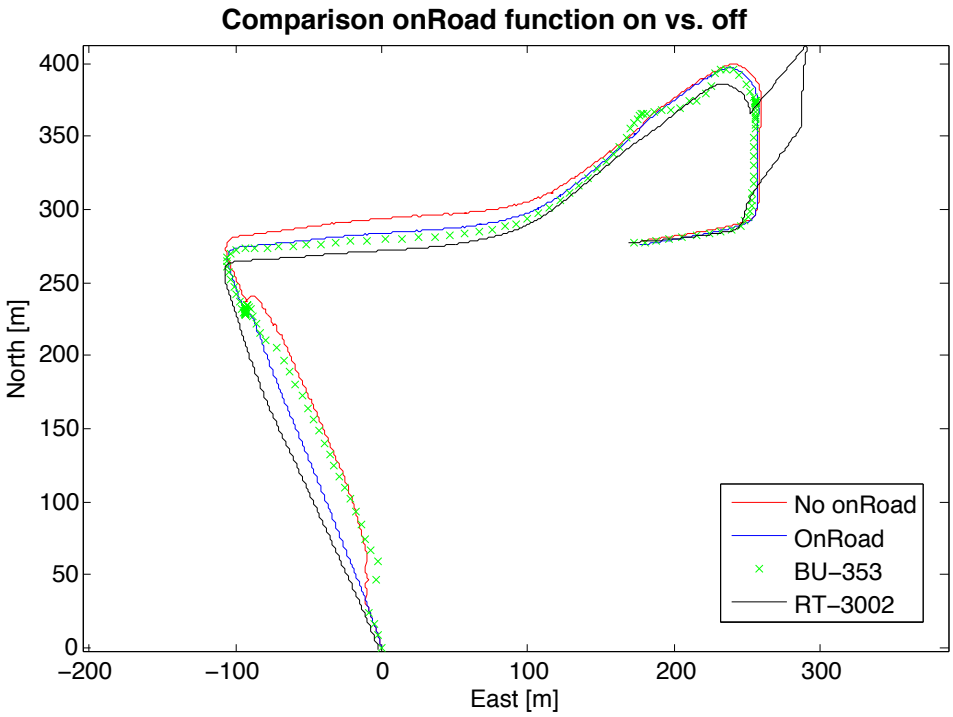


Figure 8.4: Comparison of having onRoad function enabled or disabled in an urban environment.

In Figure 8.5 the filter is either initialized with measurements, referred to as "optimal initialization", or with zeros (section 7.1.3). Difference is noticed in the beginning of the run as the trajectory without optimal initialization is subject to a jump in the position estimate output.

Filter output performance when using optimal initialization is closer to RT-3002 position estimates when the OnRoad function alters the filter parameters (between $(E, N) \approx (-10, 40)$ and $(E, N) \approx (-100, 230)$). Notice that after sufficient amount of measurement samples there is no noticeable difference in performance if optimal initialization is used or not.

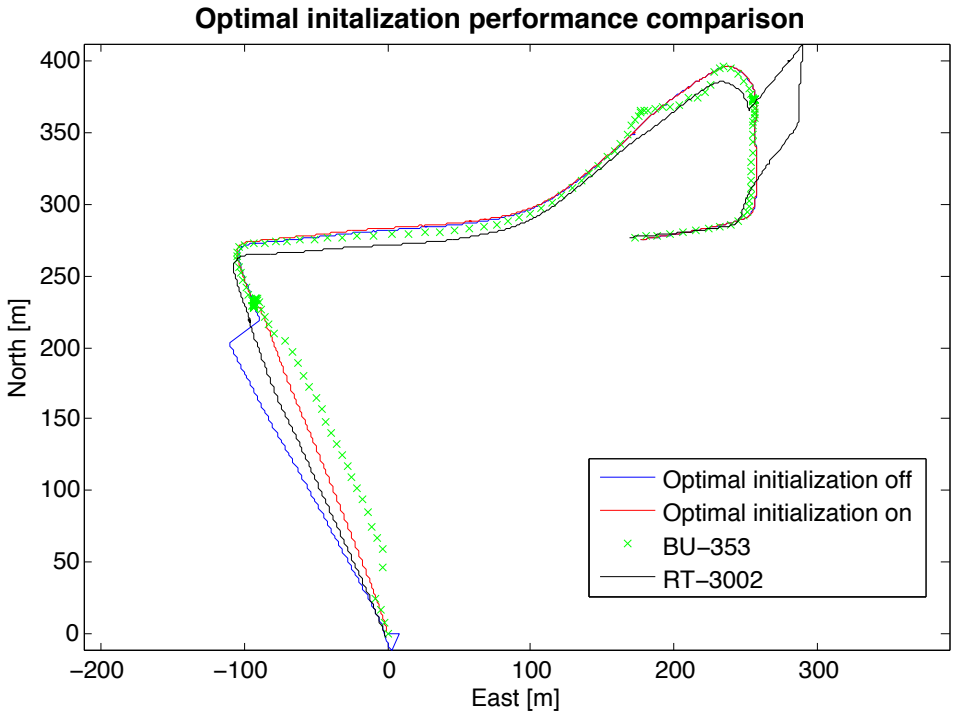


Figure 8.5: Comparison between using optimal initialization and not, OnRoad function enabled.

Figure 8.6 presents error ellipses (section 7.3.1) at a confidence interval of approximately 50%. Position covariance error increases when no absolute position measurements are provided by the BU-353, which happens when the OnRoad function is enabled. The covariance error is less for position estimates in longitudinal direction than lateral, relative vehicle. This is due to more states and measurements acting in longitudinal than lateral direction relative the vehicle.

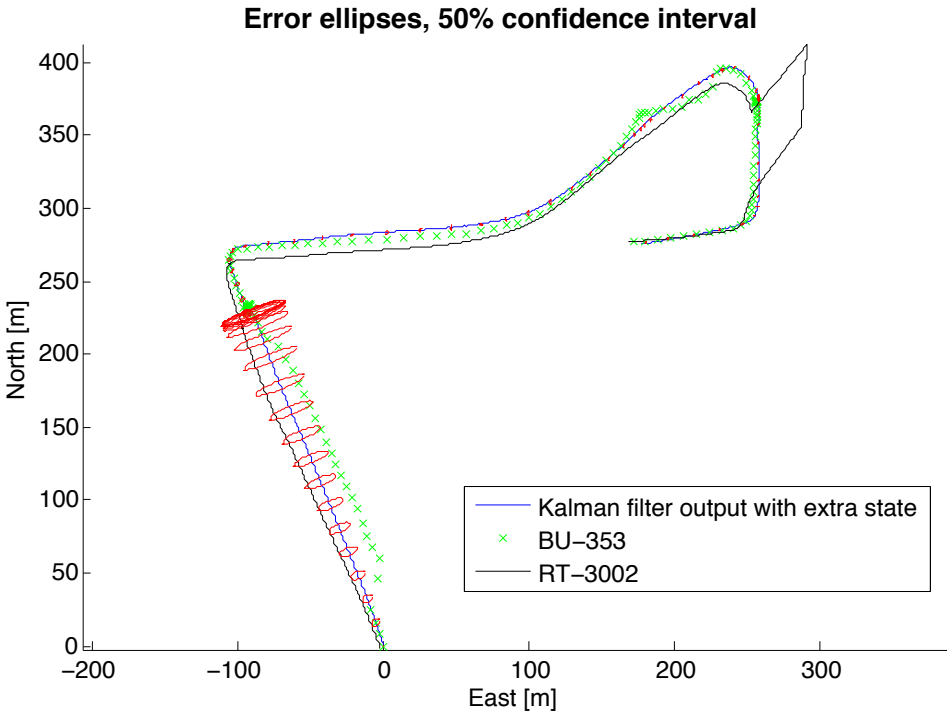


Figure 8.6: Plot showing error ellipses on top of Kalman output, based on position state covariance matrices at every time instance. $\sim 50\%$ confidence interval (2D).

Figure 8.7 presents the same trajectory as in Figure 8.6, but with higher confidence interval.

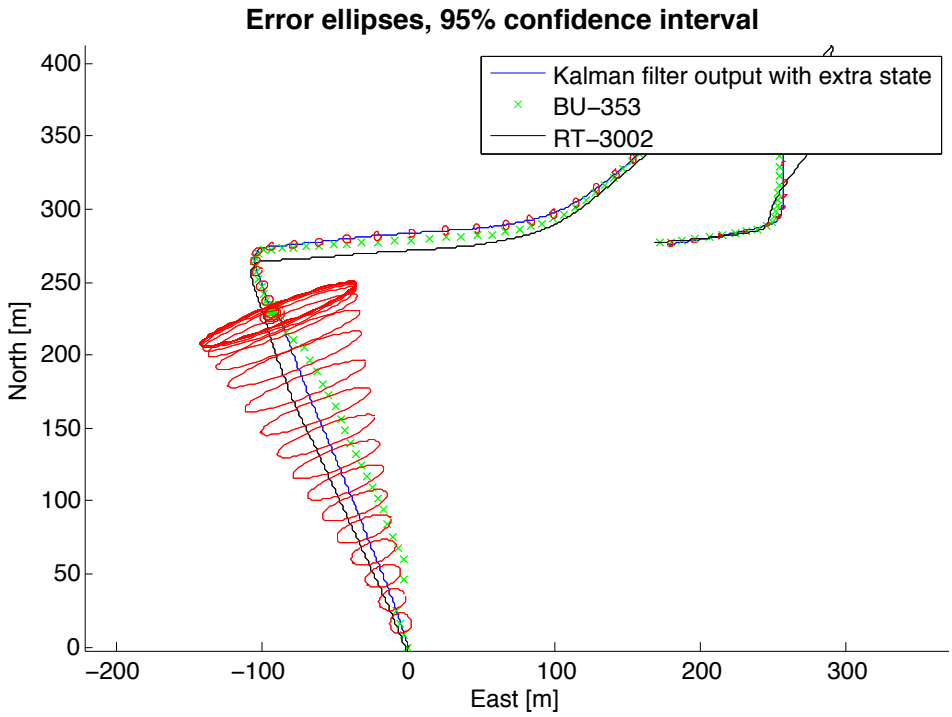


Figure 8.7: Plot showing error ellipses on top of Kalman output, based on position state covariance matrices at every time instance. 95% confidence interval ($\sim 2\sigma$ 2D).

Error ellipses at 95% confidence interval can make reading of plots more difficult.

8.2 Forest road

This scenario takes place on a typical forest road, which involves many slopes and turns, but also a dense forest that may distort GPS receiver measurements. The road is also banked in several places. In Figure 8.9 an overview of the scenario is presented. Noticeable for this scenario is that we are near the test track at Hällered, which can effect the performance of RT-3002 since it may use differential corrections (section 3.1.2), this is however unknown.

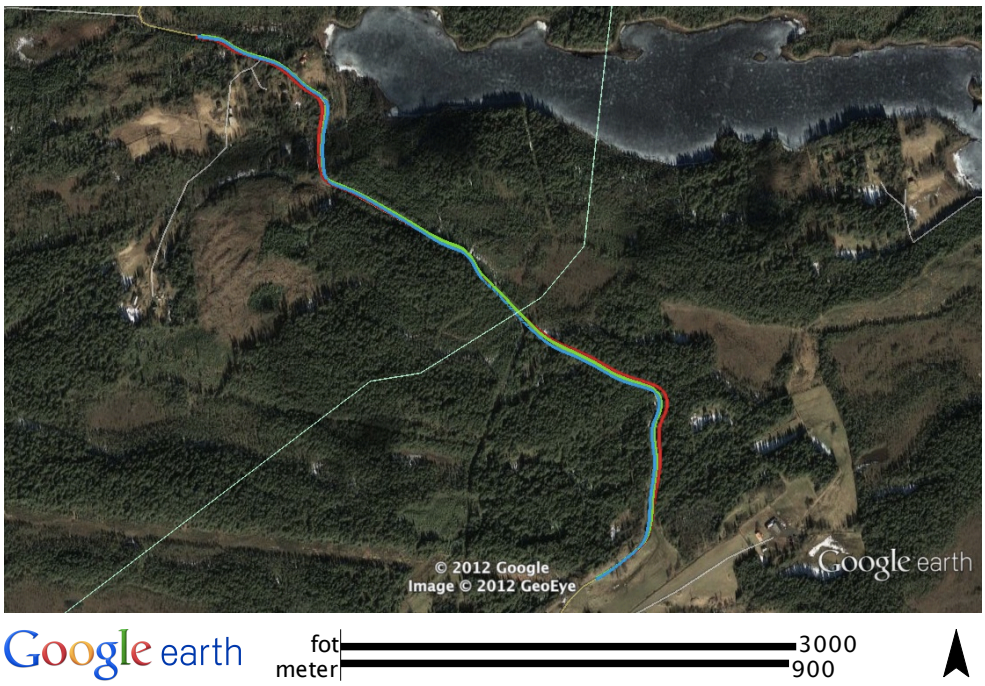


Figure 8.8: Overview of data collection scenario; Forest road near Hällered test track. Turquoise, green and red represents the measurements from RT-3002, BU-353 and our Kalman filter output, respectively.

In Figure 8.9 a close-up view of the trajectories subject to comparison is shown. During this part of the trajectory Google Maps indicates that the GPS receiver unit is mostly placed outside of the road, therefore the OnRoad function can be enabled. The figure shows that the filter output trajectory is closer to RT-3002 outputs when the OnRoad function is disabled, as opposed to the results in Figure 8.4. There is no significant difference between changing the variance of the accelerometer measurements.

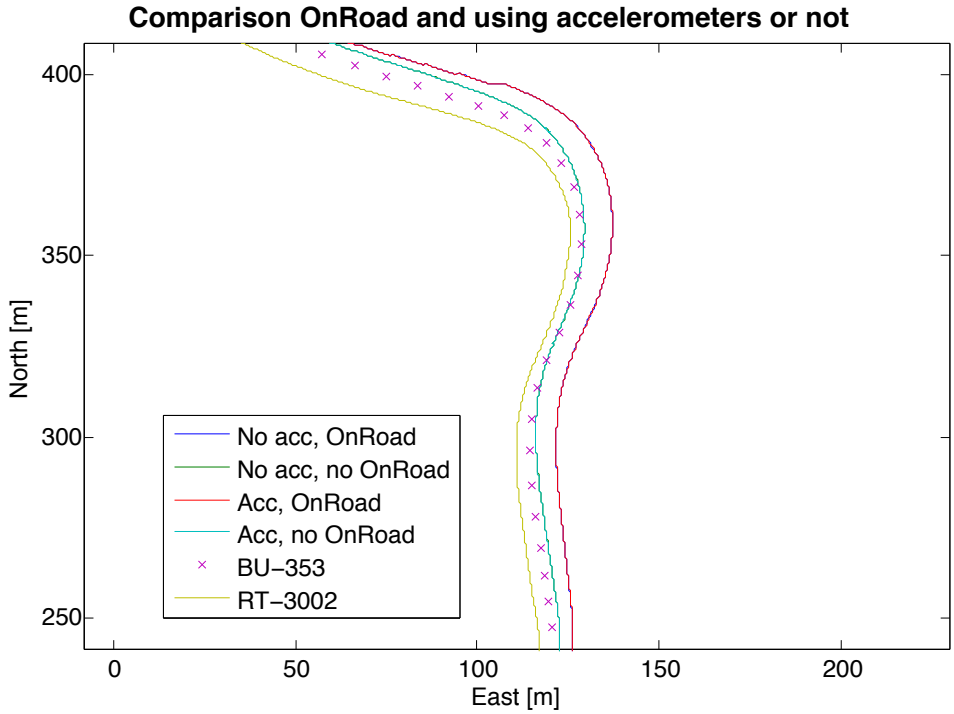


Figure 8.9: Comparison of having onRoad function enabled or disabled, and using high or low (default) variance on acceleration measurements.

Note that all trajectories in Figure 8.9 incorporates the extra gyro bias state.

8.3 Highway

This trajectory was logged on a highway on Hisingen, and stretches from Lundbytunneln in the west to Tingstadstunneln in the east. It represents a relatively wide open area where GPS receivers should provide position outputs with accuracy close to its specifications [47].

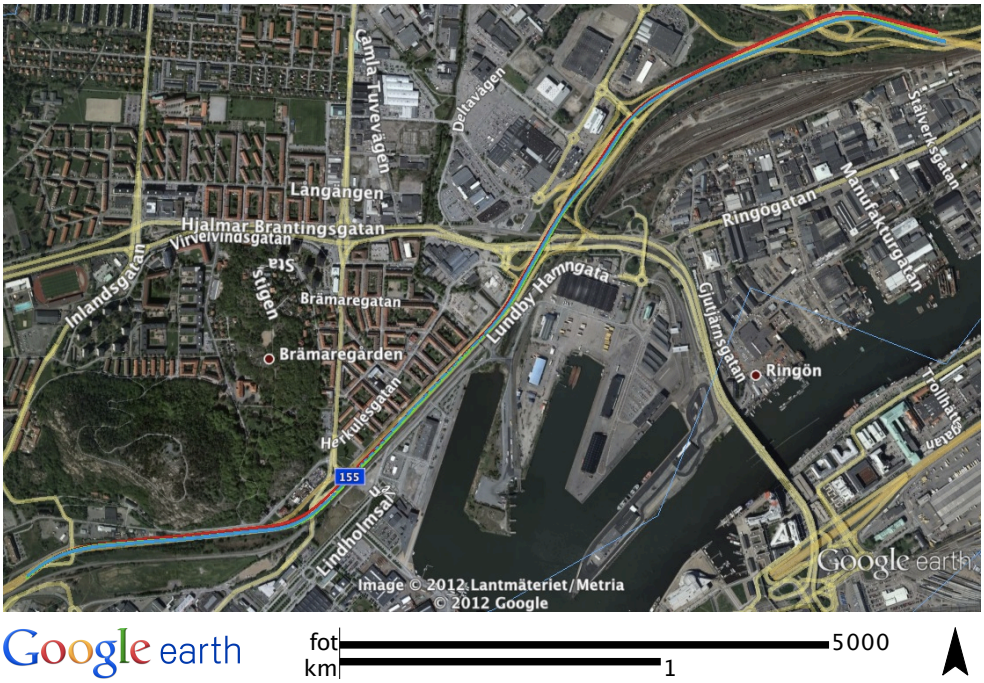


Figure 8.10: Overview of the highway data collection scenario; from Lundbytunneln in the west towards Tingstadstunneln in the east. Turquoise, green and red represents the measurements from RT-3002, BU-353 and our Kalman filter output, respectively.

As seen in Figure 8.11 the BU-353 output is similar to that of the RT-3002. After a few turns the distance between RT-3002 position output and the filter estimated position output increases. The difference between filter output and RT-3002 is estimated to be almost 10m in the top right corner of the figure.

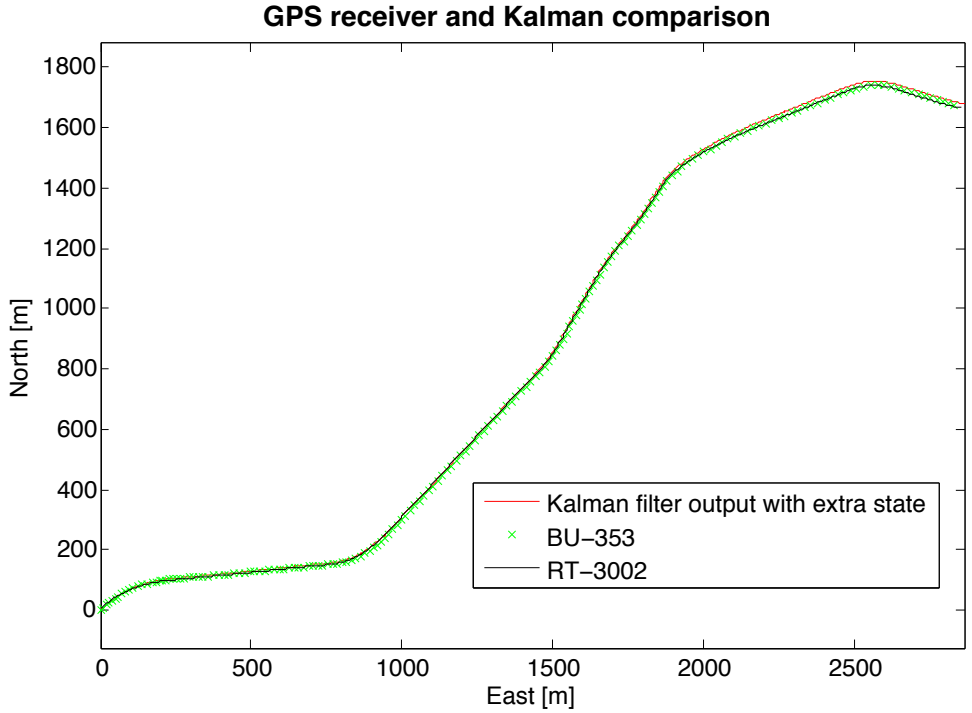


Figure 8.11: Comparison of kalman filter output, BU-353 and RT-3002 using Matlab plot.

8.4 Test track

Most of the time RT-3002 is working in a position mode called *RTK Integer* [56], which enables position accuracy of 2 cm (1σ). Therefore we assign the RT-3002 position estimates as reference for accuracy comparison with BU-353 measurements and Kalman filter output.

The test track is situated in a rural area without high obstacles covering the sky, which can be considered as an optimal environment for GPS receiver performance. Figure 8.12 shows an overview of the test track.

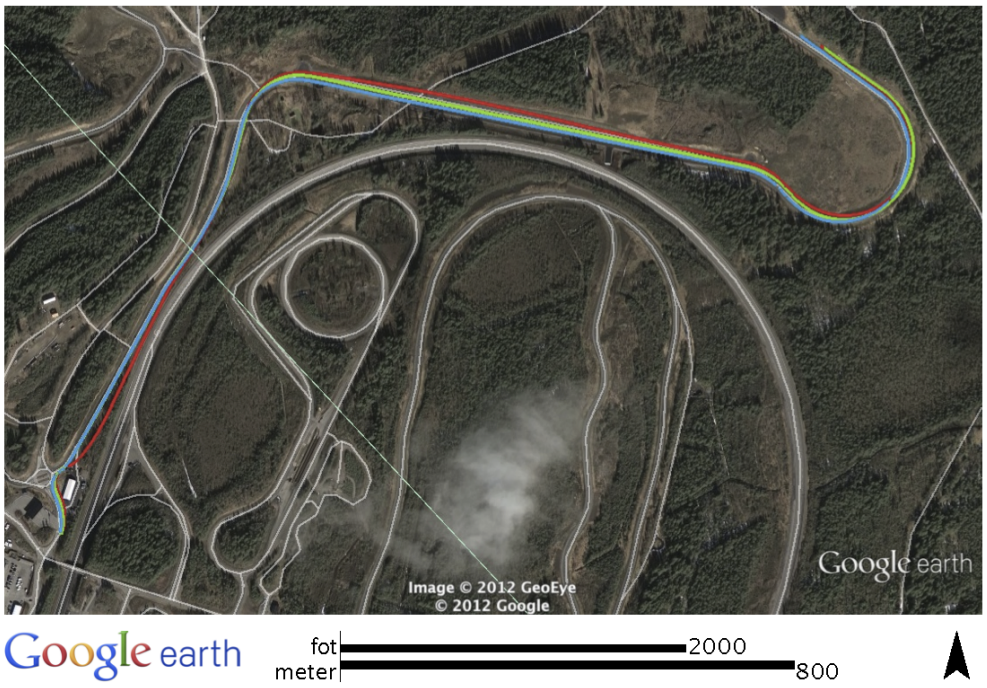


Figure 8.12: Overview of data collection scenario; Hällered test track. Turquoise, green and red represents the measurements from RT-3002, BU-353 and our Kalman filter, respectively.

In Figure 8.13 it is clear that the BU-353 position measurements are more accurate than the Kalman filter output, when using default parameter settings and implementation of the extra gyro bias state.

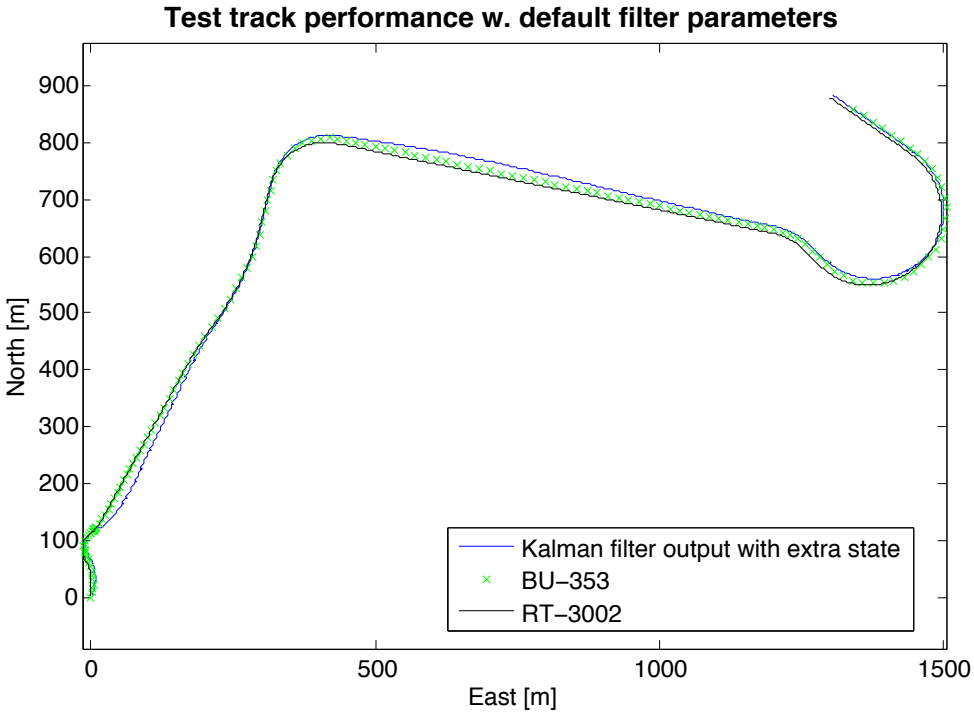


Figure 8.13: Performance of the Kalman filter using default settings, and with the extra gyro bias state, compared with BU-353 and RT-3002 position output.

Because of the high accuracy position estimates available from the RT-3002, it would be of interest to perform a quantitative comparison with the BU-353. However this was not done due to time synchronization problems due to data overlap, as seen in Figure 7.7, which is discussed in section 9.2.

8.5 Sample time and synchronization of signals

In our Kalman filter we make use of a varying sample time that changes for every Kalman filter iteration (Figure 8.14).

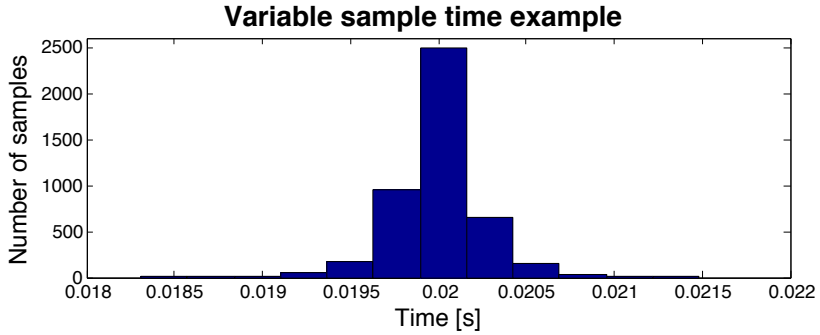


Figure 8.14: Example histogram of time between samples for a three-minute data set.

However CAN time stamps are not made at the moment of measure, but on the moment of transmission. Since our inertial and vehicle speed signals were filtered before provided on the CAN bus (HSCAN), and thus delayed, it is of interest to know how such latency could effect the output of the sensor fusion system. This was done by comparing the Kalman filter output at different constant sample times T [s].

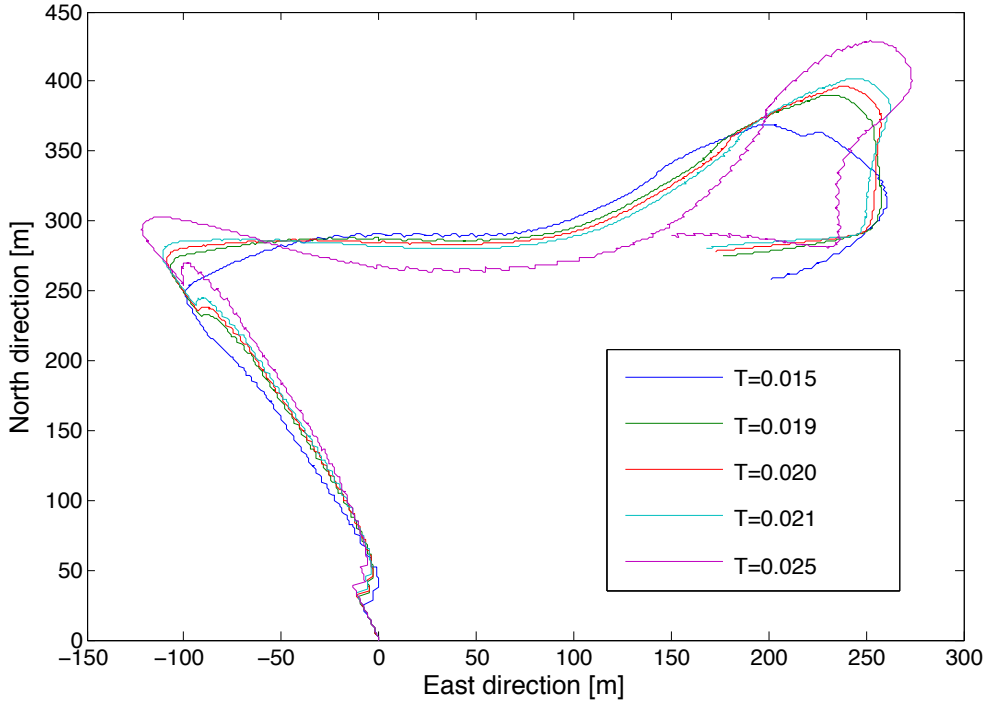


Figure 8.15: Comparison of Kalman filter output for different fixed sample rates.

As seen in Figure 8.15, small changes in the sample time T generates large differences in the Kalman filter output.

8.6 Map system differences

It is easy to assume that map data is accurate. However by comparing two map data sources, Google Maps and Yahoo maps, in Figure 8.16 it is evident that these are not equivalent, and the accuracy of these map systems is unknown.



Figure 8.16: Google Maps specify the position as off-road (left), while Yahoo maps specify the position on the road (right).

This effect the quality of of our OnRoad function, but also how we interpret the performance of the Kalman filter and GPS receivers by map comparison. Ultimately it give rise to the question of map reliability, see section 9.3 for further discussion.

8.7 Determining GPS measurement quality

It can be difficult to determine the quality of GPS measurements, however there are a few signals provided by the BU-353 GPS receiver that can be of interest to analyze. Following

GPS signals were analyzed, which were collected in city environment (Figure 8.17; horizontal dilution of precision (HDOP) (section 3.1.1), SNR and the number of visible satellites (Figure 8.18).

HDOP relates to the DOP for lateral and longitudinal measurements, while vertical DOP (VDOP) relates to DOP for altitude measurements.

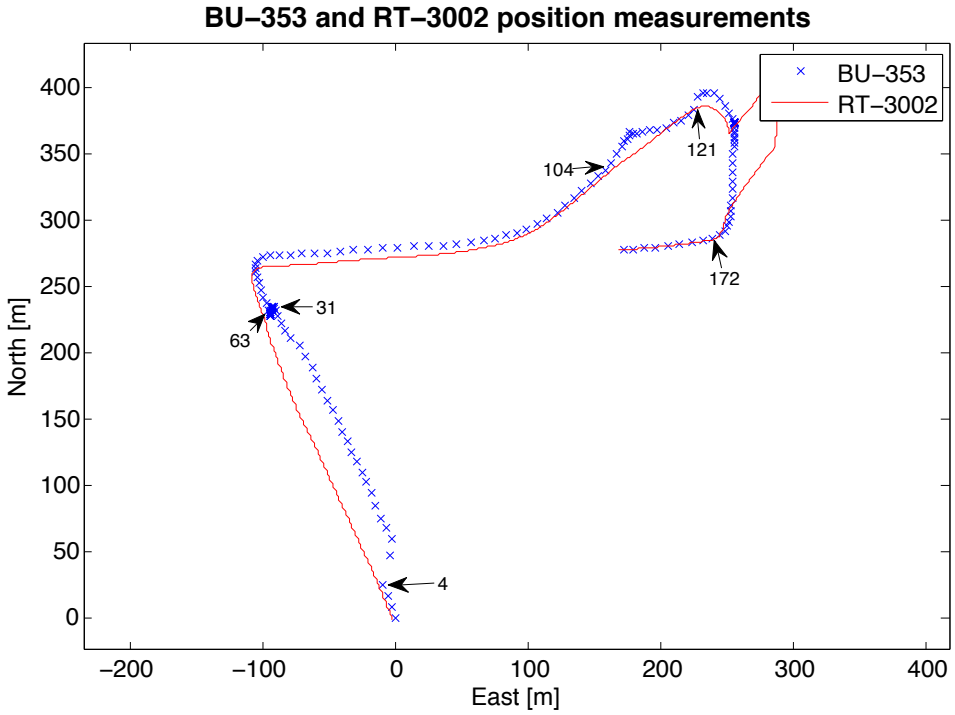


Figure 8.17: Difference between RT-3002 and BU-353. Arrows point at samples of interest, marked with a time stamp (seconds).

Interesting samples are extracted from Figure 8.17. These samples are as follows;

- $t = 4$ seconds: just before a jump in BU-353 position measurement.
- $t = 31$ seconds: vehicle is stationary, BU-353 position measurement is converging towards the RT-3002 position output.
- $t = 63$ seconds: vehicle starts to move, BU-353 position measurement is still converging towards the RT-3002 position output.
- $t = 104$ seconds: before deviation from RT-3002 position output.
- $t = 121$ seconds: before another deviation from RT-3002 position output.
- $t = 172$ seconds: after last turn in this data segment, BU-353 and RT-3002 position outputs are similar.

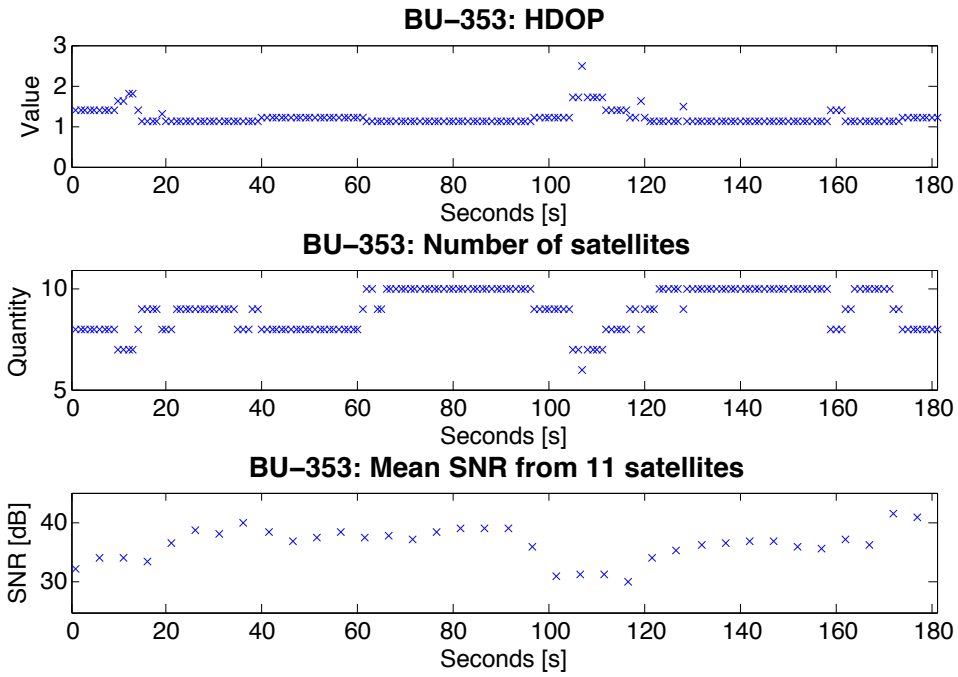


Figure 8.18: HDOP, SNR and number of satellites in view for the data set which trajectory is presented in Figure 8.17.

HDOP and number of satellites are updated every second, SNR is updated every 5th second (Figure 8.18). When Figures 8.17 and 8.18 are compared a concurrent change in the signals occurs between $t \approx 100$ and 120 seconds. During this time there is a deviation in the BU-353 position measurements compared to the RT-3002 output. As can be noticed in Figure 8.18 HDOP increases, number of visible satellites and SNR decreases which indicates that the quality for this measurement is low.

Apart from this signal concurrency, no evident pattern can be seen in this data set. See section 9.1.1 for further discussions concerning the determination of GPS measurement quality.

9 Discussion

In this section we will address issues that was encountered throughout our thesis that will be of importance when analyzing our results and for further development of a sensor fusion model for navigational purposes.

The scenarios that were used to provide many of our results (section 8), which we discuss in this section, are considered to cover sufficient dynamics and aspects of problems regarding absolute position estimation of an automotive vehicle. It is to our knowledge that each data sequence is representative for their individual scenario (city, highway etc.).

9.1 Sensors and signals

In this section discussions about GPS receivers, mainly the BU-353 and RT-3002, IMU and wheel tick sensors and their corresponding signals will be addressed.

9.1.1 GPS receiver

Our low-cost GPS receiver unit BU-353 was, as depicted in Figure 6.2, placed approximately one decimeter from the RT-3002 GPS receiver antenna. This difference should be

taken into consideration when analyzing the results, however since we do not have any quantitative results due to synchronization problems (section 7.2.3 and 9.2), this difference was omitted in our results. Instead results regarding robustness are provided.

According to the specifications of BU-353 [47], startup times can vary between 0.1 to 45 seconds depending on situation. Therefore it is desirable, in commercial automotive vehicles, to use A-GPS functionality to reduce startup times [57].

Accuracy of GPS receiver measurements are very dependent on the environment in which the GPS receiver is placed. Position measurements with higher accuracy are to be expected in environments where there is a free-field view of the sky, as can be seen in the highway simulation results (section 8.3), and thus the opposite can be expected, which was also confirmed, in the city environment (section 8.1). In fact the BU-353 position measurements in certain environments are so accurate that the sensor fusion model should be altered to place smaller position measurement variances in such cases (section 7.1.3).

GPS heading measurements proved to have a big impact on our sensor fusion model, thus heading measurements have to be reliable. However we could not find any sources worthy of citation that stated how a GPS receiver calculates heading, but it is assumed that calculations based on the difference of position measurements are used, which would explain the jumps in the heading measurements at low velocities (section 7.2.1). Another way that GPS receivers can calculate heading could be from Doppler measurements, which then would provide both velocity (absolute value) and heading (direction) of the receiver unit. According to [41] the accuracy of heading measurements could be improved with the use of two GPS receiver antennas. For further discussion about heading in the sensor fusion model see section 9.3.

Doppler measurements are known to provide a GPS receiver with velocity information (section 3.1), however they were omitted since these are considered to be inaccurate [28], [20], and odometer originated data (vehicleSpeedOverGnd) was used instead. Although this signal is also subject to noise and errors, it was considered to be more accurate which coincided with conclusions from previous research.

Coordinate transformations used in this thesis make use of latitude, longitude and altitude GPS data, however altitude measurements have significantly less accuracy than the other two [26]. In our thesis we neglected altitude measurements, and let altitude be set to zero which can result in inaccurate position estimations when driving on roads with altitude differences. This leads to a discussion regarding the accuracy of the coordinate transformations, which is further discussed in section 9.3.

An attempt at estimating GPS measurement quality was made by comparing BU-353 position data, with corresponding HDOP, SNR and a signal that provided the number of satellites. Results in Figure 8.18 showed one case where the signals concurrently provided information that the measurement was of low quality, but that no other overall conclusion could be made. After $t = 4$ seconds in Figure 8.17 the BU-353 position measurement is subject to a large jump, but it is not visible in neither HDOP, number of satellites or SNR. Instead SNR increases, HDOP decreases and the number of visible satellites is overall stationary throughout the interval $t = 4$ to $t = 31$ when the BU-353 position measurement, according to Google Maps, is highly inaccurate.

It could be a problem with using information regarding number of satellites in view, because GPS receivers normally do not make use of all these satellites when calculating a position. According to [17] the signal number of satellites in view does not provide how many satellites that are actually visible by the GPS receiver, but instead show how many satellites that should be visible given a certain position and the corresponding current GPS constellation. Therefore data concerning how many satellites that actually is in view, or preferably how many that are being used in calculation, could be of greater interest.

9.1.2 Inertial measurement unit and wheel speed

To analyze the behavior of the sensor signals provided on HSCAN, a sensor characteristics analysis was made (section 6.2.2). As a conclusion of this analysis there seemed to be no drift in either of the accelerometer signals as well as the yaw rate and wheel speed data, which was due to that the signals were filtered.

We believe that the filtering being made on the sensor data before transmission on HSCAN make use of the information that the vehicle is not moving, i.e. no wheel rotation, thus correcting some of the signals accordingly. The most cryptic signal we used that was provided on HSCAN was vehicleSpeedOverGnd, which was, according to sources on Volvo Car Corporation, derived from a complex model given data from all four wheels.

A problem with using filtered data is that the sensor fusion performance cannot be validated in the same way as when not using raw sensor data. However there still seems to be an issue of drift in the filtered yaw rate signal when the vehicle is turning (Figure 8.11 and 8.13), which corresponds to previous research [29].

If a new drift data analysis would be made, using raw sensor data, Allan Variance would be a suitable method to parametrize drift and bias in the sensors (Appendix D). In order to perform such an analysis correctly, accelerometers acquiring inertial measurements in three directions, a so-called 3D accelerometer, would be required to account for biases created by the gravitational force. With a new drift data analysis, test should also be performed where the vehicle is subject to dynamics by driving around a round-a-bout for a couple of revolutions similar to that in [29]. Such an evaluation could be made by comparing, e.g., integrated yaw rate combined with velocity data, with accurate position measurements from an RTK-GPS.

The ability of accelerometers in three dimensions to account for the gravitational force would also increase the reliability of measurements when the vehicle is situated in slopes or banks as in the scenario depicted in Figure 8.9. Many IMU have six degrees of freedom, i.e. they incorporate both 3D accelerometers and 3D gyroscopes, which can aid the sensor fusion output. However focus regarding sensors to improve absolute position estimation through sensor fusion should first of all be placed on incorporating 3D accelerometers.

Since IMU sensor accuracy cannot be quantitatively analyzed, due to that the signals are pre-filtered, it is difficult to discuss if these sensors are accurate enough, or if they should be replaced. Instead new sensor analysis should be made from raw sensor data provided on a different CAN bus called SENSOR-CAN.

9.2 Data synchronization

We encountered a problem in the position estimates provided by the RT-3002, which is that there are a few samples overlap between the three-minute data segments (section 7.2.3, RT-3002 data overlap). If removed, the sampling rate of 100 Hz would no longer be valid. This was why no quantitative comparison between BU-353 position measurements and RT-3002 position estimates could be made (section 8.4).

The rate at which the signals are transmitted on CAN may vary, depending on the structure of the CAN and what nodes that currently have priority. Because synchronization is of great importance in sensor fusion for navigational purposes, it would be beneficial to have a fixed rate at which sensor data is received in the sensor fusion model.

9.3 Sensor fusion model

There are different ways to perform sensor fusion, and Kalman filtering is a common choice. Because of the nonlinearity in both the process and measurement model of our system,

an Extended Kalman filter was a suitable option. The filter is common, and there exist plenty of research about this filter, also for navigational purposes. However Unscented Kalman filter has been proved to be slightly more efficient [25], and some research show good results by using neural networks to perform sensor fusion [7]. However we did not acquire any results pointing towards that the non-satisfactory output of the sensor fusion model was due to the usage of EKF.

The process model of our Kalman filter model was derived from kinematic relations between the states. Another model that would be interesting to analyze is the bicycle model that also take slip into account (section 7). However in our data collection, the vehicle was never subject to dynamics that would require estimation of slip.

A different model that estimates heading could be of interest, since heading proved to make a big difference in the resulting sensor fusion model position output. In our model we measure absolute heading with the GPS receiver, and relative heading (yaw rate) with a gyroscope sensor. If the absolute heading is inaccurate at the same time as GPS measurements cease, there will be a bias error in the Kalman filter output. This is seen in Figure 8.5, where the OnRoad function is enabled after just a few seconds, ceasing absolute heading measurements, thus having the Kalman filter estimating heading simply from yaw rate. The bias offset in the heading state will effect the filter until GPS measurements are resumed.

It can be seen in Figure 8.5 that when our Kalman filter is initialized with states equal to zero, instead of being initialized by measurements, outputs are far less robust. However if the Kalman filter is given time to converge, there is no noticeable difference between using optimal initialization or not, which displays the overall robustness of the Kalman filter.

Because we assumed zero altitude relative the WSG84 ellipse in our coordinate transformations (section 7.2.2), errors will occur when the vehicle is subject to a change in altitude. This can be seen in many simulation result plots, e.g. Figure 8.2, where it could be one of the reasons why the Kalman filter output have trouble following the "true" trajectory in curves. Driving in a slope will result in a 2D position movement that is different than when considering a 3D environment, e.g. driving 100 meters in a slope with 5 degree inclination will result in an error of approximately 0.4 meters. By adding a altitude state in the Kalman filter, position estimates in 3D would be possible.

Some filter parameters were easier to tune since estimates could be derived from the information given by analyzing the HSCAN signals, and from specifications of the BU-353 GPS receiver. However process noise variances were derived through trial and error (section 7.1.3). A way of optimally determining Kalman filter parameters could be of importance to provide a more accurate Kalman filter output. Because measurement variances are impossible to estimate on pre-filtered signals, we again reckon that the use of raw sensor data is important.

The extra gyro bias state added to the Kalman filter gave position outputs closer to that of the RT-3002 for all sequences analyzed, e.g. in Figure 8.2. In the same way it could be of interest to analyze the impact on filter output performance by adding states estimating accelerometer and wheel tick bias.

A tool for improving absolute position estimates, both accuracy and robustness, is map matching (section 5.2.3). Much of the research in this field now incorporates the use of map data to match features in the environment with a position on the map. We did not use any map matching in this way, however we created a function that contained the information about whether or not BU-353 position measurements acquires a position on the road or not, called OnRoad (section 7.1.4).

Figure 8.4 and 8.9, respectively, presents examples of when the OnRoad function im-

proves absolute position estimation, and when it doesn't. Using the OnRoad function is generally a good alternative when there are good conditions for dead reckoning, as in Figure 8.4 where there is a more or less straight road in the beginning of the data sequence. In Figure 8.9 the OnRoad function provides a Kalman filter position estimate that is less accurate than BU-353 position measurements, if compared to the RT-3002 output. This could be due to the difficulty of the filtered sensor signals to responds to vehicle movements on roads with high dynamics. Therefore it is difficult to address one specific cause to the non-satisfactory filter performance when using this function.

The OnRoad function is dependent on accurate map data, and in the results of section 8.6 it is clear that the accuracy of map data has to be further evaluated. Map data also has to be used when dealing with methods such as lane tracking (section 5.2.2), which would be an interesting addition to the sensor fusion model presented in this thesis. Map data could also contain information regarding if the street, or road, of which the vehicle is present is GPS receiver friendly or not, i.e. what quality of GPS measurements that are to be expected.

9.4 Position reference issue

Although the RT-3002 is presented as a highly accurate device, it is still a problem in using its output as position reference for evaluating a sensor fusion models performance when corrections from a base station are unavailable. As can be seen in Figure 8.1 the RT-3002 position output jumps into a building near the end of that data sequence. In this sequence the RT-3002 cannot perform any measurement corrections, and there is still a main issue regarding multipath errors and other effects that are difficult to detect.

Although the RT-3002 contains an accurate six degree-of-freedom IMU and performs sensor fusion between its RTK-GPS and INS-system, it is not able to account for GPS measurements problems. The Kalman filter used in the RT-3002 is not able to adapt to such events, in the same way our filter had before implementing the OnRoad function. However the OnRoad function cannot maintain a final solution to this problem.

9.5 Real-time implementation challenges

There are a lot of things that need to be tested off-line prior to implementing a real-time solution in a vehicle to improve absolute position estimation. Creating a real-time system involves new issues that has to be addressed.

The problem regarding computational latency has to be analyzed further, which includes conversion of heading angle measurements, but most importantly coordinate transformations of GPS measurements from the global format LLA to the local format ENU, and overall filter latency. However the RT-3002 unit is able to provide outputs with only 3.5 ms latency after an initialization of 10 seconds according to specifications [56], which proves that it is possible to create a system similar to ours with sufficiently low latency for real-time purposes.

Converting GPS positions from LLA to ENU also rely on a reference point to address as origin for the local ENU coordinate system. The further away from this reference point, the less accurate the local ENU approximation becomes. Thus it is of importance to evaluate how new reference points could be introduced to the system. We excluded the change of reference point in our model because we only used three-minute data sequences, which did not enable the vehicle to move that far away from the ENU origin reference point.

Apart from the real-time implementation issues brought up earlier, the most important topic concerning this matter is data synchronization, which is also discussed in section 9.2.

In a real-time system, data synchronization becomes an entirely different issue since every data input has to be continuously synchronized. It is also important to be able to estimate the measurements at time of measure, because there is a time delay between point of measure and the CAN message time stamp. The rate at which information is provided on CAN may vary, which can be seen in our results (section 8.5), and has to be addressed to enable a sufficiently accurate real-time sensor fusion model for an automotive vehicle implementation.

10 Conclusion

There are a lot of research regarding the use of EKF for navigational purposes, with good results concerning accuracy and robustness. In our thesis we could not provide satisfying result regarding neither robustness or accuracy in general. However there are some situations where our filter output is improved compared to that of solely the BU-353 measurements, using the RT-3002 output as reference.

We have not come across any issue that to our knowledge can be explicitly related to the use of EKF as a tool for sensor fusion, thus further research with this filter can be of interest. However focus should be placed on other aspects of our sensor fusion model, such as creating a model that allows the vehicle to move in three dimensions, and to acquire better estimates of the GPS measurement quality.

Although it has sometimes been difficult to trace an error to a specific source, we have encountered some issues that needs to be addressed further. When the vehicle is subject to large changes in altitude, our model will not function properly because it only considers a two-dimensional space. By adding an altitude state in the process model, and providing measurements of that state from a GPS receiver, such errors could be reduced.

The accuracy of heading proved to be important in providing good results, especially when GPS measurements were not available, because the only absolute heading measure came from the GPS receiver. If the Kalman filter would also estimate heading by differentiation of current and previous position estimates, redundancy and accuracy could increase.

Adding rules in the Kalman filter to alter the measurement covariance matrix when GPS measurements are inaccurate would improve the performance of the filter. However it was difficult to estimate the GPS measurement quality from the signals we studied; HDOP, SNR and number of satellites, because no general pattern could be found.

The lack of accelerometer measurements in all three dimensions effects the reliability of the two-dimensional accelerometer data acquired due to the inability to encounter for the error originated from gravitational forces. This problem occurs especially when driving in slopes or when the road is banked, but will have minimum contributions when the vehicle is situated on a horizontally leveled space.

Instead of extracting pre-filtered sensor signals from HSCAN, raw sensor data that can be extracted from SENSOR-CAN should be used. Raw sensor data is easier to analyze and to parametrize than filtered signals. It would also be beneficial to be able to control the entire process of sensor fusion, from acquiring sensor data to a filtered output, in order to evaluate filter performance and to deal with related problems in the system.

A fixed sampling rate would simplify signal processing, but it is even more important to be able to get a time stamp related to the time of measure rather than the time of transmission on CAN.

Sensor fusion for improving absolute position estimates using GPS, IMU and odometer is not sufficient to provide a robust and accurate system for automotive vehicle purposes.

Methods such as lane tracking and traffic sign localization together with map matching should be incorporated in a sensor fusion model to improve absolute position estimates of an automotive vehicle.

References

[1] Ieee standard specification format guide and test procedure for single-axis interferometric fiber optic gyros. *IEEE Std 952-1997*, pages 62–73, 1998.

[2] Leap second. *GPS World*, 23(2):11, 2012.

[3] Merriam-webster's dictionary. Jun 2012.

[4] Oxford english dictionary. May 2012.

[5] 3rd International Symposium on Imprecise Probabilities and Their Applications. *An Extended Set-valued Kalman Filter*, June 2003. http://www.carleton-scientific.com/isipta/PDF/030.pdf Achieved 2012-04-02.

[6] John W Allen. Use of vision sensors and lane maps to aid gps-ins navigation. Master's thesis, Auburn University, 2011.

[7] Ali Asadian, Behzad Moshiri, Ali Khaki Sedigh, and Caro Lucas. Optimized data fusion in an intelligent integrated gps/ins system using genetic algorithm, 2005.

[8] P. Axelrad, C.J. Comp, and P.F. Macdoran. Snr-based multipath error correction for gps differential phase. *Aerospace and Electronic Systems, IEEE Transactions on*, 32(2):650 –660, april 1996.

[9] C Chen. *Low-cost loosely-coupled GPS/odometer fusion: a pattern recognition aided approach*, pages 1603–1608. IEEE, 2008.

[10] V.S. Chernyak and I. Immoreev. A brief history of radar. *Aerospace and Electronic Systems Magazine, IEEE*, 24(9):B1 –B32, sept. 2009.

[11] Rolf Christensen and Nikolaj Fog. Inertial navigation system. Master's thesis, Aalborg University, 2008. http://www.control.aau.dk/uav/reports/08gr1030a/08gr1030a_student_report.pdf Achieved 2012-03-27.

[12] James R. Clynch. Geodetic coordinate conversions, Feb. 2006. http://www.gmat.unsw.edu.au/snap/gps/clynch_pdfs/coordcvt.pdf Achieved 2012-02-02.

[13] Stelian Cojocaru. Gps-glonass-galileo: A dynamical comparison. *The Journal of Navigation*, 62(1):135–170, january 2009.

[14] Youjing Cui and Shuzhi Sam Ge. Autonomous vehicle positioning with gps in urban canyon environments. *Robotics and Automation, IEEE Transactions on*, 19(1):15 – 25, feb 2003.

[15] Lars Danielsson. *Tracking Theory for Preventive Safety Systems*. Licentiate thesis, Chalmers University of Technology, Gothenburg, Sweden, April. 2008.

[16] Robert Davies. Controller area network (can) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35(3):239–272, april 2007.

[17] Dale DePriest. Nmea mode. http://www.gpsinformation.org/dale/nmea.htm.

[18] S. P. Drake. Converting gps coordinates ($\phi\lambda h$) to navigation coordinates (ENU). Technical report, Surveillance Systems Division Electronics and Surveillance Research Laboratory, 2002. http://dspace.dsto.defence.gov.au/dspace/bitstream/1947/3538/1/DSTO-TN-0432.pdf Achieved 2012-03-27.

[19] Bernd Eissfeller. Method and apparatus for eliminating ionospheric delay error in global positioning system signals. Institute of Geodesy and Navigation, University FAF Munich, April 2012. http://www.ifen.unibw-muenchen.de/research/coupling.htm.

[20] Ahmed El-Rabbany. *Introduction to GPS: The Global Positioning System, Second Edition*. Artech House Publishers, 2006.

[21] Katti Faceli, André C. de Carvalho, and Solange O. Rezende. Combining intelligent techniques for sensor fusion. *Applied Intelligence*, 20(3):199–213, 2004.

[22] Takehisa Fujita and Naohiro Kanda. Physics of leap second. Department of Physics, Faculty of Science and Technology, Nov 2009. http://arxiv.org/pdf/0911.2087v1.pdf, Achieved 2012-05-05.

[23] I. Graham. Looking back: some radar recollections. *Potentials, IEEE*, 18(5):40 –42, jan 1999.

[24] M. Grewal and A. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley, 2008.

[25] Mohinder S. Grewal. *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley, John & Sons, Incorporated, 2001.

[26] John G. Grimes. Global positioning system standard positioning service performance standard. Department of Defense: United States of America, september 2008. 4th edition.

[27] Yanlei Gu, T. Yendo, M.P. Tehrani, T. Fujii, and M. Tanimoto. Traffic sign detection in dual-focal active camera system. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 1054 –1059, june 2011.

[28] Vishisht Gupta. *VEHICLE LOCALIZATION USING LOW-ACCURACY GPS, IMU AND MAP-AIDED VISION*. Phd dissertation, The Pennsylvania State University, 2009.

[29] F. Gustafsson, S. Ahlqvist, U. Forssell, and N. Persson. Sensor fusion for accurate computation of yaw rate and absolute velocity. january 2001.

[30] D.L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6 –23, jan 1997.

[31] Ingo Harre. Gaussian probability density functions: Properties and error characterization, May 2001. http://www.mar-it.de/NavGen/final_text3.pdf, Achieved 2012-05-16.

[32] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer, 2001.

[33] Jeroen Hol. *Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS*. Phd dissertation, Linkoeping University, 2011.

[34] S.H. Huynh and George Cheng. Low profile ceramic choke. Antcom Corp., Mar 2000. Patent number: 6,040,805.

[35] USGlobalSat Inc. Bu-353 support page (driver, manual and specification etc.), 2010. http://www.trimble.com/construction/marine/sps852_gnss_modular_receiver.aspx?dtID=specifications Achieved 2012-04-13.

[36] M. Jabbour, P. Bonnifait, and V. Cherfaoui. Road tracking for multi-hypothesis localization on navigable maps. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 138 –143, june 2008.

[37] F. Jahard, D.A. Fish, A.A. Rio, and C.P. Thompson. Far/near infrared adapted pyramid-based fusion for automotive night vision. In *Image Processing and Its Applications, 1997., Sixth International Conference on*, volume 2, pages 886 –890 vol.2, jul 1997.

[38] D Jwo. Efficient dop calculation for gps with and without altimeter aiding. *The Journal of Navigation*, 54(2):269–279, 2001.

[39] Rudolph Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME - Journal of Basic Engineering*, pages 35 – 45, March 1960.

[40] Volker Kempe. *Inertial MEMS: Principles and Practice*. Cambridge University Press, 2011.

[41] Jiunhan Keong and Gerard Lachapelle. Heading and pitch determination using gps/-glonass. *GPS Solutions*, 3:26–36, 2000.

[42] ZuWhan Kim. Robust lane detection and tracking in challenging scenarios. *Intelligent Transportation Systems, IEEE Transactions on*, 9(1):16 –26, march 2008.

[43] Anna Kochan. Radar is set to revolutionise road safety. *Sensor Review*, 23(2):109–115, 2003.

[44] C. Lambiel and R. Delaloye. Contribution of real-time kinematic gps in the study of creeping mountain permafrost: examples from the western swiss alps. *The Nature and Dynamics of Mountain Permafrost: Papers from the PACE21 Contribution to the International Permafrost Association*, 15:229 –241, july-september 2004.

[45] Richard B Langley. Dilution of precision. *GPS World*, 10(5):52, may 1999.

[46] W Lewandowski. Gnss times and utc. *METROLOGIA*, 48(4):219–224, august 2011.

[47] Trimble Navigation Limited. Specifications of the trimble sps852 gnss modular, 2009. http://www.usglobalsat.com/s-122-bu-353-support.aspx Achieved 2012-04-13.

[48] Yun Luo, J. Remillard, and D. Hoetzer. Pedestrian detection in near-infrared night vision system. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 51 –58, june 2010.

[49] Bassem R. Mahafza. *Radar Signal Analysis and Processing Using MATLAB*. Chapman and Hall/CRC, 2008.

[50] David McNeil Mayhew. Multi-rate sensor fusion for gps navigation using kalman filtering. Master's thesis, Virginia Polytechnic Institute and State University, 1999. http://scholar.lib.vt.edu/theses/available/etd-062899-064821/ Achieved 2012-04-01.

[51] Kevin Milton and Ron Meyer. Method and apparatus for eliminating ionospheric delay error in global positioning system signals. Rockwell Science Center, Inc, May 1999. 5,903,654.

[52] H.B. Mitchell. *Multi-Sensor Data Fusion: An Introduction*. Springer, 2007.

[53] William M. Mularie. Department of defense world geodetic system 1984. its definition and relationships with local geodetic system. Technical report, National Imagery and Mapping Agency, Jan. 2000. http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf Achieved 2012-03-28.

[54] National Technical Meeting of The Institute of Navigation, Anaheim, CA. *Performance Analysis and Architectures for INS-Aided GPS Tracking Loops*, January 2003.

[55] Dragan Obradovic, Henning Lenz, and Markus Schupfner. Fusion of sensor data in siemens car navigation systemn. *IEEE*, 56(1):43–50, 2007.

[56] Oxford Technical Solutions. *RT Inertial and GPS Measurement Systems: User Manual*, 2011.

[57] Bryant R. Assisted gps. 16(5):40–46, may 2005.

[58] Maria Isabel Ribeiro. Gaussian probability density functions: Properties and error characterization. Institute for Systems and Robotics, Instituto Superior Tcnico, Feb 2004. http://users.isr.ist.utl.pt/~mir/pub/probability.pdf, Achieved 2012-05-16.

[59] C. Rizos and University of New South Wales. *Principles and practice of GPS surveying / Chris Rizos*. Monograph (University of New South Wales. School of Geomatic Engineering) ; no. 17. School of Geomatic Engineering, University of New South Wales, Kensington, N.S.W. :, 1997.

[60] D. Schleicher, L.M. Bergasa, M. Ocana, R. Barea, and M.E. Lopez. Real-time hierarchical outdoor slam based on stereovision and gps fusion. *Intelligent Transportation Systems, IEEE Transactions on*, 10(3):440 –452, sept. 2009.

[61] M.S. Schlosser and K. Kroschel. Limits in tracking with extended kalman filters. *Aerospace and Electronic Systems, IEEE Transactions on*, 40(4):1351 – 1359, oct. 2004.

[62] J. Schmackers and A. Glasmachers. Landmark based fast positioning for sensor data fusion; receiver design and measurement results. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 25 –30, oct. 2011.

[63] Michael Shaw. Modernization of the global positioning system. *Acta Astronautica*, 54:943 –947, 2004.

[64] D.A. Shnidman. Radar detection in clutter. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(3):1056 – 1067, july 2005.

[65] S. Sivaraman and M.M. Trivedi. Improved vision-based lane tracker performance using vehicle localization. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 676 –681, june 2010.

[66] I. Skog and P. Handel. In-car positioning and navigation technologies #x2014;a survey. *Intelligent Transportation Systems, IEEE Transactions on*, 10(1):4 –21, march 2009.

[67] Portland State Aerospace Society. Conversion of geodetic coordinates to the local tangent plane, Sep. 2007. http://psas.pdx.edu/CoordinateSystem/Latitude_to_LocalTangent.pdf Achieved 2012-02-28.

[68] T. Soler. A compendium of transformation formulas useful in gps work. *Journal of Geodesy*, 72:482–490, April 1998. http://users.auth.gr/~kvek/720-482_SOLER-compendium.pdf Achieved 2012-03-28.

[69] Jeff Stefan. A protocol for gps communications. *Embedded Systems Programming*, 12(5):58, may 1999.

[70] Gideon P. Stein, Ofer Mano, and Amnon Shashua. Vision-based acc with a single camera: Bounds on range and range rate accuracy. http://www.mobileye.com/wp-content/uploads/2011/09/VisiobBasedACC.pdf, Achieved 2012-06-18.

[71] Q Sultana. Effects of pseudolite positioning on dop in laas.(dilution of precision and local area augmentation system). *Positioning*, 1(1):18–21, november 2010.

[72] David Titterton and John Weston. *Strapdown Inertial Navigation Technology (IEE Radar, Sonar, Navigation and Avionics Series)*. The Institution of Engineering and Technology, 2005.

[73] Trimble Navigation Limited Engineering and Construction group. *Trimble R8 GNSS Receiever: User Manual*, september 2009.

[74] O Tsimhoni, J Bargman, T Minoda, and M. Flannagan. Pedestrian detection with near and far infrared night vision enhancement. Technical report, The University of Michigan Transportation Research Institute, December 2004.

[75] A. Weckenmann, X. Jiang, K.-D. Sommer, U. Neuschaefer-Rube, J. Seewig, L. Shaw, and T. Estler. Multisensor data fusion in dimensional metrology. *CIRP Annals - Manufacturing Technology*, 58(2):701 – 721, 2009. http://www.sciencedirect.com/science/article/pii/S0007850609001759 Achieved 2012-04-01.

[76] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Department of Computer Science, University of North Carolina at Chapel Hill, 2006.

[77] Oliver J. Woodman. An introduction to inertial navigation. Technical Report 696, Computer Laboratory, University of Cambridge, 2007.

[78] Guochang Xu. *GPS: Theory, Algorithms and Applications*. Springer, 2007.

[79] Thomas P Yunck, William I. Beringer, and Stephen M. Lichten. Robust real time wide area differential gps navigation. The United States of America as Represented by the Administrator of the National Aeronautics and Space Administration Washington DC, November 1998. 5,828,336.

[80] Peyton Z. and Peebles. *Radar Principles*. Wiley-Interscience, 1998.

A Matlab code: Error ellipses

```
function [ xx , yy ] = calcErrorEllipse( Eo,No,C,ci )
% CALCULATES ERROR ELLIPSE GIVEN POSITION COVARIANCE MATRIX, ORIGIN AND
% CONFIDENCE INTERVAL. PLOT THE RESULT WITH: plot(xx,yy)
%   Input:  - Eo: East origin of ellipse
%           - No: North origin of ellipse
%           - C: Position covariance matrix [2x2]
%           - ci: Confidence interval, 50 or 95 [%]
%
%   Output: - xx: Error ellipse plot x-vector
%           - yy: Error ellipse plot y-vector

if nargin<3
    error('Too few input arguements');
end
if nargin<4
    ci = 50;
end
if ci == 50
    K = 1.1572;
elseif ci == 95
    K = 2.449;
else
    error('Input either 50 or 95 percentage as confidence interval');
end
dt=0.1*pi/180;              % Angular resolution of plot (0.1 degrees)
t=[(0:dt:2*pi)';0];
[eVal,eVec]=eig(C);         % Eigenvalues and eigenvectors
r=K*sqrt(diag(eVec));       % Length of axes
x=r(1)*cos(t);
y=r(2)*sin(t);
xx=Eo+[x y]*eVal(1,:)';
yy=No+[x y]*eVal(2,:)';
end
```

B Matlab code: Heading angle adjustment

```
%% Transform heading measurement from compass degree format to ...
    radians on polar coordinates
    % This requires a shift of 90 degrees since North is 0 deg, and X ...
       axis
    % is 0 degrees in unit circle. But also it needs to flip 180 degrees
    % (thus the minus sign). Also convert to radians.
Measurements.heading =  Measurements.heading * pi/180;
notNaN_index = find(~isnan(Measurements.heading));
n1=zeros((length(Measurements.heading())),1);
n=0;
for i=2:length(notNaN_index)
    if Measurements.heading(notNaN_index(i)) - ...
       Measurements.heading(notNaN_index(i-1)) < -pi
        n = n+1;
    end
    if Measurements.heading(notNaN_index(i)) - ...
       Measurements.heading(notNaN_index(i-1)) > pi
        n = n-1;
    end
    n1(notNaN_index(i),1) = n;
end

 for i=1:length(notNaN_index)
     Measurements.heading(notNaN_index(i))= ...
        Measurements.heading(notNaN_index(i))+ ...
        n1(notNaN_index(i),1)*2*pi;
 end

Measurements.heading = - Measurements.heading + pi/2;
```

C Linearized Kalman Filter equations

C.1 Linearized Process and Measurement model of the Kalman Filter

Linearized Process model F -matrix

$$F = \begin{bmatrix} 1 & 0 & F_{1,3} & F_{1,4} & F_{1,5} & F_{1,6} \\ 0 & 1 & F_{2,3} & F_{2,4} & F_{2,5} & F_{2,6} \\ 0 & 0 & 1 & 0 & sign(v)T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.1}$$

$$\begin{aligned} F(1,3) &= -sin(\varphi)(vT + aT^2/2 - ... \\ &\quad cos(\varphi)(v\dot{\varphi}T^2/2 + 2a\dot{\varphi}T^3/6) \end{aligned} \tag{C.2}$$

$$F(1,4) = cos(\varphi)T - sin(\varphi)(\dot{\varphi}T^2/2) \tag{C.3}$$

$$F(1,5) = -sin(\varphi)(vT^2/2 + 2aT^3/6) \tag{C.4}$$

$$F(1,6) = cos(\varphi)T^2/2 - sin(\varphi)(2\dot{\varphi}T^3/6) \tag{C.5}$$

$$\begin{aligned} F(2,3) &= cos(\varphi)(vT + aT^2/2 - ... \\ &\quad sin(\varphi)(v\dot{\varphi}T^2/2 + 2a\dot{\varphi}T^3/6) \end{aligned} \tag{C.6}$$

$$F(2,4) = sin(\varphi)T + cos(\varphi)(\dot{\varphi}T^2/2) \tag{C.7}$$

$$F(2,5) = cos(\varphi)(vT^2/2 + 2aT^3/6) \tag{C.8}$$

$$F(2,6) = sin(\varphi)T^2/2 + cos(\varphi)(2\dot{\varphi}T^3/6) \tag{C.9}$$

Linearized Γ -matrix

$$\Gamma = \begin{bmatrix} \Gamma_{1,1} & \Gamma_{1,2} \\ \Gamma_{2,1} & \Gamma_{2,2} \\ sin(\varphi)T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix} \tag{C.10}$$

$$\Gamma_{1,1} = -sin(\varphi)(vT^3/6 + aT^4/8 + \dot{a}T^5/20) \tag{C.11}$$

$$\Gamma_{1,2} = cos(\varphi)T^3/3 - sin(\varphi)(\dot{\varphi}^4/8 + \ddot{\varphi}T^5/20) \tag{C.12}$$

$$\Gamma_{2,1} = cos(\varphi)(vT^3/6 + aT^4/8 + \dot{a}T^5/20) \tag{C.13}$$

$$\Gamma_{2,2} = sin(\varphi)T^3/3 + cos(\varphi)(\dot{\varphi}^4/8 + \ddot{\varphi}T^5/20) \tag{C.14}$$

Linearized Measurement model of the Kalman Filter

Linearized Measurement model H -matrix

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{C.15}$$

C.2 Linearized augmented matrices for extra bias state

Augmented state vector

$$x = \begin{bmatrix} E\ [m] \\ N\ [m] \\ \varphi\ [rad] \\ v\ [m/s] \\ \dot{\varphi}\ [rad/s] \\ a\ [m/s^2] \\ \dot{\varphi}^{bias}\ [rad/s] \end{bmatrix} \tag{C.16}$$

Linearized augmented process model: F -matrix

$$F = \begin{bmatrix} 1 & 0 & F_{1,3} & F_{1,4} & F_{1,5} & F_{1,6} & 0 \\ 0 & 1 & F_{2,3} & F_{2,4} & F_{2,5} & F_{2,6} & 0 \\ 0 & 0 & 1 & 0 & sign(v)T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.17}$$

Process nois covariance matrix using augmented state

$$P_{k=1} = \begin{bmatrix} 25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (10 \cdot \pi/180)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (0.1 \cdot \pi/180)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & (.0001 \cdot \pi/180)^2 \end{bmatrix} \tag{C.18}$$

Linearized Measurement model H matrix with augmented state

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{C.19}$$

D IMU noise and bias detection and parametrization

As the components of the IMU, gyroscopes and accelerometers, are subject to noise and bias errors, causing the drift in sensor output, a method to detect and analyze these is needed. The time domain analysis technique called Allan variance is suggested in [77] as a method to estimate the random walk and the bias instability appearing in gyroscope and accelerometer outputs. The random walk appearing in accelerometer and gyroscope output are called velocity random walk (VRW) and angle random walk (ARW) respectively. Following steps explain how the Allan variance and the Allan deviation, which is used to determine the noise and bias characteristics.

1. Divide the data set into at least nine bins with length t, fewer bins will have the result loosing its significance.
2. Average the data in all n bins, creating a vector of averages, $a(t)_1, a(t)_2, \ldots, a(t)_n$.
3. The Allan variance is calculated according to Eqn. D.1 below

$$A_{VAR}(t) = \frac{1}{2(n-1)} \sum i(a(t)_{i+1} - a(t)_i)^2 \tag{D.1}$$

4. The Allan deviation used for plotting is calculated by taking the square root of the Allan variance, see Eqn. D.2 below.

$$A_{DEV}(t) = \sqrt{A_{VAR}(t)} \tag{D.2}$$

By plotting the Allan deviation, A_{DEV}, as a function of time on a log-log scale different processes such as white noise and biases will appear in different regions with different gradients, allowing them to both be identified and parameterized directly from the plot. Random walks are caused by white noise and are depicted in Allan deviation plots as a -0.5 gradient slope, and the random walk measurement is derived by extracting the value at $t = 1$. Bias instability is depicted in Allan deviation plots as a horizontal region, where the minimum value is extracted [77]. More information about Allan variance can be found in [1], Annex C.