Institute of Flight System Dynamics
Technische Universität München
Prof. Dr.-Ing. Florian Holzapfel

Executed at Robotics and Automation Lab
Centre for Intelligent Information Processing Systems
University of Western Australia

Low-Cost Integrated Navigation System

Semesterarbeit

Franz Viertler

Matr.-No.: 2832700

01.08.2009

Advisors: Prof. Dr. rer. nat. habil. Thomas Bräunl

Dipl.-Ing. Benjamin Braun

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe und nur unter Verwendung der angegebenen Hilfsmittel angefertigt habe.

_____ _____

Datum Vorname Name

Abstract

In the context of the Renewable Energy Vehicle (REV) project at the University of Western Australia, a GPS-aided inertial navigation system has been developed. In the first part of this work the inertial navigation equations are derived. The inertial navigation solution is then combined with the GPS information to an integrated navigation system. Hence, an Extended Kalman Filter is presented, which is necessary to improve the inertial navigation solution due to low quality sensors.

The equations are implemented in Matlab for testing and validating the results which are also presented in this work. Finally everything is implemented on the EyeBot M6, a microcontroller system developed at the Centre for Intelligence Information Processing Systems, which is installed at one of the Renewable Energy Vehicles, the REV Eco, a plug-in electric conversion of a Hyundai Getz.

Table of Contents

List of Figures

Nomenclature

ADC	Analog Digital Converter
ASCII	American Standard Code for Information Interchange
B	Body Fixed Frame
DoF	Degrees of Freedom
ECI	Earth Centered Inertial Frame
ECEF	Earth Centered Earth Fixed Frame
EKF	Extended Kalman Filter
GRS 80	Geodetic Reference System 1980
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISA	Inertial System Assembly
LKF	Linearized Kalman Filter
MEMS	Micro-Electro-Mechanical-Systems
NED	North East Down Frame
NMEA	National Marine Electronics Association
REV	Renewable Energy Vehicle
RoBIOS	Robot Basic Input Output System
SAE	Society of Automotive Engineers International
WGS 84	World Geodetic System 1984

Mathematical notation (superscripts and subscripts)

$\dot{v} = \frac{\partial v}{\partial t}$	Derivation with respect to time
$\left(\vec{\mathbf{r}}^{\,R}\right)_E$	Position of the point R (Point of reference) described in the ECEF frame

$\left(\dot{\vec{\mathbf{r}}}^{R}\right)_{E}^{E}$ Velocity of the point R with respect to the ECEF frame, described in the ECEF frame

$\left(\ddot{\vec{\mathbf{r}}}^{R}\right)_{O}^{II}$ Acceleration of the point R with respect to the inertial frame, described in the NED frame

$\left(\vec{\boldsymbol{\omega}}^{IE}\right)_{E}$ Angular velocity between the ECI frame and the ECEF frame, described in the ECEF system

1 Introduction

1.1 Renewable Energy Vehicle Project

The Renewable Energy Vehicle project (REV) is an initiative to design and develop environmentally, sustainable technology for future transportation. Further developments in electrical motors and improvements in battery technologies make electrical vehicles more and more attractive for public use. In the REV project we want to demonstrate the feasibility of modern technology to convert a petrol car into an electrical vehicle. In the application environment many people are using their car only for driving to work and back, that means less than 80 km a day. This distance is now achievable with battery-based electrical cars, which can be charged over night. The use of electrical vehicles reduces the air pollution in large cities and by producing the energy with a photovoltaic system, like for this demonstration project, completely clean, emission free energy is used.

There is still the challenge to achieve the same performance as a conventional car has with a combustion engine, since people will always compare new technology to previous systems. We want to improve the disadvantages of electrical vehicles, namely the limited range, the long charging time and lower performance. Hence an economy car, the Hyundai Getz, see Figure 1.1, has already been converted in last year's project. To develop the performance of electric vehicles, a Lotus Elise, see Figure 1.2, and a racing car of the formula SAE are going to be converted until the end of this year (2009).

Figure 1.1: Economy Car, Hyundai Getz

Figure 1.2: Performance Car, Lotus Elise

The main tasks in converting a car into electrical systems are of course to replace the combustion engine and the fuel tank by an electrical engine and batteries. Hence we need a motor controller and battery management system for charging and discharging. Besides that, we also require a new instrumentation and for our research we design a black box to collect all necessary data for evaluating the performance of the electrical vehicle.

1.2 Black Box and Navigation System

The primary function of the black box is to log important data about the state of the car and about the states of the electrical systems, similar to a black box in an aircraft. Hence we installed several new digital and analog sensors to measure the state of charge of the main batteries, safety critical systems and the acceleration of the car. We also included a GPS-Receiver to locate the car and a mobile broadband modem is projected for a connection to the internet.

We also use the black box to display the import information for the driver on a touch screen. The mobile broadband connection will be used to get online information about the car during a test drive. The main part of the black box consists of the microcontroller system “EyeBot M6” from the robotic and automation research at the UWA[1]. It has many digital I/Os, analog inputs, serial and USB connections to collect the data and log it on an USB-stick. A Linux operation system is installed and “RoBIOS Library Functions” are developed to facilitate software development. A large touch screen provides the user interface.

The main focus of this thesis is on the integrated navigation system. After a more detailed description of the hardware setup in chapter 1.3, a brief introduction to the Global Positioning System GPS is given in chapter 1.4, before we derive the inertial navigation equations in Chapter 2. In chapter 3 the problem of measurement noise is specified and we discuss methods to align the inertial sensors. After that, in Chapter 4, we add the information of the GPS-Receiver to the calculated position of the inertial navigation equations. For this we use a Kalman Filter, which bases on the state space description of dynamic systems and is the optimal estimator for linear systems. After a brief introduction to the probability theory of random signals and the

[1] The University of Western Australia

basics of the Kalman Filter we derive the linearized Kalman Filter and the extended Kalman Filter version to apply it to our nonlinear system. Chapter 5 shows the results of the integrated navigation solution and compares the two Filter versions. It also mentions the in-car visualization with “Open Street Maps” as a source of maps, before finally a conclusion is given in chapter 6.

1.3 Hardware Setup

An inertial measurement unit is structured as shown in Figure 1.3. It includes the inertial system assembly (ISA) which consists of the accelerometers and gyroscopes. The IMU has already a processor with analog-digital converters (ADC) and the required software to provide digital data for the INS equations ([4] Dorobantu).

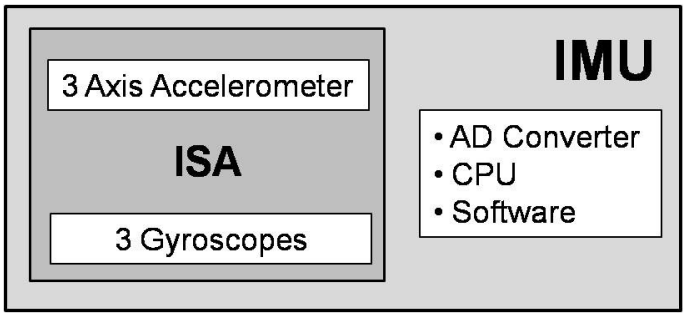

Figure 1.3: IMU Structure

In our project we are working with the Atomic six degree of freedom IMU from “SparkFun” ([17] Atomic IMU). It uses a MEMS three axes accelerometer ([8] MMA7260Q) and three MEMS single axis gyroscopes ([18] LISY300AL) shown in Figure 1.4.

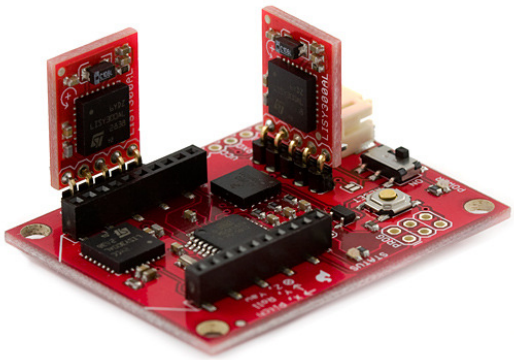

Figure 1.4: Atomic six degree of freedom IMU ([17] Atomic IMU)

The principle of operation of a low-cost micro-machined capacitive accelerometer is the variation of the gap between a fixed electrode and a moving electrode, the proof mass. The capacitor's value will change with the distance of the electrodes Eq. (1.1),

$$C = \varepsilon \cdot \frac{A}{D} \tag{1.1}$$

Figure 1.5 shows a simplified physical model. The advantages of that kind of accelerometer are: a simple structure, high performance, low temperature sensitivity and low power dissipation.

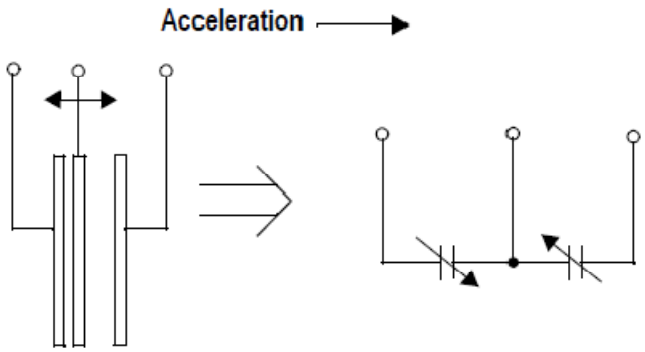

Figure 1.5: Simplified Transducer physical model ([8] MMA7260Q)

MEMS gyros are known as vibrating structure gyros. The physical principle is that a linearly vibrating object at a known frequency, changes the frequency after rotating it because of the Coriolis acceleration. Measuring and interpreting this frequency modification yields the angular velocity ([19] Sukkarieh, [21] Titterton/Weston). For more information about the sensors see the datasheets ([17] Atomic IMU, [8] MMA7260Q, [18] LISY300AL). The Atomic IMU is designed for mobile applications and therefore it has an XBee socket to connect it over an XBee antenna to a wireless network. We are using an "UART to RS232 Interface Card" (Figure 1.6) from SparkFun to connect the IMU via a serial cable to our microcontroller system, because in a car it is not necessary to use a wireless connection.

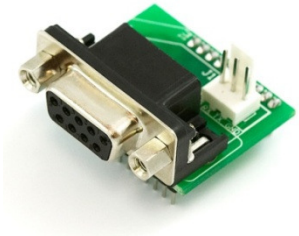

Figure 1.6: UART to RS232 Interface Card

To read out the sensor signals and to calculate the navigation equations we use the “EyeBot M6” microcontroller system ([1] Bräunl). As already mentioned the main advantages are numerous digital and analog inputs, the serial and USB ports and a large touch screen for the visualization, see Figure 1.7 and Figure 1.8. It has an ARM9 processor which provides us enough processing power. For more information see the documentation ([1] Bräunl, [6] EyeBotM6).

Figure 1.7: EyeBotM6 Front

Figure 1.8: EyeBotM6 Rear

For detailed information about the IMU settings and for the serial connection see appendix A.1.

1.4 Global Positioning System

The Global Positioning System (GPS) is used in this work as additional position information to the inertial navigation system. The GPS aided navigation system is able to limit the growing error of the stand alone low-cost inertial navigation system.

GPS needs 24 satellites in six orbits to provide the minimum of four required satellites in the view of the user anywhere on the globe, see Figure 1.9. Four satellites are needed to obtain the three unknown in position and to correct the receiver clock bias.

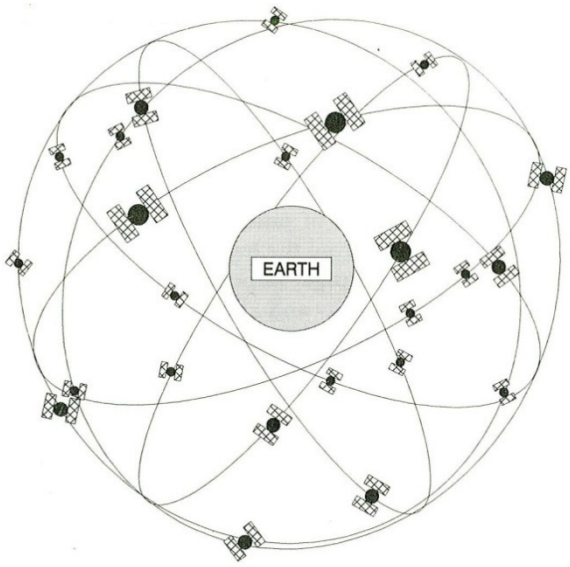

Figure 1.9: GPS satellites around the globe ([21] Titteton)

The principle of the position fixing is to determine the flight time of the signal from the satellite to the receiver. The range between the receiver and the satellite is then given by multiplying the time difference with the speed of light. Very accurate clocks are needed and hence a fourth satellite is required to correct the receiver clock bias ([19] Sukkarieh, [22] Wagner).

Besides the receiver clock bias there are a few other error sources which can cause a loss of accuracy in the GPS navigation. In Table 1-1 some error sources are listed in combination with their estimated range error for a stationary GPS receiver ([22] Wagner).

Error source	Estimated fault tolerance [m]
Satellite clock error	3.1
Ephemeris error	2.6
Ionosphere delay error	6.4
Troposphere delay error	0.4
Receiver noise/quantization error	2.4
Receiver interchannel bias	0.6
Multipath	3.1

Table 1-1: Error source and range error

We are using the standard USB-GPS-receiver “BU-353 (SiRF III) from “GlobalSat Technology Cooperation”. For detailed information about the GPS-receiver settings see appendix A.2.

2 Inertial Navigation Equations

Accelerometers and gyroscopes do not measure directly the position. They only measure the linear acceleration and angular velocity. Since they measure only the first or second derivative we have to integrate the signals to achieve the velocity, position and attitude of the vehicle ([16] Skog/Händel). The inertial navigation is a kind of “dead reckoning” method because we cannot measure the initial state. Hence we need external information to initialize the INS ([22] Wagner). The accuracy of an INS depends on the quality of the inertial sensors. There is a large loss of accuracy over a longer period of time due to the double integration of the acceleration, including also the measurement error, to obtain the position. Submarines use inertial navigation with high accurate sensors since they should be able to navigate without external sources, like GPS for example. INS is also used in aircrafts because for a high dynamic system, like an aircraft, an accurate navigation system is desired, which is even operating without a satellite signal or between the GPS sampling time steps. The conception of an INS is shown in Figure 4.2.

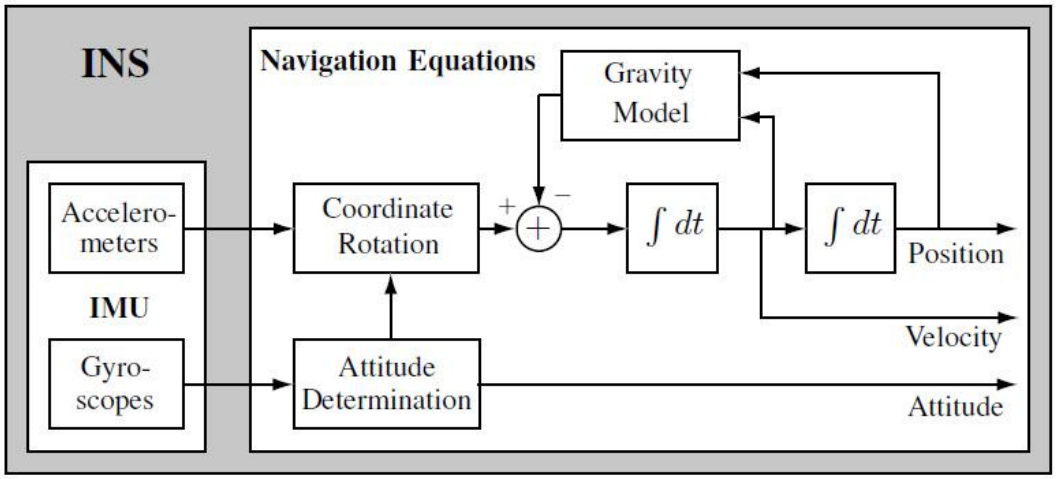

Figure 2.1: INS information flow ([16] Skog/Händel)

The angular velocities of the gyroscopes are used to determine the attitude of the vehicle and thus the direction of the linear accelerometer axes with respect to the navigation coordinate frame. Subtracting the earth gravity of the accelerometer signal and integrating it twice yields the position. The motion of a vehicle in-plane for example has three degrees of freedom, two linear and one rotational motion. Hence we need two accelerometers and one gyroscope. The motion in the real world has six

degrees of freedom, three linear and three rotational motions. Hence, the Atomic six degree of freedom IMU ([17] Atomic IMU) consists of a three axes linear accelerometer and three gyroscopes. It is also possible to use only accelerometers to measure all six degrees of freedom. If we combine a minimum of six single linear accelerometers in a special cube configuration, we can build a gyroscope-free six degree of freedom IMU. For more information see ([20] Tan/Park).

2.1 Coordinate Transformations

In comparison to an expensive gimballed IMU, where the accelerometers are mounted on platform which is always aligned to the navigation frame ([19] Sukkarieh, [22] Wagner), our low-cost MEMS sensors are mounted in the vehicle aligned to the body frame. Hence the sensors require a higher bandwidth and dynamic range. For the transformation of the accelerations in the body frame to the navigation frame we derive the strap down navigation equations.

The following coordinate frames are used for the strap down navigation equations:

- Earth Centered Inertial Frame - ECI
- Earth Centered Earth Fixed Frame - ECEF
- North East Down Frame - NED
- Body Fixed Frame – B

Figure 2.2 shows the inertial frame, which is applied for using Newton's laws, because for the vehicle dynamics it is regarded as not accelerated.

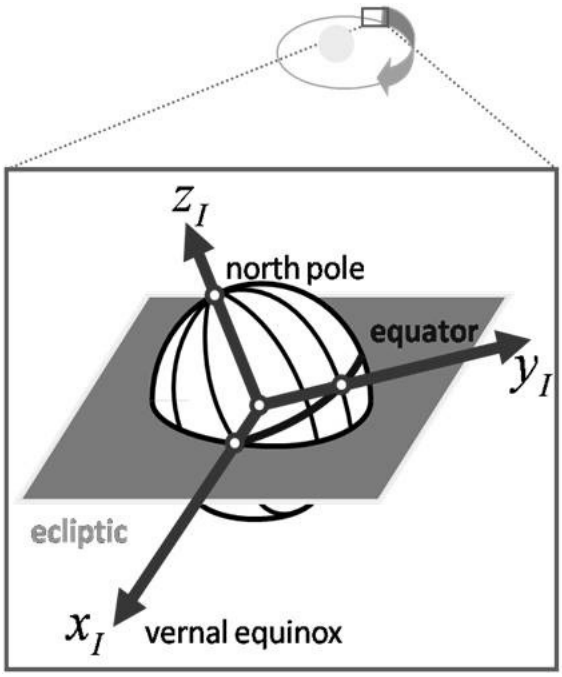

Figure 2.2: Earth Centered Inertial Frame ([10] Holzapfel)

ECI – Earth Centered Inertial Frame	
Index	I
Role	Inertial system
Origin	Geocenter
Translation	Around the sun
Rotation	No
x-Axis	In the equatorial plane, pointing at vernal equinox
y-Axis	In the equatorial plane, building a right hand system with x-axis and z-axis
z-Axis	Earth rotation axis

Table 2-1: ECI – Earth Centered Inertial Frame

In the Earth Centered Earth Fixed Frame, shown in Figure 2.3 the position of the vehicle is noted.

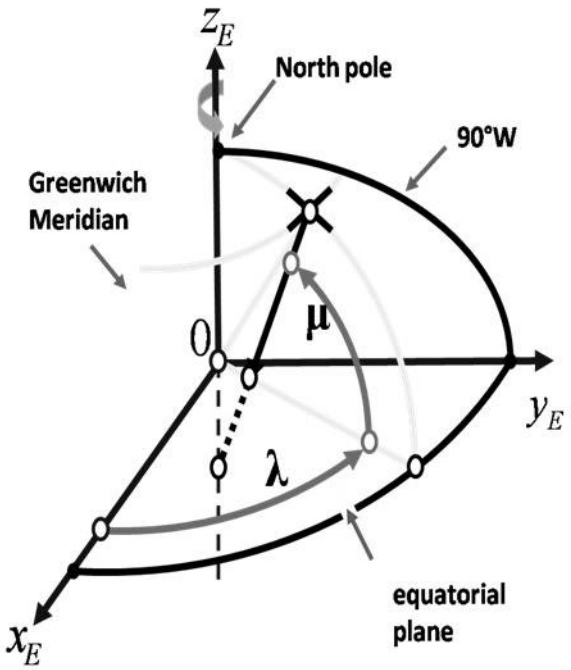

Figure 2.3: Earth Centered Earth Fixed Frame ([10] Holzapfel)

ECEF - Earth Centered Earth Fixed Frame	
Index	E
Role	Position system
Origin	Geocenter
Translation	Moves with ECI system
Rotation	Earth rotation
x-Axis	In the equatorial plane, pointing at Greenwich Meridian
y-Axis	In the equatorial plane, building a right hand system with x-axis and z-axis
z-Axis	Earth rotation axis

Table 2-2: ECEF - Earth Centered Earth Fixed Frame

The angular velocity (2.1) from the Earth Centered Earth Fixed Frame relative to the Earth Centered Inertial Frame is the earth rotation:

$$\left(\vec{\omega}^{IE}\right)_I = \begin{pmatrix} 0 \\ 0 \\ \approx \dfrac{2\pi}{24hours} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 7{,}292115 \cdot 10^{-5}\, rad/s \end{pmatrix} \tag{2.1}$$

Figure 2.4 shows the orientation system, the North East Down Frame.

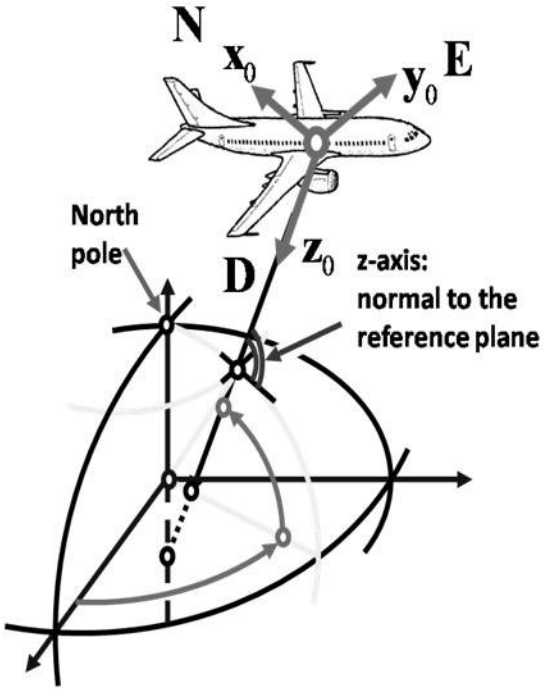

Figure 2.4: North East Down Frame ([10] Holzapfel)

The main task of the North East Down Frame is to note the information about the orientation of the vehicle. In this work we use the NED frame as navigation frame for deriving the navigation equations.

NED - North East Down Frame	
Index	O
Role	Orientation system
Origin	Point of reference of the vehicle
Translation	Moves with the point of reference
Rotation	Transport rate
x-Axis	Parallel to the geoid-surface, pointing at the geographic north pole
y-Axis	Parallel to the geoid-surface, pointing east direction
z-Axis	Pointing down, normal to the geoid-surface

Table 2-3: NED - North East Down Frame

The transformation matrix (2.2) from the Earth Centered Earth Fixed Frame to the North East Down Frame is:

$$\mathbf{M}_{OE} = \begin{pmatrix} -\sin\mu\cos\lambda & -\sin\mu\sin\lambda & \cos\mu \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\mu\cos\lambda & -\cos\mu\sin\lambda & -\sin\mu \end{pmatrix} \tag{2.2}$$

And from the North East Down Frame to the Earth Centered Earth Fixed Frame:

$$\mathbf{M}_{EO} = \mathbf{M}_{OE}^{T} \tag{2.3}$$

The angular velocity (2.4) from the North East Down Frame relative to the Earth Centered Earth Fixed Frame becomes:

$$\left(\vec{\omega}^{EO}\right)_O = \begin{pmatrix} \dot{\lambda}\cos\mu \\ -\dot{\mu} \\ -\dot{\lambda}\sin\mu \end{pmatrix} \tag{2.4}$$

The Body Fixed Frame is shown in Figure 2.5. It is used as notation frame for all forces and moments which affect on the vehicle. That means our inertial sensor input is noted in the B-frame.

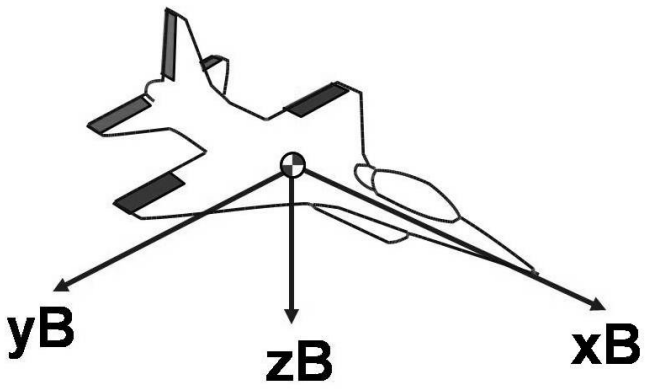

Figure 2.5: Body Fixed Frame ([10] Holzapfel)

B - Body Fixed Frame	
Index	B
Role	For notation of forces and moments
Origin	Point of reference of the vehicle
Translation	Moves with the Point of reference
Rotation	Rotate with the vehicle
x-Axis	Pointing at the nose of the vehicle, in the xz-plane of symmetry
y-Axis	Pointing starboard, building an orthogonal system
z-Axis	Pointing down in the plane of symmetry

Table 2-4: B - Body Fixed Frame

The transformation matrix (2.5) from the North East Down Frame to the Body Fixed Frame is:

$$\mathbf{M}_{BO} = \begin{pmatrix} \cos\Psi\cos\Theta & \sin\Psi\cos\Theta & -\sin\Theta \\ \cos\Psi\sin\Theta\sin\Phi - \sin\Psi\cos\Phi & \sin\Psi\sin\Theta\sin\Phi - \cos\Psi\cos\Phi & \cos\Theta\sin\Phi \\ \cos\Psi\sin\Theta\cos\Phi + \sin\Psi\sin\Phi & \sin\Psi\sin\Theta\cos\Phi - \cos\Psi\sin\Phi & \cos\Theta\cos\Phi \end{pmatrix} \tag{2.5}$$

Where Ψ, Θ, Φ are the three Euler angle. And the transformation matrix (2.6) from the Body Fixed Frame to the North East Down Frame is:

$$\mathbf{M}_{OB} = \mathbf{M}_{BO}^{T} \tag{2.6}$$

The angular velocity (2.7) from the Body Fixed Frame relative to the North East Down Frame becomes:

$$\left(\vec{\omega}^{OB}\right)_B = \begin{pmatrix} \dot{\Phi} - \dot{\Psi}\sin\Theta \\ \dot{\Theta}\cos\Phi + \dot{\Psi}\sin\Phi\cos\Theta \\ -\dot{\Theta}\sin\Phi + \dot{\Psi}\cos\Phi\cos\Theta \end{pmatrix} \tag{2.7}$$

We also use the World Geodetic System 1984 definition ([10] Holzapfel, [22] Wagner) and consider that the earth is flattened and not completely spherical. Geodetic coordinates are used in form of two angles, longitude λ and latitude μ, and the altitude h in meters, shown in Figure 2.6:

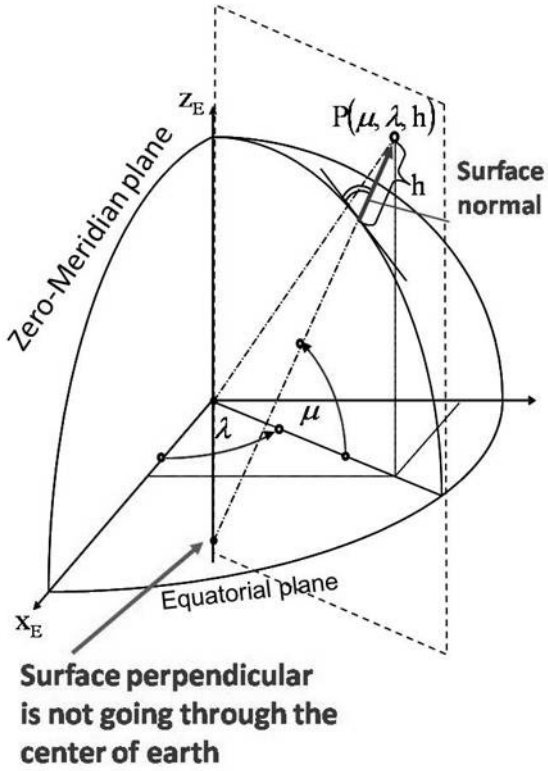

Figure 2.6: WGS 84 – World Geodetic System 1984 ([10] Holzapfel)

To describe the reference ellipsoid the major and minor semi axes are given by:

Semi-major: $a = 6378137\ .0m$

Semi-minor: $b = 6356752\ .3142\ m$

The first and second eccentricity with the flattening is given by:

Flattening: $f = \frac{a-b}{a} = 0.0033528$

First eccentricity: $e = \sqrt{f(2-f)} = 0.0818191$

Second eccentricity: $(e')^2 = \frac{a^2 - b^2}{a^2} = 0.0066944$

Figure 2.7 shows the difference between the geoid, topographic and the ellipsoid reference surface.

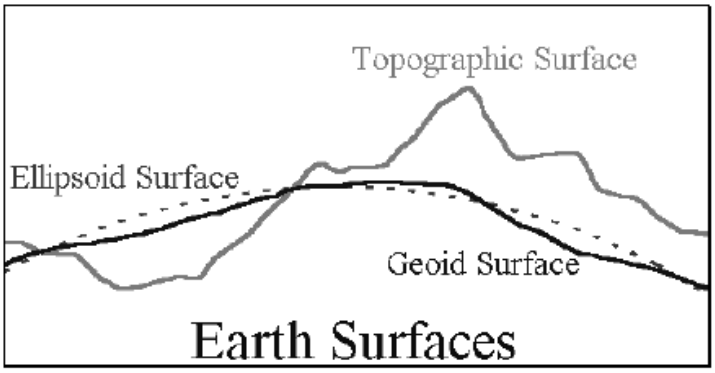

Figure 2.7: Reference Ellipsoid ([22] Wagner)

Since the earth is regarded as an ellipsoid, the radii differ, dependent on the latitude. We obtain the meridian radius of curvature with Eq. (2.8):

$$N_\mu = \frac{a}{\sqrt{1 - e^2 \sin^2 \mu}} \tag{2.8}$$

And the transverse radius of curvature with Eq. (2.9):

$$M_\mu = a\frac{1-e^2}{\left(1-e^2 \sin^2 \mu\right)^{3/2}} = N_\mu \cdot \frac{1-e^2}{1-e^2 \sin^2 \mu} \tag{2.9}$$

The transformation of geodetic to Cartesian coordinates is given by the following set of equations (2.10), ([19] Sukkarieh, [22] Wagner):

$$\begin{aligned} x &= (N + h)\cos\ \mu \cos\ \lambda \\ y &= (N + h)\cos\ \mu \sin\ \lambda \\ z &= \left[N\left(1 - e^2\right) + h\right]\sin\ \mu \end{aligned} \tag{2.10}$$

On the other hand, the transformation of Cartesian coordinates into geodetic coordinates is only possible with iteratively calculations or with the following approximation (2.12). For this purpose we have to calculate some coefficients first (2.11):

$$\begin{aligned} p &= \sqrt{x^2 + y^2} \\ t &= \arctan\left(\frac{z \cdot a}{p \cdot b}\right) \end{aligned} \tag{2.11}$$

$$\begin{aligned} \mu &= \arctan\left(\frac{z + (e')^2 b \sin^3 t}{p - e^2 \cos^3 t}\right) \\ \lambda &= \arctan\left(\frac{y}{x}\right) \\ h &= \frac{p}{\cos \mu} - N \end{aligned} \tag{2.12}$$

For our application we can assume that the meridian radius of curvature is constant, since the latitude doesn't change significantly. And if we neglect the altitude above sea level by regarding only a two-dimensional model, we can use the following equations for the latitude and longitude solution without iterations ([5] Koordinaten Transformation):

$$\begin{aligned} \mu &= \arctan\left(\frac{z / p}{1 - e^2}\right) \\ \lambda &= \arctan\left(\frac{y}{x}\right) \end{aligned} \tag{2.13}$$

2.2 Strap down navigation equations

For using Newton's laws we have to derive the acceleration with respect to the inertial frame. Since the earth is rotating, we have to consider the Coriolis theorem. Thus the velocity with respect to the inertial frame is the velocity with respect to the earth centered earth fixed frame, which is the ground speed, plus the Coriolis term, see Eq. (2.14).

$$\left(\frac{d}{dt}\right)^I \left(\vec{\mathbf{r}}^R\right) = \left(\vec{\mathbf{v}}^R\right)^I = \left(\vec{\mathbf{v}}^R\right)^E + \left(\vec{\boldsymbol{\omega}}^{IE}\right) \times \left(\vec{\mathbf{r}}^R\right) \tag{2.14}$$

Where "R" is the point of reference of the vehicle, $\left(\vec{\mathbf{r}}^R\right)$ is the position vector, $\left(\vec{\mathbf{v}}^R\right)^E$ is the velocity with respect to the ECEF frame and $\left(\vec{\boldsymbol{\omega}}^{IE}\right)$ is the earth rotation. The next step is to derive the velocity again with respect to the inertial frame to achieve the acceleration, see Eq. (2.15).

$$\left(\frac{d}{dt}\right)^I \left(\vec{\mathbf{v}}^R\right)^I = \left(\dot{\vec{\mathbf{v}}}^R\right)^{EE} + \left(\vec{\boldsymbol{\omega}}^{IE}\right)\times\left(\vec{\mathbf{v}}^R\right)^E + \underbrace{\left(\dot{\vec{\boldsymbol{\omega}}}^{IE}\right)^I}_{=0}\times\left(\vec{\mathbf{r}}^R\right) + \left(\vec{\boldsymbol{\omega}}^{IE}\right)\times\left(\vec{\mathbf{v}}^R\right)^I$$
$$= \left(\dot{\vec{\mathbf{v}}}^R\right)^{EE} + 2\cdot\left(\vec{\boldsymbol{\omega}}^{IE}\right)\times\left(\vec{\mathbf{v}}^R\right)^E + \left(\vec{\boldsymbol{\omega}}^{IE}\right)\times\left[\left(\vec{\boldsymbol{\omega}}^{IE}\right)\times\left(\vec{\mathbf{r}}^R\right)\right] \tag{2.15}$$

The earth rotation is considered to be constant, thus $\left(\dot{\vec{\boldsymbol{\omega}}}^{IE}\right)^I = 0$. To derive the equations with respect to the NED frame we have to include another term considering the Coriolis acceleration, Eq. (2.16).

$$\left(\dot{\vec{\mathbf{v}}}^R\right)^{EE} = \left(\dot{\vec{\mathbf{v}}}^R\right)^{EO} + \left(\vec{\boldsymbol{\omega}}^{EO}\right)\times\left(\vec{\mathbf{v}}^R\right)^E \tag{2.16}$$

This results in Eq. (2.17):

$$\left(\dot{\vec{\mathbf{v}}}^R\right)^{II} = \left(\dot{\vec{\mathbf{v}}}^R\right)^{EO} + \left(2\cdot\left(\vec{\boldsymbol{\omega}}^{IE}\right) + \left(\vec{\boldsymbol{\omega}}^{EO}\right)\right)\times\left(\vec{\mathbf{v}}^R\right)^E + \left(\vec{\boldsymbol{\omega}}^{IE}\right)\times\left[\left(\vec{\boldsymbol{\omega}}^{IE}\right)\times\left(\vec{\mathbf{r}}^R\right)\right] \tag{2.17}$$

Now we can use Newton's first and second law, the fundamentals of the inertial navigation, see Eq. (2.18):

$$\vec{\mathbf{F}} = \frac{d}{dt}\left(m\vec{\mathbf{v}}\right) \tag{2.18}$$

The mass of the vehicle is assumed to be constant while moving and the external forces acting on the vehicle are the gravitation force and the specific force measured by the IMU.

$$\left(\dot{\vec{\mathbf{v}}}^R\right)^{II} = \frac{\vec{\mathbf{G}}}{m} + \frac{\vec{\mathbf{F}}}{m} = \left(\vec{\mathbf{g}}\right) + \left(\vec{\mathbf{f}}\right) \tag{2.19}$$

Including Eq. (2.17) into Eq. (2.19) yields Eq. (2.20):

$$\left(\dot{\vec{\mathbf{v}}}^R\right)^{EO} = \left(\vec{\mathbf{f}}\right) - \left(2\cdot\left(\vec{\boldsymbol{\omega}}^{IE}\right) + \left(\vec{\boldsymbol{\omega}}^{EO}\right)\right)\times\left(\vec{\mathbf{v}}^R\right)^E + \left(\vec{\mathbf{g}}\right) - \left(\vec{\boldsymbol{\omega}}^{IE}\right)\times\left[\left(\vec{\boldsymbol{\omega}}^{IE}\right)\times\left(\vec{\mathbf{r}}^R\right)\right] \tag{2.20}$$

The final velocity propagation equation in the NED frame is given by Eq. (2.21):

$$\left(\dot{\vec{\mathbf{v}}}^{R}\right)_{O}^{EO} = \underbrace{\mathbf{M}_{OB}\cdot\left(\vec{\mathbf{f}}\right)_{B}}_{\substack{specific\\forces}} - \underbrace{\left(2\cdot\mathbf{M}_{OE}\left(\vec{\boldsymbol{\omega}}^{IE}\right)_{E}+\left(\vec{\boldsymbol{\omega}}^{EO}\right)_{O}\right)\times\left(\vec{\mathbf{v}}^{R}\right)_{O}^{E}}_{\substack{coriolis\\acc.}} +$$
$$+ \underbrace{\left(\vec{\mathbf{g}}\right)_{O}}_{\substack{gravitation}} - \underbrace{\mathbf{M}_{OE}\cdot\left(\left(\vec{\boldsymbol{\omega}}^{IE}\right)_{E}\times\left[\left(\vec{\boldsymbol{\omega}}^{IE}\right)_{E}\times\left(\vec{\mathbf{r}}^{R}\right)_{E}\right]\right)}_{\substack{centrifugal\\acc.}} \tag{2.21}$$

The vectors in Eq. (2.21) are:

The acceleration of the vehicle with respect to the NED frame… $\left(\dot{\vec{\mathbf{v}}}^{R}\right)_{O}^{EO}$

The velocity in north, east and down components…… $\left(\vec{\mathbf{v}}^{R}\right)_{O}^{E}=\begin{pmatrix}u^{R}\\v^{R}\\w^{R}\end{pmatrix}_{O}^{E}$

The position in the ECEF frame with Cartesian coordinates…… $\left(\vec{\mathbf{r}}^{R}\right)_{E}=\begin{pmatrix}x\\y\\z\end{pmatrix}_{E}$

The specific force vector in the B frame…… $\left(\vec{\mathbf{f}}\right)_{B}=\begin{pmatrix}f_{x}\\f_{y}\\f_{z}\end{pmatrix}_{B}$

The gravitation vector in the NED frame…… $\left(\vec{\mathbf{g}}\right)_{O}$

The position equations in the NED frame considering the WGS 84 definition is given by Eq. (2.22):

$$\begin{pmatrix}\dot{\lambda}\\\dot{\mu}\\\dot{h}\end{pmatrix}=\begin{pmatrix}\dfrac{v^{R}}{(N_{\mu}+h)\cos\mu}\\\dfrac{u^{R}}{M_{\mu}+h}\\-w^{R}\end{pmatrix}_{O}^{E} \tag{2.22}$$

u, v and w are again the north, east and down velocities. While N_μ and M_μ are the meridian and transverse radius of curvature depending on the latitude.

Furthermore the attitude propagation equations are dependent on the gyroscope measurements. The gyroscopes are measuring the angular velocities of the body frame axes with respect to the inertial system. To get the angular velocities with respect to the NED frame we have to subtract the earth rotation and the transport rate of the measurement, see Eq. (2.23):

$$\left(\vec{\boldsymbol{\omega}}^{OB}\right)_B = \underbrace{\left(\vec{\boldsymbol{\omega}}^{IB}\right)_B}_{\substack{Measurement\\ Gyros}} - \mathbf{M}_{BO} \cdot \left(\underbrace{\mathbf{M}_{OE} \cdot \left(\vec{\boldsymbol{\omega}}^{IE}\right)_E}_{\substack{earth\\ rotation}} + \underbrace{\left(\vec{\boldsymbol{\omega}}^{EO}\right)_O}_{\substack{transport\\ rate}} \right) \tag{2.23}$$

Then we can use the following attitude equations (2.24) to calculate the propagation of the Euler angles:

$$\begin{pmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{pmatrix} = \begin{pmatrix} 1 & \tan\Theta\sin\Phi & \tan\Theta\cos\Phi \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \dfrac{\sin\Phi}{\cos\Theta} & \dfrac{\cos\Phi}{\cos\Theta} \end{pmatrix} \cdot \left(\vec{\boldsymbol{\omega}}^{OB}\right)_B \tag{2.24}$$

There are more options to describe the attitude of a vehicle. The Quaternion representation for example uses four variables to specify the attitude of a vehicle and avoids the singularity problems of the Euler description. In this work we use the Euler representation for a better comprehension.

For our application, using the integrated navigation system in a car, we can make some simplifications to the inertial navigation equations to reduce the computational effort. The Coriolis term depends on the velocity of the car. Since the velocity of a car is much smaller than for example an aircraft or a missile we can neglect the Coriolis term and we obtain Eq. (2.25):

$$\left(\dot{\vec{\mathbf{v}}}^{R}\right)_{O}^{EO} = \underbrace{\mathbf{M}_{OB}\cdot\left(\vec{\mathbf{f}}\right)_{B}}_{\substack{specific\\forces}} + \underbrace{\left(\vec{\mathbf{g}}\right)_{O}}_{gravitation} \underbrace{-\mathbf{M}_{OE}\cdot\left(\left(\vec{\boldsymbol{\omega}}^{IE}\right)_{E}\times\left[\left(\vec{\boldsymbol{\omega}}^{IE}\right)_{E}\times\left(\vec{\mathbf{r}}^{R}\right)_{E}\right]\right)}_{\substack{centrifugal\\acc.}} \tag{2.25}$$

The gravitation and the centrifugal acceleration are often combined to the local gravity vector $\vec{\mathbf{g}}_l$, see Eq. (2.26) ([21] Titterton/Weston). In our application we can use a constant value for the gravity, because the vehicle is traveling only small distances.

$$\vec{\mathbf{g}}_l = \underbrace{\vec{\mathbf{g}}}_{gravitation} \underbrace{-\left(\vec{\boldsymbol{\omega}}^{IE}\right)\times\left[\left(\vec{\boldsymbol{\omega}}^{IE}\right)\times\left(\vec{\mathbf{r}}^{R}\right)\right]}_{\substack{centrifugal\\acc.}} \tag{2.26}$$

A closed equation, called Somigliana's formula, is used to calculate the local gravity with latitude correction ([3] WGS 84, [7] Featherstone/Dentith). It uses the gravitation at the equator and at the pole. After a transformation for numerical purposes it yields Eq. (2.27):

$$\gamma = \gamma_a \frac{1 + k\sin\mu}{\sqrt{1 - e^2\sin^2\mu}} \tag{2.27}$$

The constant numerical values of the Geodetic Reference System 1980 (GRS 80) are:

Normal gravity at the equator: $\gamma_a = 9.7803267715\ \frac{m}{s^2}$

Normal gravity at the equator: $\gamma_b = 9.832\ 186\ 368\ 5\ \frac{m}{s^2}$

Square of the first eccentricity: $e^2 = 0.006\ 694\ 380\ 022\ 90$

Normal gravity constant: $k = \frac{b\cdot\gamma_b}{a\cdot\gamma_a} - 1 = 0.001\ 931\ 851\ 353$

We could also use an altitude depending gravity model. But for our application we are neglecting the altitude in the calculation of the gravity, because it doesn't change significantly in the application area.

After including the simplifications the velocity equations are given by Eq. (2.28):

$$\left(\dot{\vec{\mathbf{v}}}^{R}\right)_{O}^{EO} = \underbrace{\mathbf{M}_{OB} \cdot \left(\vec{\mathbf{f}}\right)_{B}}_{\substack{specific \\ forces}} + \underbrace{\left(\vec{\boldsymbol{\gamma}}\right)_{O}}_{gravity} \tag{2.28}$$

Besides the simplification of the coordinate transformation due to constant values for the meridian and transverse radius of curvature, N_{μ} and M_{μ}, we can also assume that the transport rate Eq. (2.29) depending on the velocity of the car is very small and can be neglected.

$$\left(\vec{\omega}^{EO}\right)_{O} = \begin{pmatrix} \dfrac{v^{R}}{(N_{\mu}+h)} \\ -\dfrac{u^{R}}{M_{\mu}+h} \\ -\dfrac{v^{R}\tan\mu}{(N_{\mu}+h)} \end{pmatrix} \tag{2.29}$$

The earth rotation is very small in comparison to the measurement errors of the low-cost gyros, we are using. Thus we can neglect it as well. Hence the measured turn rates are directly the angular velocities of the B frame with respect to the NED frame and the attitude propagation depends only on the prior attitude and current measurement of the gyros. What we are losing is the so called Schuler damping. It can bound the error of the INS. An error in the attitude for example causes incorrect calculation of the acceleration and therefore leads to an error in the velocity. The velocity is then used to calculate the transport rate and with the transport rate it is feed back to the attitude propagation ([19] Sukkarieh, [22] Wagner). Finally the bounded position error is only useful, if the bias of the sensors is small enough, like it is of laser gyros in aircrafts for example, but not with the bias of our low-cost MEMS gyros.

3 Inertial Sensor Characteristics

In this section we are describing the output signal of the inertial sensors and what we have to do for using it as input of the INS and later the Kalman Filter to estimate the errors. Also the alignment and calibration of the IMU is described.

3.1 Measurement Noise

Figure 3.1 and Figure 3.4 show some raw data of the inertial sensors to get an impression of the noisy signals we are getting. To achieve better results and for reducing the computational effort a simple kind of low pass filter is implemented, the so-called over-sampling. Hence the mean over ten readings is computed, see Figure 3.2 and Figure 3.5. Our navigation equations are then calculated only ten times a second instead of hundred times a second, by using a 100 Hz sampling frequency of the IMU.

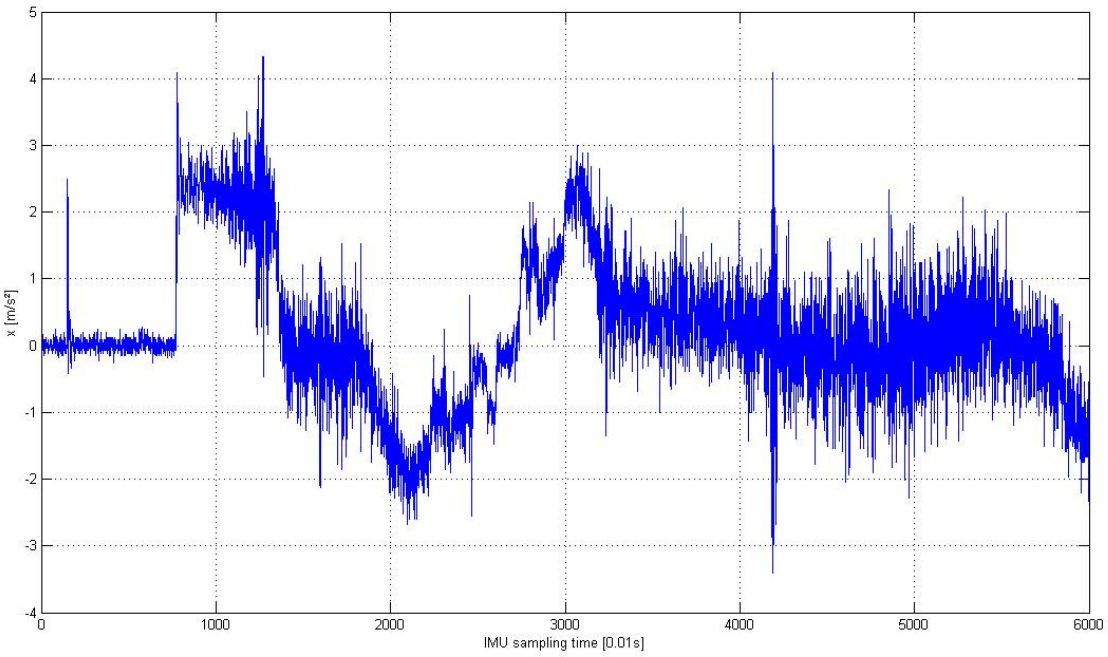

Figure 3.1: x-axis accelerometer measurement of 60s

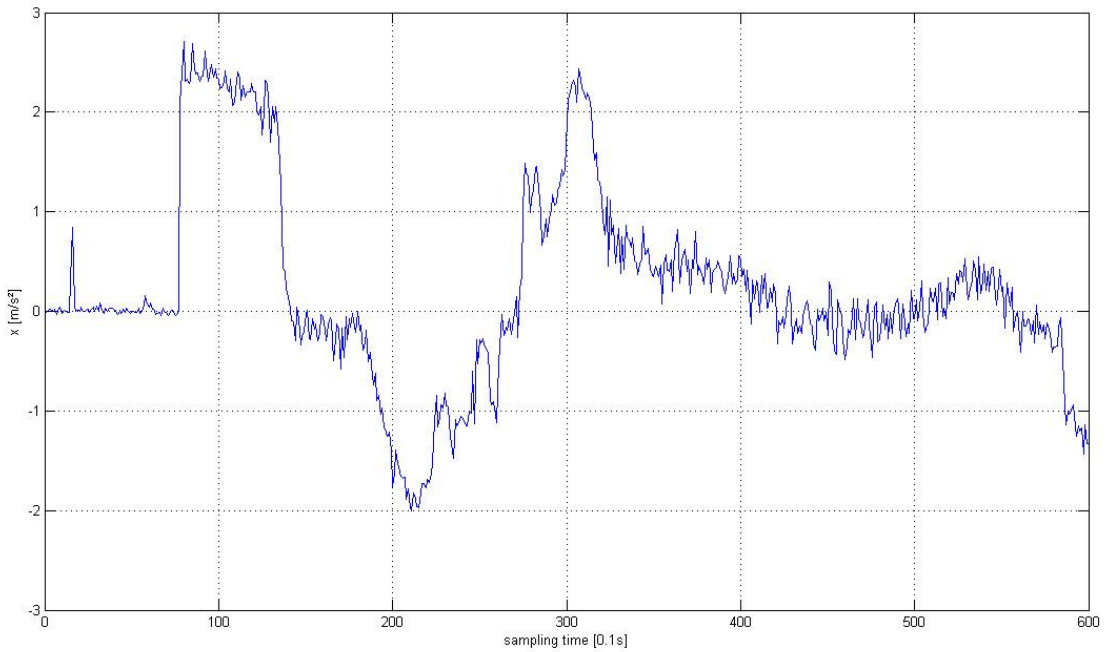

Figure 3.2: Mean x-axis acceleration measurement of 60s

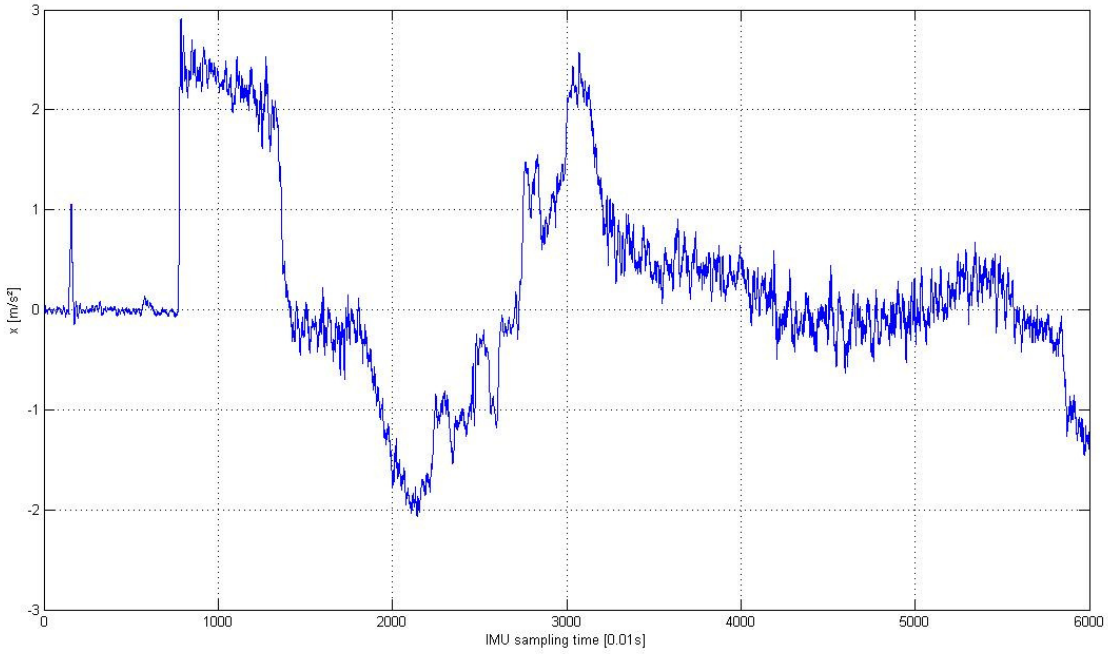

Figure 3.3: x-axis acceleration measurement of 60s (moving average)

If we want to keep the calculation frequency at the same level as the sampling frequency, we can use a type of finite impulse response filter, in statistics called a *moving average*. The moving average is obtained by calculating the mean over the last ten samplings, see Eq. (3.1):

$$Moving \quad Average \quad = \frac{s_N + s_{N-1} + \cdots + s_{N-n}}{n} \tag{3.1}$$

Where N is the current sampling and n is the number of regarded samplings in the past, in our example ten, see Figure 3.3 and Figure 3.6.

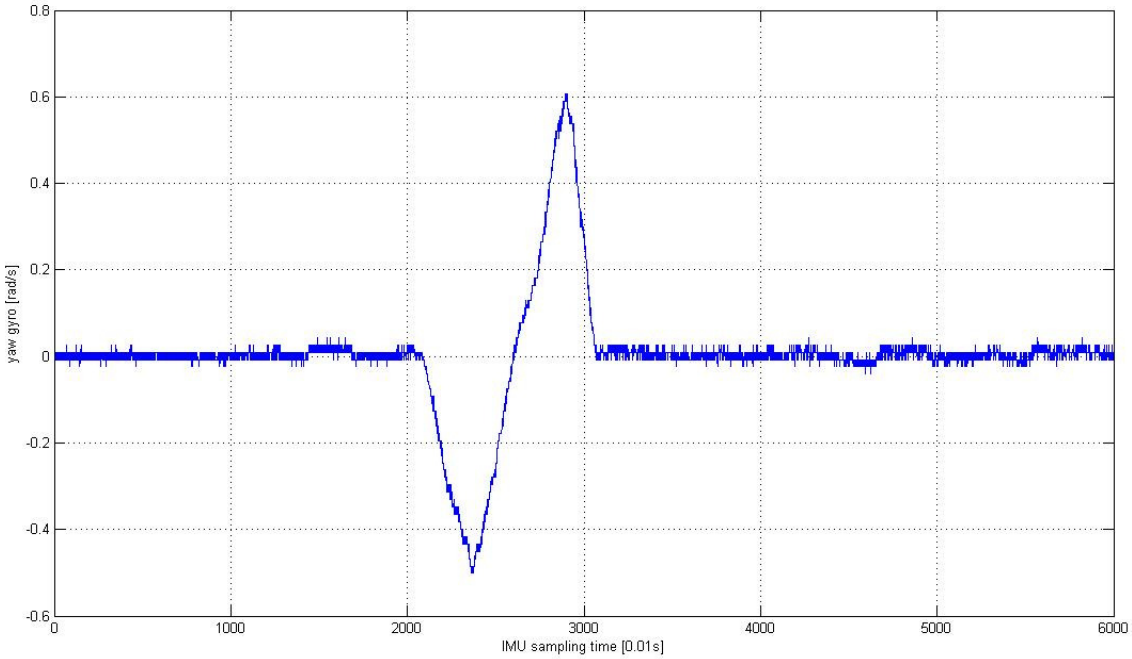

Figure 3.4: Yaw-gyro measurement of 60s

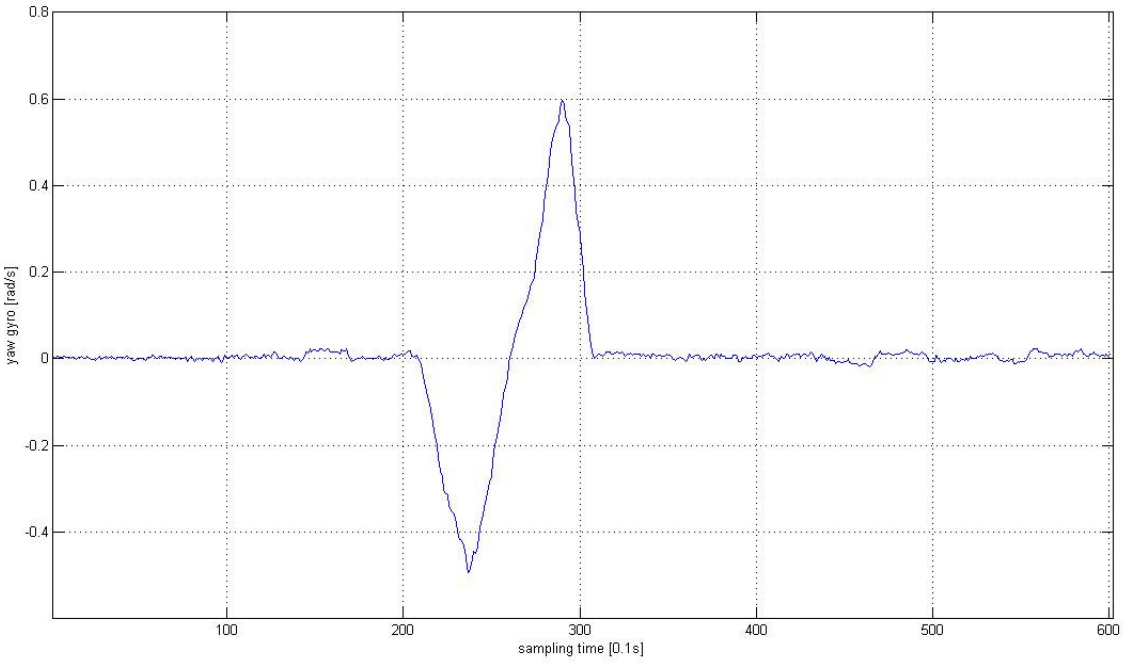

Figure 3.5: Mean yaw-gyro measurement of 60s

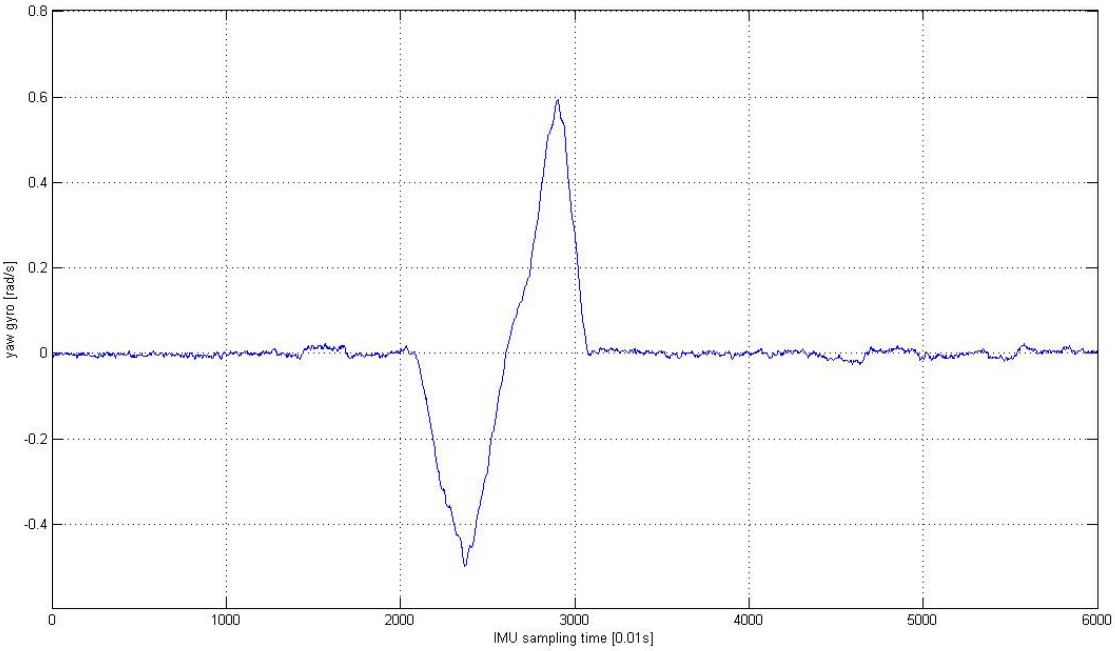

Figure 3.6: Yaw-gyro measurement of 60s (moving average)

The sensor measurement ($\tilde{f}_x, \tilde{\omega}_x$) includes the specific forces and some sensor error factors. It can be modeled by the following sensor error equations ([19] Sukkarieh, [21] Titterton). Eq. (3.2) is the linear accelerometer error compensation equation.

$$\tilde{f}_x = b + s_x f_x + m_y f_y + m_z f_z + \eta \tag{3.2}$$

And Eq. (3.3) is the gyroscope error compensation equation.

$$\tilde{\omega}_x = b + b_g \vec{f} + s_x \omega_x + m_y \omega_y + m_z \omega_z + \eta \tag{3.3}$$

Including the following coefficients:

Residual bias……..	b
g-dependent bias……	b_g
Specific forces……	f_x
Turn rates……	ω_x
Scale factor term……	s_x
Cross-coupling coefficients…….	m_y, m_z
Random noise……	η

The bias is the gap between the physical acceleration input and the measured accelerometer output and it is defined by the IEEE standards ([12] IEEE Standard) as:

“The average over a specified time of accelerometer output measured at specified operating conditions that has no correlation with input acceleration or rotation. Bias is expressed in [m/s², g].”

Figure 3.7 shows the input-output characteristic of a gyroscope. It gives an example of the scale factor hysteresis and the bias.

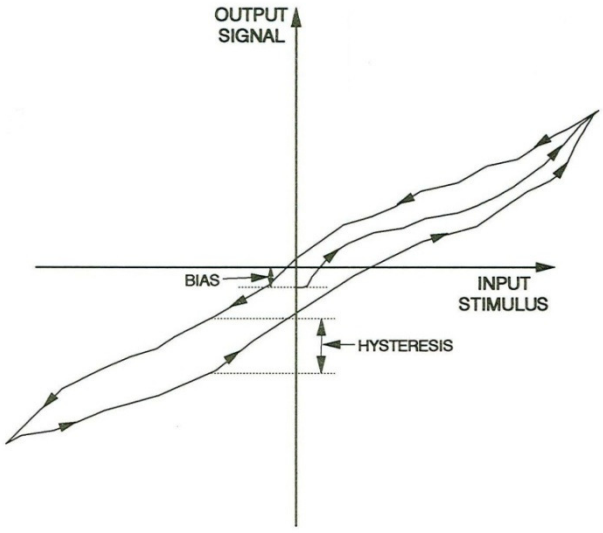

Figure 3.7: Scale factor characteristics of a gyro ([21] Titterton/Weston)

Except of the random noise, all other coefficients of the error equations can be validated through the technical datasheets of the manufacturer or by additional calibration tests. There are some more aspects, which can be included, if there is additional information available, like temperature influence and drift coefficients, which describe the variation with time ([4] Dorobantu). The cross-coupling coefficients specify the mounting and misalignment of the inertial sensors, but are not given by the manufacturer, as well as the g-dependent bias. They are also difficult to determine experimentally for us. Thus we are only considering the scale factor and bias estimation with the Kalman Filter described in Chapter 4.

We are getting the position, velocity and attitude error associated with the sensor errors by integrating Eq. (3.2) and Eq. (3.3). Regarding only the predominant error sources of low accurate inertial sensors, in fact the bias and random noise, we obtain

$$V = V + b_f \cdot t + \int \eta dt \tag{3.4}$$

$$P = P + \frac{b_f \cdot t^2}{2} + \int\int \eta dt \tag{3.5}$$

$$\Psi = \Psi + b_\omega \cdot t + \int \eta dt \tag{3.6}$$

The bias is the main source of error. It causes an error in velocity and attitude linearly with time and in position quadratically with time, see Eq. (3.4) to Eq. (3.6). But considering that the error in the attitude leads to a misalignment of the accelerometer axes and therefore the position error increases with the cube of time ([19] Sukkarieh). Hence we can see that very accurate gyros are necessary for inertial navigation.

The second important source of the error is the random noise, which can be treated as white Gaussian noise. The mean of that random noise is zero, Eq. (3.7), but at each time the error increases in a random direction, also called random walk.

$$E[x(t)] = E[\int_0^t \eta(u)\delta u] = \int_0^t E[\eta(u)]\delta u = 0 \tag{3.7}$$

Hence the random walk can cause an unbounded error without any external correction ([19] Sukkarieh).

3.2 Sensor Calibration and Alignment

The firmware of the IMU returns only ADC-values. Thus we have to find out first the offset values for zero-g of the linear accelerometers and the offset values for the three gyros. The accelerometer offset is determined by the midpoint of a positive one g and negative one g measurement with the IMU in a horizontal position for each axis. The offset values of the gyros are given by a measurement in a stationary state. The calibration of the linear accelerometers varies for each sensitive mode and the measurement values for that are listed in the appendix A.1.

Before we can use the IMU for navigation we have to determine the initial alignment. If we measure the accelerations of the car stationary we can use the gravity force to

obtain the pitch and bank angle ([19] Sukkarieh). Without any acceleration of the car we can resolve the velocity equation (2.28) to obtain Eq. (3.8):

$$\left(\vec{\mathbf{f}}\right)_B = \mathbf{M}_{OB}^{-1} \cdot \left(-\vec{\boldsymbol{\gamma}}\right)_O = \mathbf{M}_{BO} \cdot \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} \tag{3.8}$$

This yields Eq. (3.9) to Eq. (3.11).

$$f_x = g \sin \Theta \tag{3.9}$$

$$f_y = -g \cos \Theta \sin \Phi \tag{3.10}$$

$$f_z = -g \cos \Theta \cos \Phi \tag{3.11}$$

Resolving Eq. (3.9) yields the pitch angle Θ and substituting this into Eq. (3.10) or Eq. (3.11) yields the bank angle Φ.

The initial position (longitude, latitude and altitude) is provided by the GPS receiver. We also have to use the heading information of the GPS receiver. To improve the initial alignment the use of a compass sensor is necessary. Also tilt sensors to obtain the pitch and bank angle can improve the alignment, since the low-cost sensors are not very accurate. The in-car calibration of the linear accelerometer to determine the initial bias is only possible after the exact alignment of the attitude.

4 Integrated Navigation System

Current state-of-the-art car navigation systems use GPS as source for the position information. It is very convenient, because no further information about the car is necessary and thus it is easy to install. But there are many other sources we can use to develop an improved navigation system ([16] Skog/Händel 2007). Map matching is an option for example to match the current measured position with existing roads. Integration of the vehicle dynamic into the navigation system is another alternative, as well as for example integrating information of cameras, radar or laser scanners, see Figure 4.1.

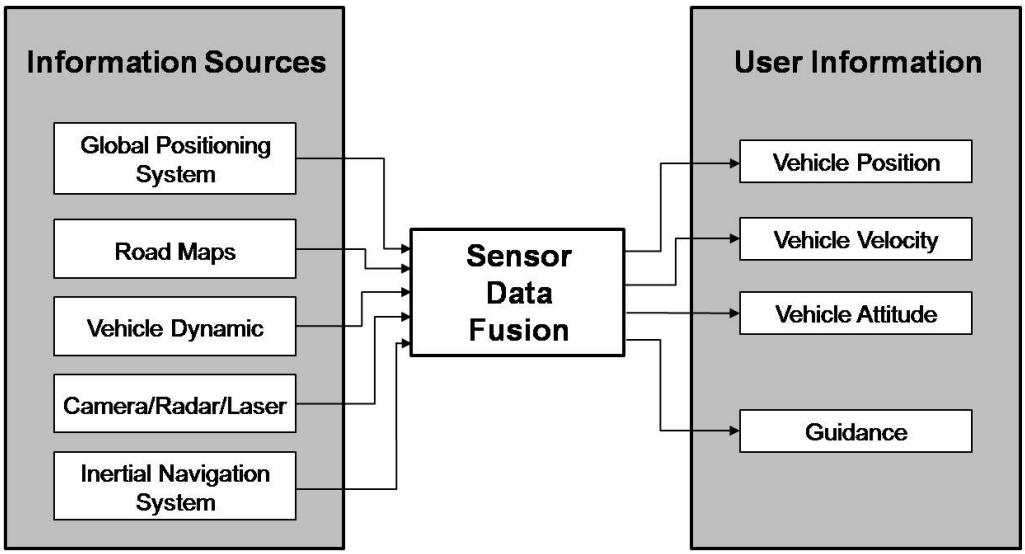

Figure 4.1: Navigation Information Sources

The option we have chosen is to measure the vehicle motion with accelerometers and gyroscopes to design an Inertial Navigation System (INS). Then we use a GPS-Receiver to aid the INS. If more than one information source is used, it is often called an Integrated Navigation System. The disadvantages of integrating more information sources are a higher complexity of the system, more effort to integrate the sensors in the vehicle and synchronization of the data, since measurements occur at different times. In our case the INS and GPS have redundant information. That means both return the vehicle's position. The goal is to enhance the accuracy of the complete system and to have position information even if there is no GPS signal available for a short period of time. The Advantage of the INS is that it is independent from any other source outside the vehicle. If the GPS-Receiver looses the satellite signal by

driving through a tunnel or because of high skyscrapers in big cities, it stops giving the user information. But the INS still works. On the other hand, the INS needs the stability of the GPS because of the low sensor accuracy after a longer period of time, since we use low-cost MEMS accelerometer and gyro sensors.

If we use more than one information source, no matter which, the challenge is to combine them for an optimal solution. There are different options how to handle the sensor data fusion, for example with fuzzy logic, neural networks, stochastic procedures or a combination of the different methods ([24] Wang/Gao) to upgrade low-level sensor data to beneficial user information.

The most popular option is to use a Kalman Filter as it is described in this section. The Kalman Filter is an effective tool to estimate the state of a linear system with noisy measurement. It is a recursive filter, which minimizes the mean of the squared error. Rudolf E. Kalman has developed it in 1960 ([25] Welch/Bishop). Figure 4.2 demonstrates the principle information flow of an integrated navigation system ([15] Skog/Händel 2005).

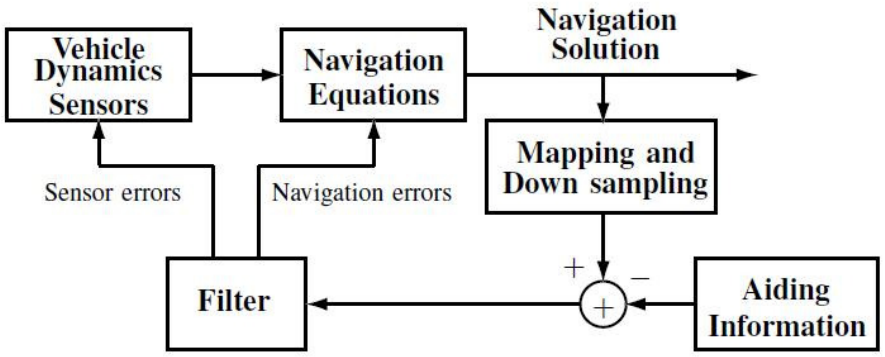

Figure 4.2: Integrated Navigation System – information flow ([16] Skog/Händel)

It shows the inertial sensor input, the aiding information (GPS input) and their information flow through the navigation equations and the Kalman Filter to achieve an optimal navigation solution.

4.1 Random Signals

First we have to describe our measurement noise mathematically, so that we can handle it later in the Kalman Filter. As already mentioned in chapter 3.1, our inertial sensor measurement is very noisy, even if the car is stationary and the IMU is not moving, see Figure 4.3.

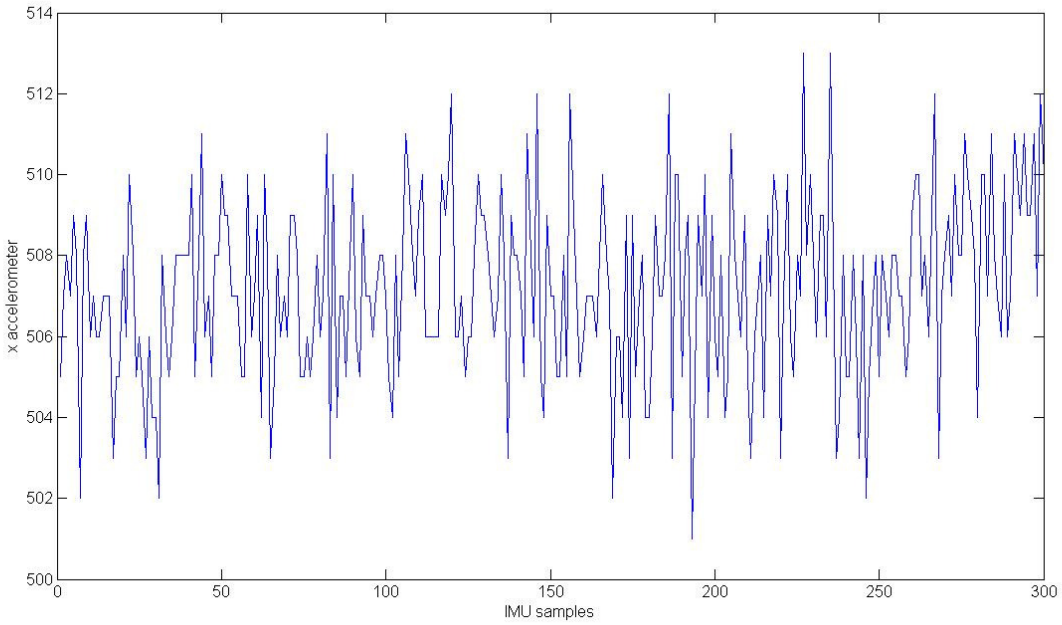

Figure 4.3: Stationary measurement of accelerometer

The measurement of the GPS receiver is also noisy, because of the mentioned error sources in chapter 1.4. Hence we have to start with the probability theory for describing our measurement noise mathematically.

The expected value of a random variable X is the mean or average over a large number of trials or measurements. It is given for the discrete case by Eq. (4.1):

$$E(X) = \sum_{i=1}^{n} p_i x_i \tag{4.1}$$

Where x_i occurs with the corresponding probability p_i. The variance of X is what we expect that the random variable vary from its mean or expectation, Eq. (4.2).

$$Var\, X = \sigma^2 = E\left[(X - E(X))^2\right] \tag{4.2}$$

The standard deviation of a random variable is the square root of its variance, see Eq. (4.3):

$$\sigma = \sqrt{Var\, X} \tag{4.3}$$

It is a *normal* or *Gaussian* random variable, if the probability density function is given by Eq. (4.4), also called the Laplace distribution:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x - m_X)^2}{2\sigma^2}\right] \tag{4.4}$$

Where m_X is the mean of the random variable X. The integration of the probability density function yields the normal distribution function, Eq. (4.5), both shown in Figure 4.4.

$$F_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left[-\frac{1}{2\sigma^2}(u - m_X)^2\right] du \tag{4.5}$$

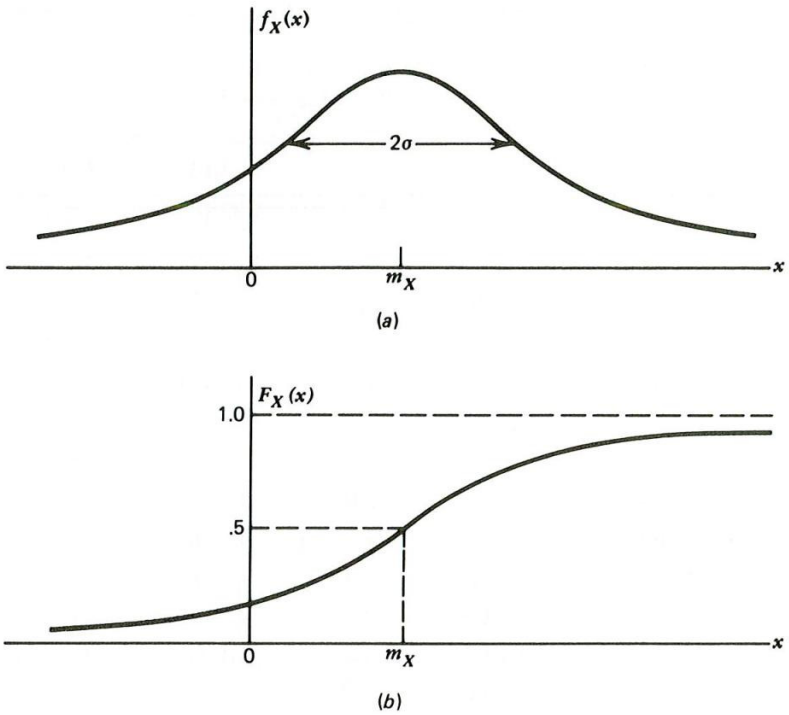

Figure 4.4: (a) Probability density function,
(b) Normal distribution function ([2] Brown/Hwang)

An often used, short notation of a random variable, with the mean m_X and the variance of σ^2 is given by (4.6):

$$X \sim (m_X, \sigma^2) \tag{4.6}$$

Regarding the correlation of different random variables we can say, if multiple random variables are statistically independent, then they are *uncorrelated*. Hence the expectation of their product is the product of their particular expectation, Eq. (4.7).

$$E(XY) = E(X)E(Y) \tag{4.7}$$

However the correlation of an n-element random variable X and an m-element random variable Y is given by Eq. (4.8), with their covariance in Eq. (4.9):

$$R_{XY} = E(XY^T) = \begin{bmatrix} E(X_1Y_1) & \cdots & E(X_1Y_m) \\ \vdots & & \vdots \\ E(X_nY_1) & \cdots & E(X_nY_m) \end{bmatrix} \tag{4.8}$$

$$C_{XY} = E\left[(X - m_X)(Y - m_Y)^T\right] \tag{4.9}$$

Considering Eq. (4.8) and Eq. (4.9) the autocovariance of an n-element random variable is defined by Eq. (4.10):

$$C_X = E\left[(X - m_X)(X - m_X)^T\right] = \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1n} \\ \vdots & & \vdots \\ \sigma_{n1} & \cdots & \sigma_n^2 \end{bmatrix} \tag{4.10}$$

Now we look at random processes. A random variable $X(t)$ is called *white noise*, if $X(t_1)$ at time t_1 is independent from $X(t_2)$ at any other time t_2. This is the case when the random process is stationary and its spectral density function is constant, which is defined as the Fourier transform of the autocorrelation function. Additional information can be found in the references ([2] Brown/Hwang, [14] Simon).

4.2 Fundamentals of Kalman Filtering

The Kalman Filter is the optimal linear filter, if the process noise and measurement noise are zero-mean, uncorrelated and white. It is a recursive filter because after each time update (*Prediction)* and measurement update (*Correction*) the process is repeated, as we can see in Figure 4.5.

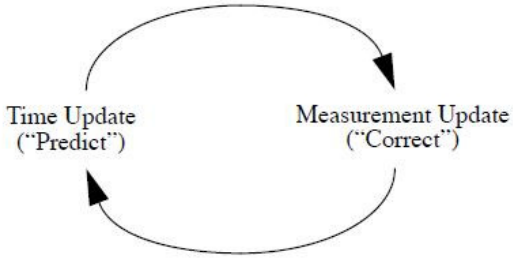

Figure 4.5: The recursive discrete Kalman Filter cycle ([25] Welch/Bishop)

The filter is based on the linear state space description, which is given by Eq. (4.11) and Eq. (4.12) in a discrete form:

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{4.11}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k \tag{4.12}$$

Where $\mathbf{w}_k$ is the process noise, in our application the noise of the INS, and $\mathbf{v}_k$ is the measurement noise of the GPS receiver, both zero-mean, white and uncorrelated, with the covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$.

$$\mathbf{w}_k \sim (\mathbf{0}, \mathbf{Q}_k) \tag{4.13}$$

$$\mathbf{v}_k \sim (\mathbf{0}, \mathbf{R}_k) \tag{4.14}$$

Now we have to distinguish between a priori estimate $\hat{\mathbf{x}}_k^-$ and a posteriori estimate $\hat{\mathbf{x}}_k^+$ ([2] Brown/Hwang, [14] Simon and [25] Welch/Bishop). It is called a priori estimate at the time k, after the time update but before the measurement is taken into account, while the posteriori estimate at the time k is after the measurement. The covariance $\mathbf{P}_k$ of the estimated error is given by Eq. (4.15):

$$\mathbf{P}_k = \mathbf{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \tag{4.15}$$

In Figure 4.6 a timeline is shown for the time update and measurement update.

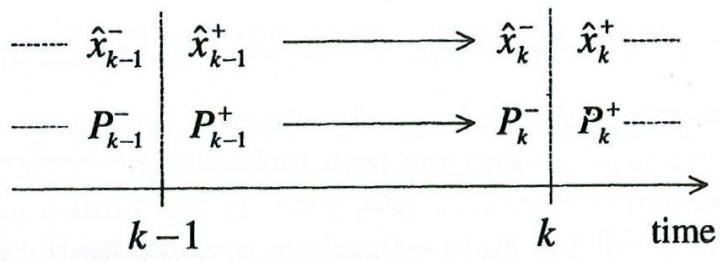

Figure 4.6: Timeline showing a priori and a posteriori state estimate and estimation error covariances ([14] Simon)

Then the time update equations Eq. (4.16) and Eq. (4.17) are:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \tag{4.16}$$

$$\mathbf{P}_k^- = \mathbf{A}_{k-1}\mathbf{P}_{k-1}^+\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \tag{4.17}$$

At the time a GPS measurement occurs, we have to calculate the so called Kalman gain $\mathbf{K}_k$, which weights the difference between the measurement and the priori state estimation $(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k^-)$, often called innovation or residual. Then we can update the estimated state and calculate the posteriori error covariance. Eq. (4.18) up to Eq. (4.20) are the so called measurement update equations:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}^T(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \mathbf{R})^{-1} \tag{4.18}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k^-) \tag{4.19}$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_k^- \tag{4.20}$$

The K-gain is dependent on the measurement noise covariance $\mathbf{R}$ and the estimated error covariance $\mathbf{P}$. If we have a closer look at Eq. (4.18) and rewrite it, we obtain:

$$\mathbf{K}_k = \frac{\mathbf{P}_k^-\mathbf{C}^T}{\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \mathbf{R}} \tag{4.21}$$

In Eq. (4.21) we can see, if the measurement noise covariance $\mathbf{R}$ becomes smaller, the K-gain weights the residual more. And if the error covariance $\mathbf{P}$ becomes smaller the K-gain weights the residual less heavily ([25] Welch/Bishop). There exist

several different algebraic forms of the Kalman Filter equations. Some of them are more robust, but at the cost of computational effort.

4.3 Linearized Kalman Filter

Figure 4.7 shows the information flow of the integrated navigation system with a linearized Kalman Filter and a bias feedback.

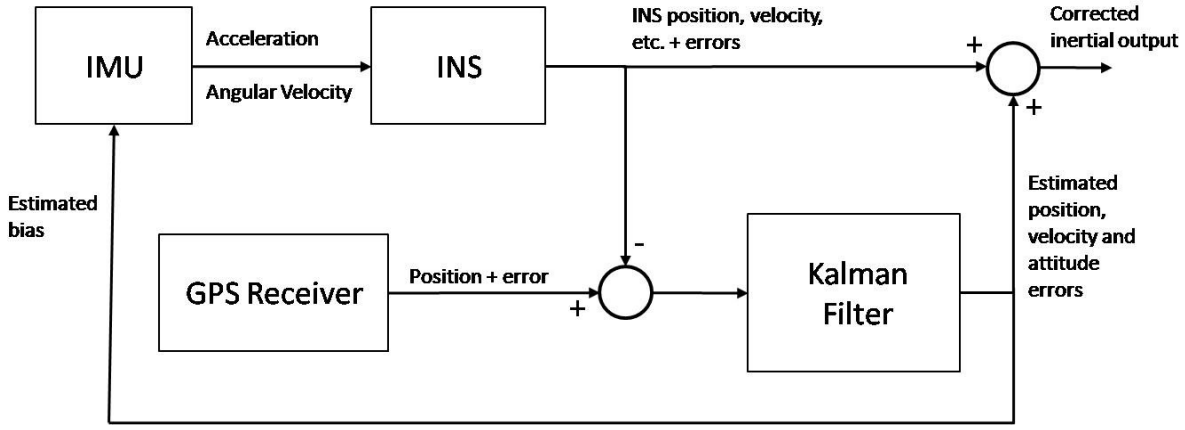

Figure 4.7: Linearized Kalman Filter information flow

The linear acceleration and angular velocity of the IMU sensors are feed into the INS to calculate the position, velocity and attitude of the vehicle. The difference of the GPS position and the inertial measured position is given to the Kalman Filter to estimate the position-, velocity-, attitude- errors and the bias of the inertial sensors. Finally the errors are used to correct the navigation output and the inertial sensors to obtain better results of the INS.

The Kalman Filter is the optimal state estimator for a linear system. Hence we have to linearize our nonlinear navigation equations. First we start with a nonlinear system, Eq. (4.22) and Eq. (4.23).

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \tag{4.22}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{4.23}$$

Where $\mathbf{w}_k$ and $\mathbf{v}_k$ are again the process and measurement noise. To obtain a linear system, which approximates our nonlinear system, we use the Taylor series expansion around a nominal operating point $\bar{\mathbf{x}}$, given by Eq. (4.24):

$$f(x) = f(\bar{x}) + \left.\frac{\partial f}{\partial x}\right|_{\bar{x}} \Delta x + \left.\frac{1}{2!}\frac{\partial^2 f}{\partial x^2}\right|_{\bar{x}} \Delta x^2 + \left.\frac{1}{3!}\frac{\partial^3 f}{\partial x^3}\right|_{\bar{x}} \Delta x^3 + \dots \tag{4.24}$$

Where the deviation of the nominal point is defined by Eq. (4.25):

$$\Delta x_k = x_k - \bar{x}_k \tag{4.25}$$

For our Kalman Filter we consider only the linear terms. Thus we have to calculate the partial derivatives, the Jacobian matrix of the continuous nonlinear system, Eq. (4.26), since our state vector is an n-element vector.

$$\mathbf{F}(\mathbf{x}) = f'(\mathbf{x}) = \begin{bmatrix} \partial f_1/\partial x_1 & \cdots & \partial f_1/\partial x_n \\ \vdots & & \vdots \\ \partial f_n/\partial x_1 & \cdots & \partial f_n/\partial x_n \end{bmatrix} \tag{4.26}$$

After discretizing the continuous system we obtain the state transition matrix $\mathbf{A}_k$ by Eq. (4.27):

$$\mathbf{A}_k = e^{\mathbf{F}\Delta t} = \mathbf{I} + \mathbf{F}\Delta t + \frac{1}{2!}(\mathbf{F}\Delta t)^2 + \dots \approx \mathbf{I} + \mathbf{F}\Delta t \tag{4.27}$$

There we also approximate it with only the linear terms of the Taylor series expansion. And Δt is the time step of updating the navigation equations, assuming that during the sampling intervals the signal of the inertial sensors is constant. Furthermore the observation matrix is given by Eq. (4.28):

$$\mathbf{C}_k = h'(\bar{\mathbf{x}}) \tag{4.28}$$

The Kalman filter equations are then obtained by the following set of equations separated in time update (4.29) and measurement update (4.30) equations:

$$\begin{aligned} \Delta\hat{\mathbf{x}}_k^- &= \mathbf{A}_{k-1}\Delta\hat{\mathbf{x}}_{k-1}^+ \\ \mathbf{P}_k^- &= \mathbf{A}_{k-1}\mathbf{P}_{k-1}^+\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \end{aligned} \tag{4.29}$$

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}_k)^{-1} \\ \Delta\hat{\mathbf{x}}_k^+ &= \Delta\hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{C}_k \bar{\mathbf{x}}_k - \mathbf{C}_k \Delta\hat{\mathbf{x}}_k^-) \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k)\mathbf{P}_k^-\end{aligned} \tag{4.30}$$

Finally we have to add the estimated deviation to the nominal trajectory, which is in our case the solution of the INS, to obtain the corrected navigation output with Eq. (4.31):

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \Delta\hat{\mathbf{x}}_k \tag{4.31}$$

Since the INS solution is diverging, because of the drift of the inertial sensors, we include the bias of the acceleration sensors into our state vector to correct the acceleration input and improve so the INS.

Example:

For a better comprehension an example is given here for a two dimensional navigation system. The state vector (4.32) consists of the position, velocity, heading and the acceleration biases:

$$\mathbf{x} = \begin{bmatrix} x & y & u & v & \psi & bx & by & br \end{bmatrix}^T \tag{4.32}$$

The inputs are the accelerometer and gyro signals of the IMU (4.33):

$$\mathbf{u} = \begin{bmatrix} fx & fy & r \end{bmatrix}^T \tag{4.33}$$

Earth rotation and transport rate are neglected in this example. The observations are the GPS measurements in Cartesian coordinates (4.34):

$$\mathbf{y} = \begin{bmatrix} x & y \end{bmatrix}^T \tag{4.34}$$

The measurement noise covariance matrix is given by the diagonal matrix of the GPS position variance (4.35):

$$\mathbf{R} = diag \begin{bmatrix} \sigma_{px}^2 & \sigma_{py}^2 \end{bmatrix} \tag{4.35}$$

And the process noise covariance includes the variance of the acceleration on the velocity update (4.36):

$$\mathbf{Q} = diag\begin{bmatrix}0 & 0 & \sigma^2_{x\_acc} & \sigma^2_{y\_acc} & \sigma^2_{r\_gyro} & 0 & 0 & 0\end{bmatrix} \tag{4.36}$$

The nonlinear navigation equations are given by (4.37):

$$f(\mathbf{x},\mathbf{u}) = \begin{bmatrix} u \\ v \\ (fx+bx)\cos\psi - (fy+by)\sin\psi \\ (fx+bx)\sin\psi + (fy+by)\cos\psi \\ r+br \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.37}$$

Hence our INS solution, Eq. (4.38), is our nominal trajectory. A simple rectangular integration is used:

$$\bar{\mathbf{x}}_k = \bar{\mathbf{x}}_{k-1} + f(\mathbf{x},\mathbf{u})\cdot\Delta t \tag{4.38}$$

Starting with the initial state $\bar{\mathbf{x}}_0$. The observation matrix is given by Eq. (4.39), using Eq. (4.28):

$$\mathbf{C}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.39}$$

Now we have to calculate the discrete state transition matrix (4.40) to update the Kalman equations in time.

$$\mathbf{A}_k = \mathbf{I} + f'(\mathbf{x},\mathbf{u})\Delta t =$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -(fx+bx)\sin\psi - (fy+by)\cos\psi & \cos\psi & -\sin\psi & 0 \\ 0 & 0 & 0 & 1 & (fx+bx)\cos\psi - (fy+by)\sin\psi & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Delta t \tag{4.40}$$

Finally we can calculate the Kalman Filter equations from (4.30) to obtain the deviation of the nominal trajectory, which we can use to correct the navigation output. With the initial values given by $\Delta\hat{\mathbf{x}}_0$ and $\mathbf{P}_0$.The estimated biases are feed back to correct the IMU input signals.

The linearized Kalman Filter works very well as long as the nominal trajectory goes not too far away from the true state, which would be the case without bias correction, because of the drift of the inertial solution.

4.4 Extended Kalman Filter

The information flow of the integrated navigation system with an extended Kalman Filter is shown in Figure 4.8.

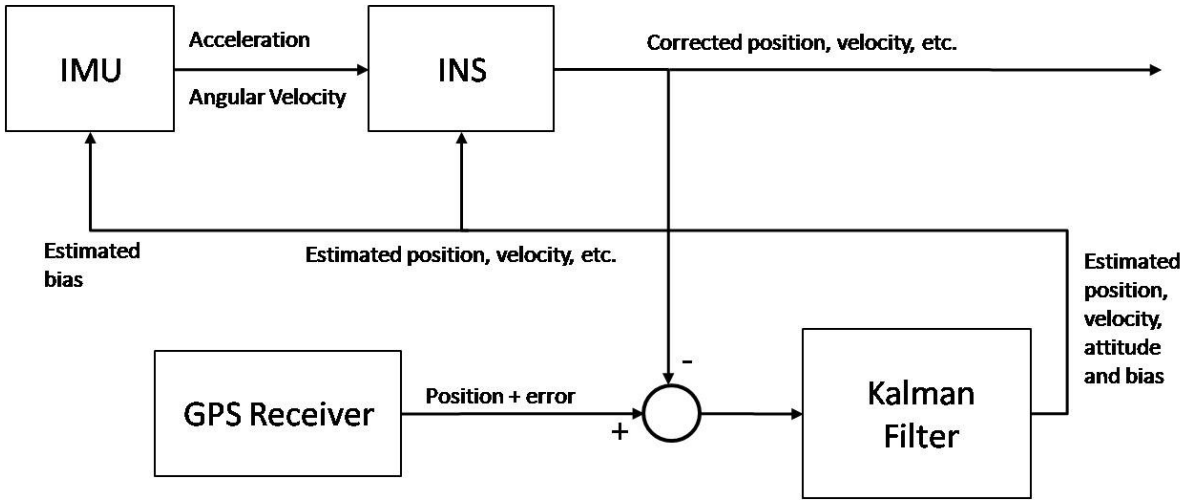

Figure 4.8: Extended Kalman Filter information flow

The main difference to the linearized Kalman Filter is that the extended Kalman Filter estimates directly the true state and not the deviation of the nominal trajectory. The estimated state is then fed back into the INS to process the nominal trajectory ahead which is then recursively estimated by the Kalman Filter. Thus the nominal trajectory is the estimated state.

The Kalman Filter equations are then modified to be executed as in (4.41):

$$
\begin{aligned}
\hat{\mathbf{x}}_k^- &= \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) \\
\mathbf{P}_k^- &= \mathbf{A}_{k-1}\mathbf{P}_{k-1}^+\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \\
&\text{---------------------------------} \\
\mathbf{K}_k &= \mathbf{P}_k^-\mathbf{C}_k^T(\mathbf{C}_k\mathbf{P}_k^-\mathbf{C}_k^T + \mathbf{R}_k)^{-1} \\
\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-) \\
\mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\mathbf{P}_k^-
\end{aligned}
\tag{4.41}
$$

5 Results

This chapter shows the results of the INS, the linearized Kalman Filter and the extended Kalman Filter. Several test drives have been done with the REV Eco of the Renewable Energy project. For comparing the different performances we use the streets around the UWA campus shown in Figure 5.1 with the GPS Visualizer[2]:

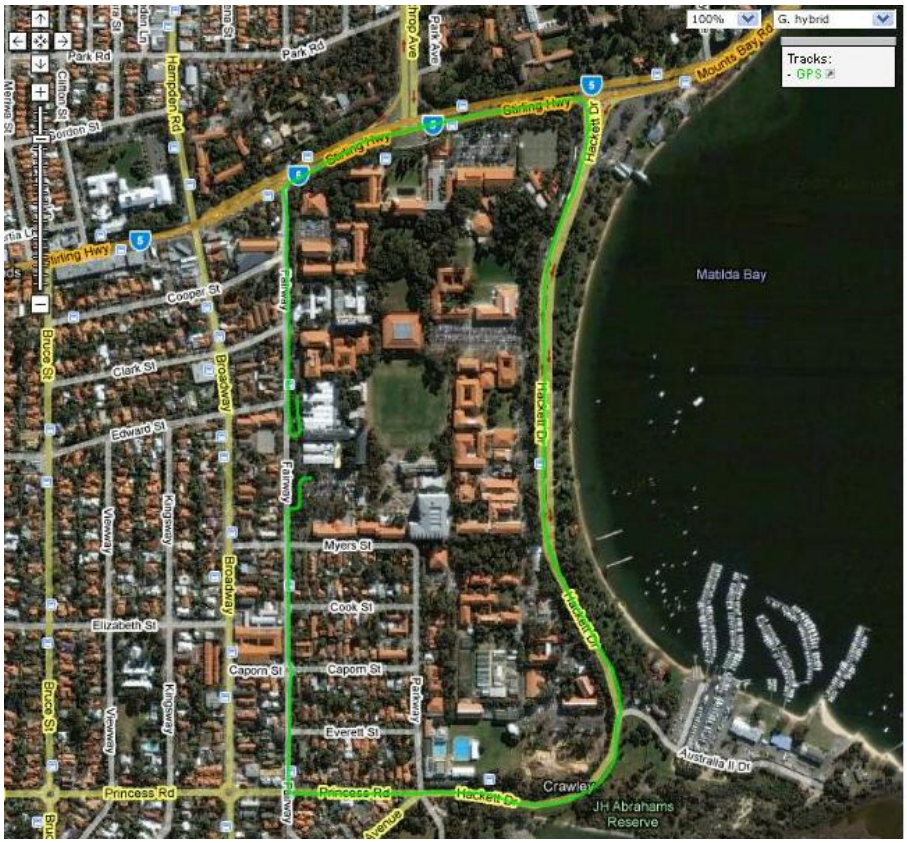

Figure 5.1: GPS track around the UWA campus

5.1 GPS-aided Inertial Navigation System

Only the two dimensional example of chapter 4.3 is taken into account. Figure 5.2 shows the stand-alone result of the INS in comparison to the GPS measurement. The GPS measurement is very accurate and close to the real track, while the INS solution is drifting far away, because of low accurate sensors and their bias.

[2] www.gpsvisualizer.com

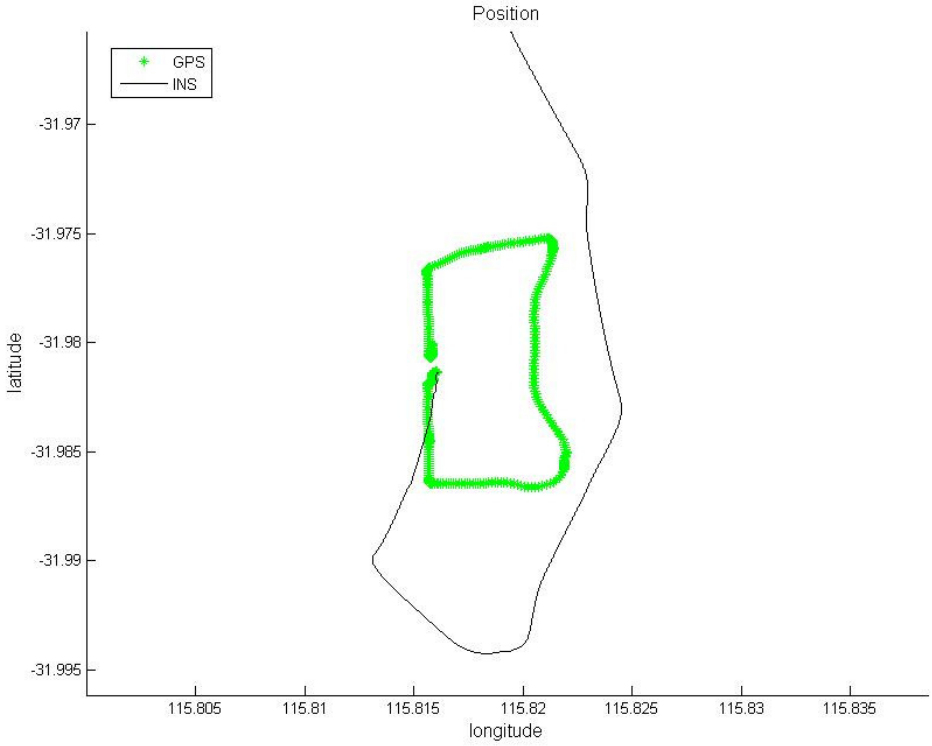

Figure 5.2: INS solution in comparison to the GPS measurement

Comparing the results of the linearized Kalman Filter (LKF) and the extended Kalman Filter (EKF) we used the same process noise covariance matrix Q and measurement noise covariance matrix R. In Figure 5.3 to Figure 5.6 the results are shown with a measurement update every second and every five seconds only.

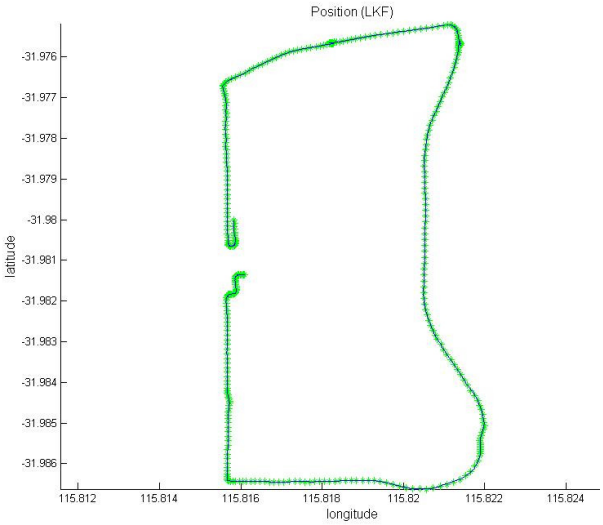

Figure 5.3: LKF solution (update 1sec)

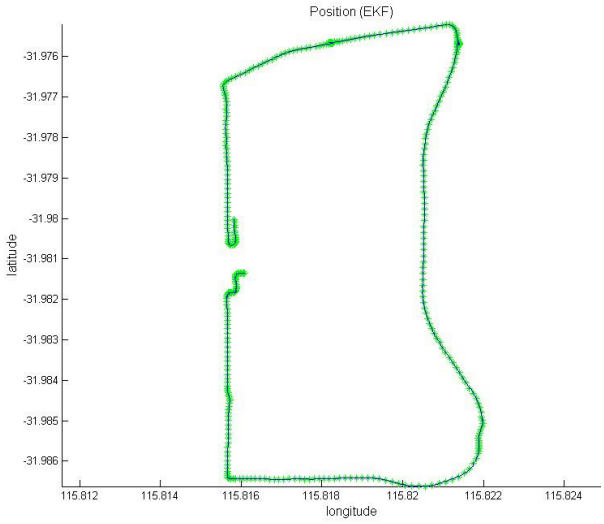

Figure 5.4: EKF solution (update 1sec)

If there is a GPS signal available every second to update the INS, both filter versions work very well. Even if we use the measurement only every five seconds, we can see

no recognizable difference to decide which filter version is better. In total the results are as expected worse than with an update every second. Only at some points it looks like the EKF version works better than the LKF version.

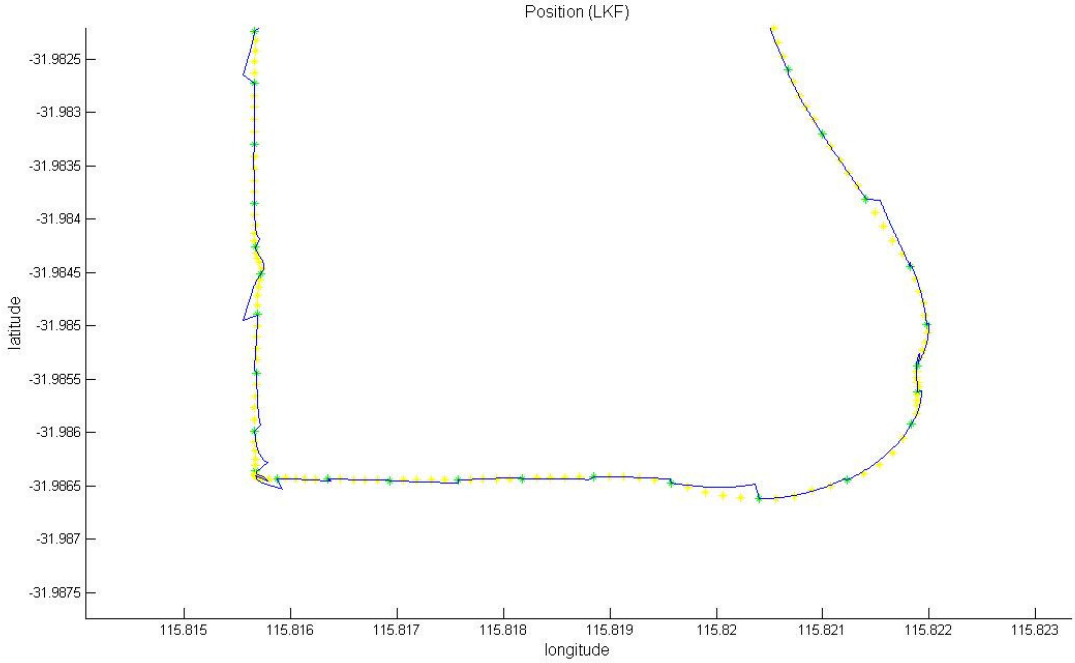

Figure 5.5: LKF simulated (update every 5sec)

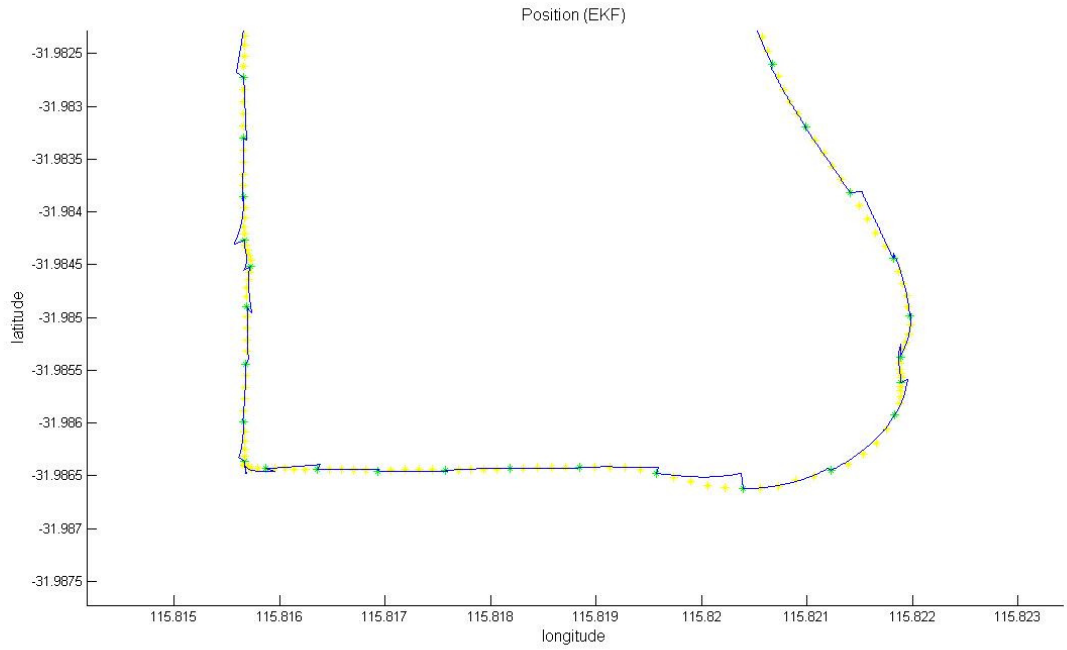

Figure 5.6: EKF simulated (update every 5sec)

5.2 Inertial Navigation during Loss of GPS Signals

Now a loss of the measurement signal over ten seconds and over twenty seconds is simulated. The next four graphs (Figure 5.7 to Figure 5.10) show that a loss of the GPS signal for a period of ten seconds is compensated very well, a little bit better with the LKF version.

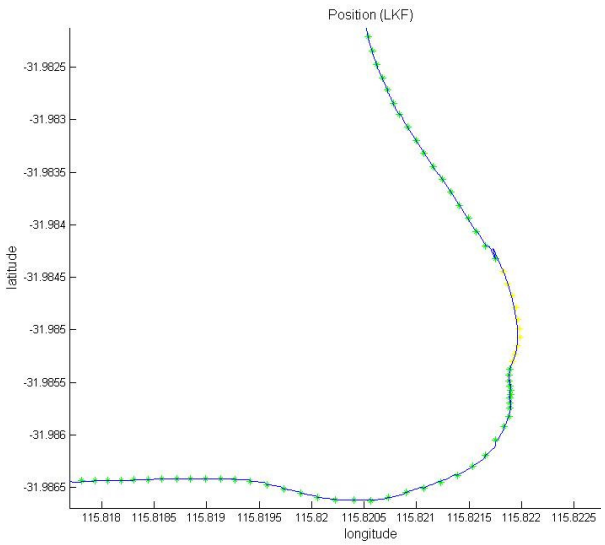

Figure 5.7: LKF signal loss of 10sec

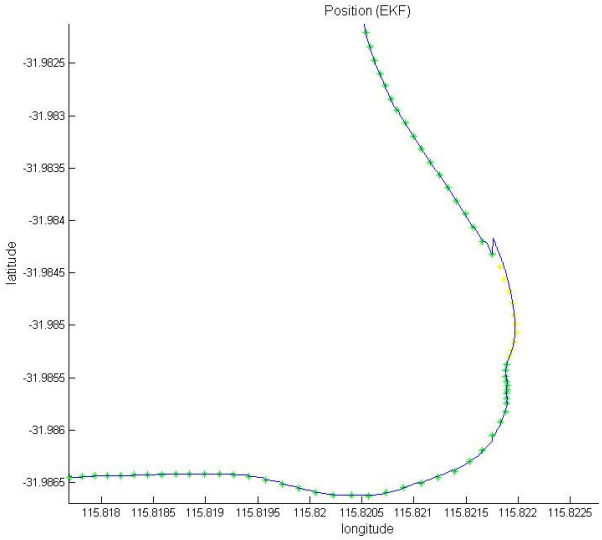

Figure 5.8: EKF signal loss of 10sec

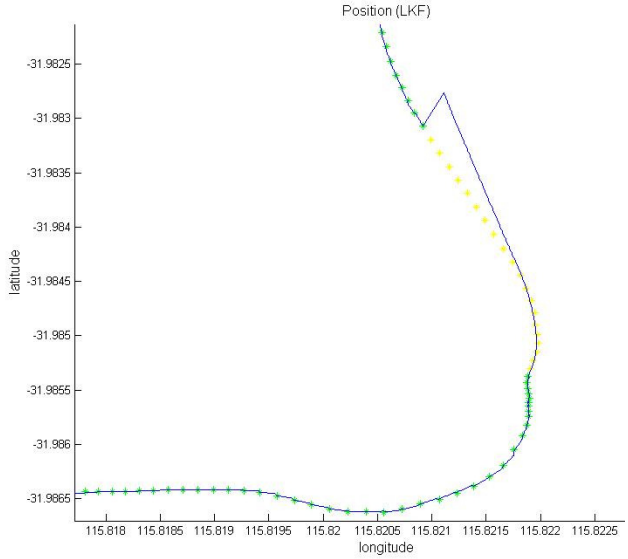

Figure 5.9: LKF signal loss of 20sec

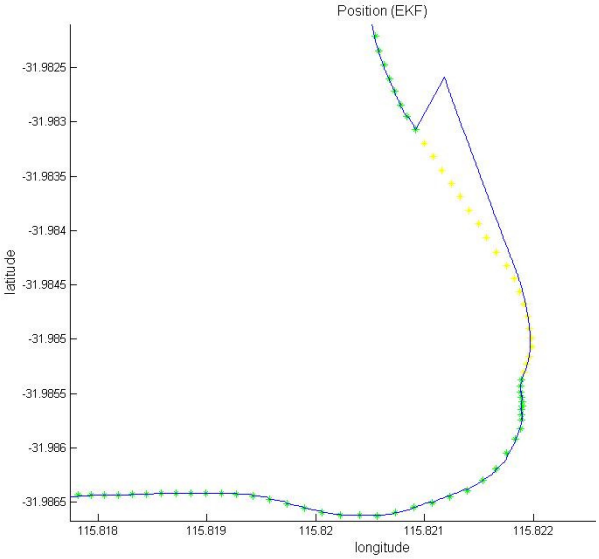

Figure 5.10: EKF signal loss of 20sec

While a loss of the GPS signal over twenty seconds yields already bad results. It also depends on how good the filter worked before the updating signal is lost. If the signal is lost for example while the car is in a stop position in front of a traffic light the INS position information is drifting away, see Figure 5.11. It is caused of misalignments of the IMU in the car or because of the simplification of regarding only a two-

dimensional model, which does not take into account that there can be a remaining acceleration measurement in x- or y- direction due to the gravity, when the vehicle is not in a complete horizontal attitude. And without measurement update it results into a position error. Hence in this case some additional information has to be included for recognizing when the car is not moving.

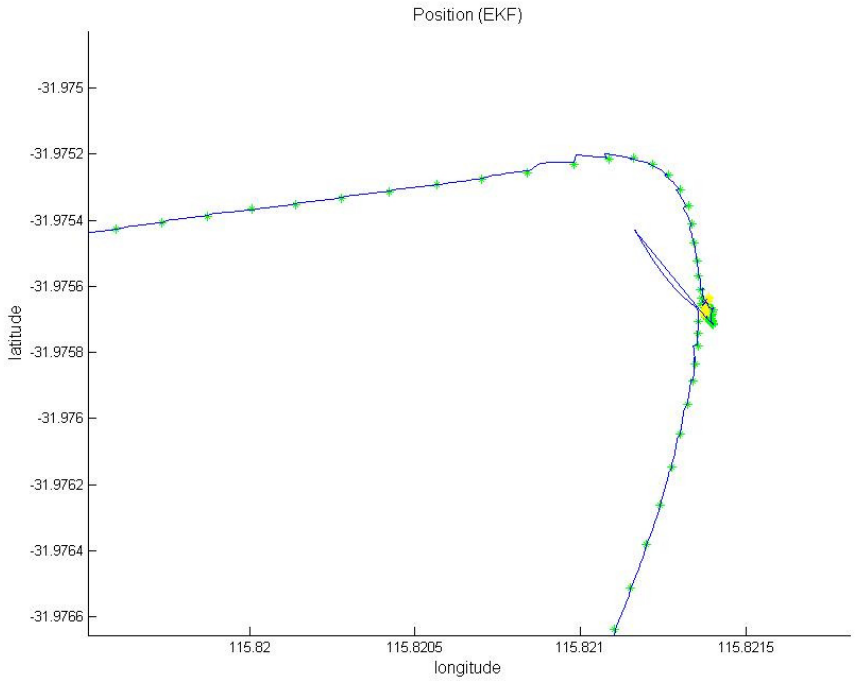

Figure 5.11: GPS signal loss of 20sec while car is not moving

5.3 Estimated States and Bias

Figure 5.12 demonstrates that the states of the LKF solution and the EKF solution do not differ very much from each other.

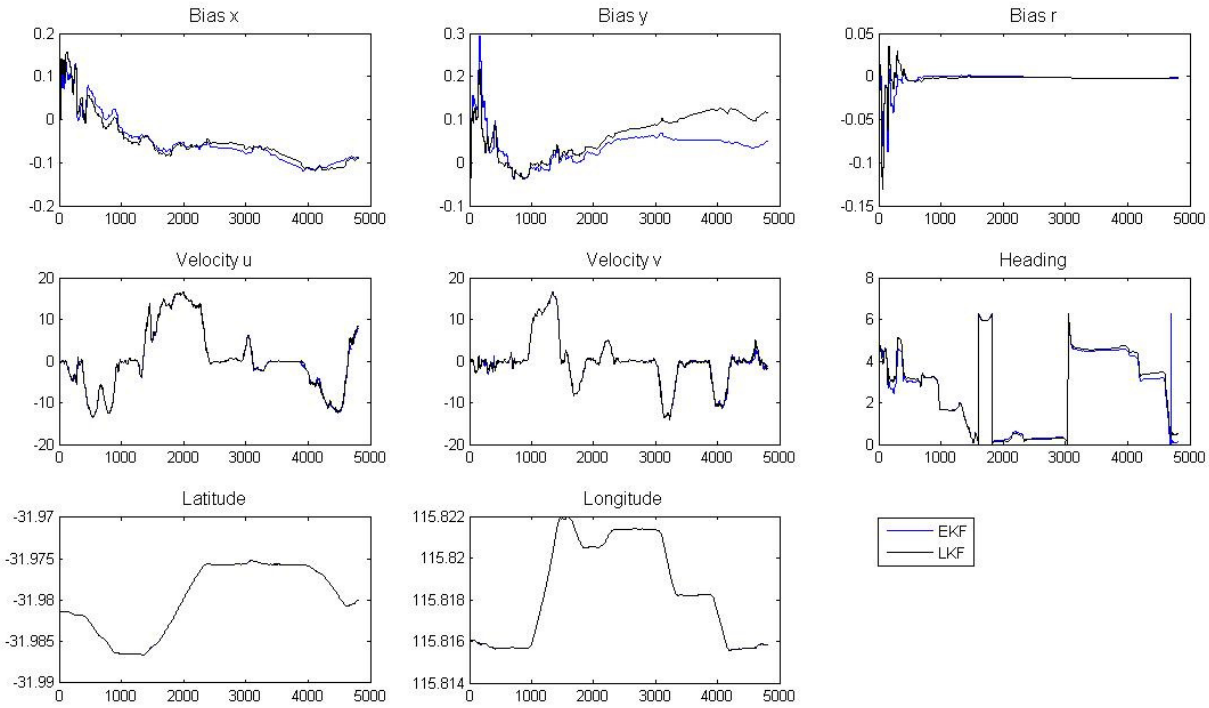

Figure 5.12: Comparison of the LKF and EKF solution

The position and velocity curves are close together. Only the heading and the linear acceleration biases differ a little bit. Especially the acceleration bias of the LKF in y-direction is getting larger until the end of the test measurement. A possible cause of this could be that the nominal trajectory, thus the INS solution, differs too much from the real trajectory. Also the estimated error state of the LKF could then underlie a boundless growth. This is also reflected in the heading solution of the LKF in comparison to the EKF. After the sampling >3000 it differs more and more from the EKF and GPS heading.

In Figure 5.13 the estimated heading of the two filter versions is compared to the heading information of the GPS receiver (the heading angle is in [rad] and between $0 < \psi < 2\pi$). We can see that the estimated heading is close to the measured heading of the GPS. But while the vehicle is not moving the heading information is varying randomly (between samples: 2300-3000 and samples 3300-3800). Hence we can only use the GPS heading information for the measurement update if we know when the vehicle is in a stop position.

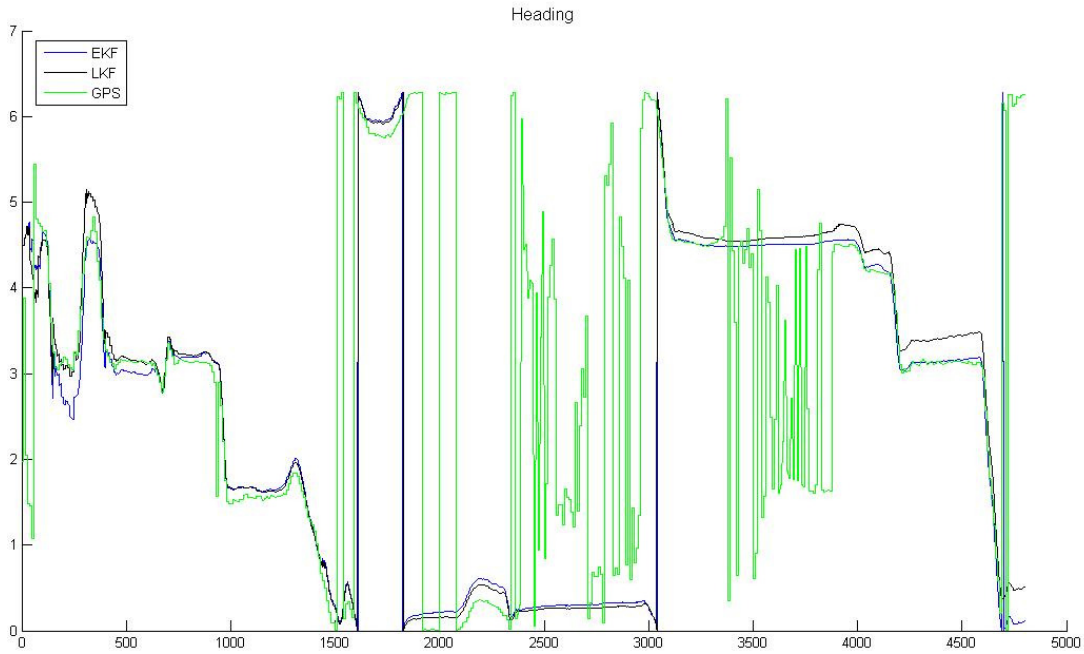

Figure 5.13: Comparison of EKF, LKF and GPS heading

Figure 5.14 shows that after a certain start-up time the error covariance of the matrix P of both filters converge to a fixed level. If there is a loss of the updating measurement signal the uncertainty grows and the trace of the error covariance rises as shown in Figure 5.15 (the peak at sample 1600). It returns to a constant level if measurement updates are again taken into account.

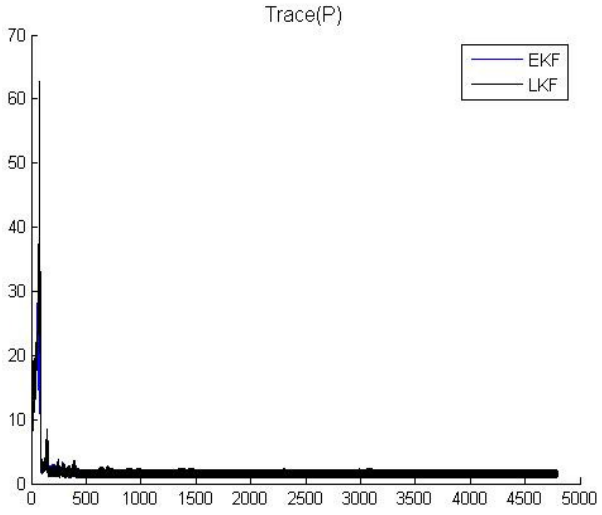

Figure 5.14: Trace Covariance P

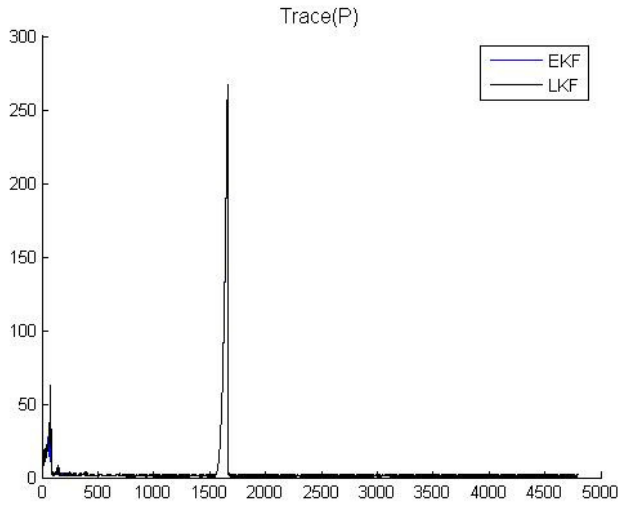

Figure 5.15: Trace Covariance P (No Signal 10sec)

5.4 In-Car Visualization

For using the integrated navigation system in one of the renewable energy vehicles we implemented a graphical user interface. Openstreetmap[3] provides open source maps. After converting the maps into binary format the testprogram (OSM_Test_binary.c) is able to show the current position of the vehicle in form of a filled rectangular at the map on the EyeBot M6 screen, see Figure 5.16.

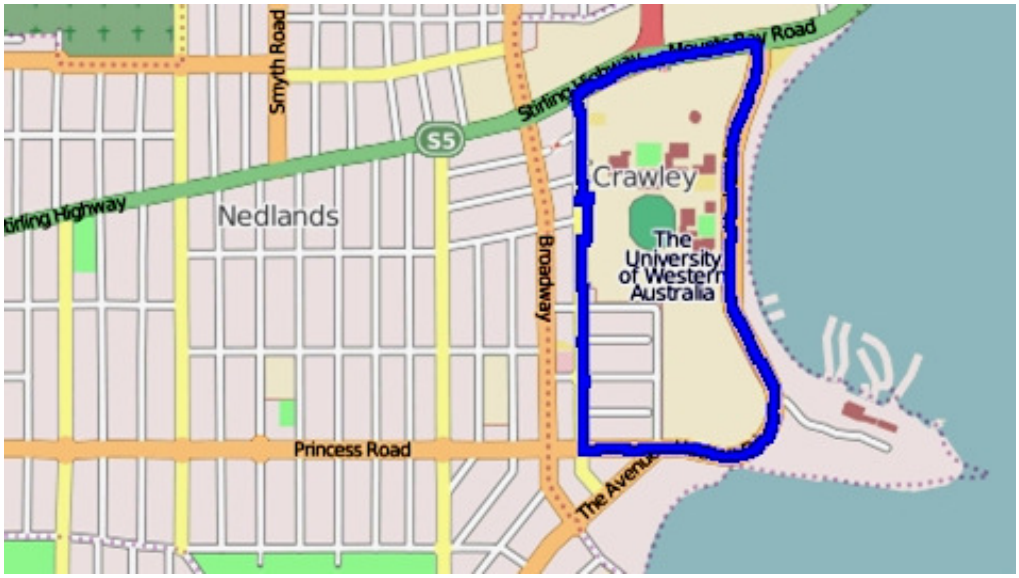

Figure 5.16: Screenshot of the EyeBot M6

Since the EyeBot M6 screen is not very large for drawing maps (480x270 pixel), we implemented that the program is automatically shifting from one map section to the next. Some of the RoBIOS library functions for drawing on the screen are very slow. Hence the screen is refreshed only after each map shifting. For covering the desired area one has to set up an appropriate database of maps.

[3] www.openstreetmap.org

6 Conclusion

For the development of an integrated navigation system a linearized and extended version of the Kalman Filter have been derived together with the strap down navigation equations. We installed the necessary hardware, which means the inertial measurement unit for the INS, the GPS receiver for additional position information and the EyeBot M6 to read out the sensor signals and to calculate the equations.

Additional sensors could be included to improve the results. A barometer would be very helpful as additional altitude information for the three-dimensional navigation, since the altitude information of the GPS receiver is not very accurate and the z-accelerometer can get quite disturbed by pot-holes for example. For a better initialization a compass sensor is necessary for the heading information and tilt sensors could be used for initializing the pitch and bank angle instead of using the gravity ([19] Sukkarieh).

Besides of more additional sensors, other information sources could be integrated for a better navigation solution like map information or a vehicle dynamic model ([16] Skog/Händel). The dynamic constraints of the vehicle can be used to specify the inertial equations ([14] Simon, [19] Sukkarieh). A fuzzy controller could be implemented to identify the vehicle dynamics ([24] Wang/Gao]), instead of using wheel sensors to detect if the car is moving or not.

Finally, the results show that our low-cost inertial sensors can be used for the navigation challenge to bridge a gap for short periods of losing aiding information like GPS. The Kalman filter plays an important role in the fusion of the sensor data. It is a perfect tool to handle the noisy measurement of the low-cost sensors in our application.

Bibliography

[1] T. Bräunl: *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*; second edition, Springer, Berlin, 2006.

[2] R. G. Brown and P. Y. C. Hwang: *Introduction to Random Signals and Applied Kalman Filtering*; second edition, John Wiley & Sons, Inc., 1992.

[3] Department of Defense World Geodetic System 1984: *Its Definition and Relationships with Local Geodetic System*; NIMA TR8350.2, 2000.

[4] R. Dorobantu: *Simulation des Verhaltens einer low-cost Strapdown IMU unter Laborbedingungen*; Technische Universität München.

[5] Eidgenössisches Departement für Verteidigung: *Formeln und Konstanten für die Berechnung der Transformation zwischen Koordinatensystemen*; Schweiz 2008.

[6] EyeBotM6 Documentation; CD:\References\Hardware\EyeBotM6Docs.pdf

[7] W. E. Featherstone and M. C. Dentith: *A Geodetic Approach to Gravity Data Reduction for Geophysics*; Computers and Geosciences Vol. 23, No. 10, 1997.

[8] Freescale Semiconductor, Datasheet: *MMA7260Q: XYZ Axis Accelerometer*; http://www.freescale.com

[9] M. S. Grewal and A. P. Andrews: *Kalman Filtering: Theory and Practice Using MATLAB*; Second Edition, John Wiley & Sons, Inc., 2001.

[10] F. Holzapfel: *Flugsystemdynamik I/II*; lecture manuscript, Technische Universität München

[11] Q. Honghui and J. B. Moore: *A direct Kalman Filtering Approach for GPS/INS Integration*; IEEE Transactions on Aerospace and Electronic Systems, 2002

[12] IEEE Aerospace and Electronic Systems Society: *IEEE Standard for Inertial Sensor* Terminology; IEEE Std 528-2001.

[13] NMEA 0183 Standard; CD:\References\Hardware\NMEA0183.pdf

[14] D. Simon: *Optimal State Estimation*; Wiley, 2006.

[15] I. Skog and P. Händel: *A Low-Cost GPS Aided Inertial Navigation System for Vehicle Applications*; Royal Institute of Technology, Stockholm, 2005.

[16] I. Skog and P. Händel: *State-of-the art and future in - car navigation systems - a survey*; IEEE Transactions on Intelligent Transportation Systems, 2007.

[17] SparkFun Electronics, Datasheet: *Atomic IMU – 6 Degrees of Freedom – XBee Ready*; http://www.sparkfun.com

[18] STMicroelectronics, Datasheet: *LISY300AL*; http://www.st.com

[19] S. Sukkarieh: *Low Cost, High Integrity, Aided Inertial Navigation Systems for Autonomous Land Vehicles*; Ph.D. Thesis, University of Sydney, March 2000.

[20] C.-W. Tan and S. Park: *Design of Acceleromter-Based Inertial Navigation Systems*; IEEE Transactions on Instrumentation and Measurement, Vol. 54, No. 6, December 2005.

[21] D. H. Titterton and J. L. Weston: *Strapdown Inertial Navigation Technology*; IEE Radar, Sonar Navigation and Avionics Series 5; London, 1997.

[22] O. Wagner: *Grundlagen der modernen Flugführung*; lecture manuscript, Technische Universität München.

[23] K. J. Walchko and P. A. C. Mason: *Inertial Navigation*; Florida Conference on Recent Advances in Robotics, 2002.

[24] J.-H. Wang and Y. Gao: *Multi-Sensor Data Fusion for Land Vehicle Attitude Estimation Using a Fuzzy Expert System*; Data Science Journal, Vol. 4, 2005.

[25] G. Welch and G. Bishop: *An Introduction to the Kalman Filter*; University of North Carolina, 2006.

Appendix

A.1 IMU Settings

The IMU is mounted in a Box and it is installed with the EyeBot M6 behind the center of the dashboard. For powering the IMU, a regulated 5 V motor output is used. After booting the EyeBot M6 we have to change the menu settings of the IMU by software. Changes, provided by the IMU firmware, which can be made are: choosing channels, output mode (ASCII or binary), auto run mode, accelerometer sensitivity mode (1.5g, 2g, 4g, 6g) and output frequency, see Figure A.1.1.

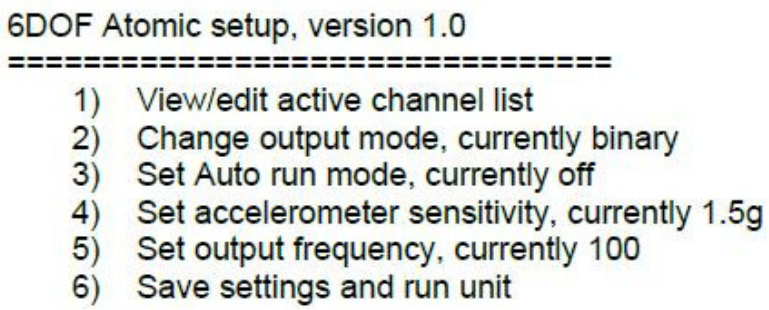

Figure A.1.1: IMU Setup Menu

The communication between the IMU and the EyeBot M6 runs with the following settings for the serial port:

Baud Rate	115200
Data Bit	8
Parity	None
Stop Bit	1
Flow Control	None

Table A.1.1: IMU serial connection settings

Figure A.1.2 gives an example of the IMU sensor reading sent over the serial port in ASCII-code. It starts with an identifier “A”, followed by a counter and the six sensor channels. Each sampling ends with the identifier “Z”.

```
A     19    481   467   777   520   503   493   Z
A     20    484   472   779   522   503   492   Z
A     21    483   468   778   520   503   492   Z
A     22    481   469   776   521   503   492   Z
A     23    481   469   774   520   503   491   Z
A     24    481   469   780   520   503   492   Z
A     25    482   469   781   520   503   492   Z
A     26    485   469   781   520   503   492   Z
```

Figure A.1.2: IMU reading example in ASCII-code

A.2 GPS-Receiver Settings

After installing the USB to serial interface driver we can start a serial connection with the following settings, as shown in Table:

Baud Rate	4800
Data Bit	8
Parity	None
Stop Bit	1
Flow Control	None

Table A.2.1: GPS receiver serial connection settings

The GPS receiver returns ASCII-Code with the “NMEA 0183 Standard”, which defines the electrical interface and data protocol for communications. The general sentence format is explained in ([13] NMEA Standard) and an example of the sentences is given by:

```
$GPGGA,080002.000,3158.7662,S,11548.9836,E,2,08,1.1,22.8,M,-29.4,M,0.8,0000*7B
$GPGSA,A,3,21,16,29,18,06,24,22,03,,,,,2.1,1.1,1.8*3D
$GPRMC,080002.000,A,3158.7662,S,11548.9836,E,0.08,209.47,040509,,*1B
$GPGGA,080003.000,3158.7662,S,11548.9836,E,2,08,1.1,22.8,M,-29.4,M,1.8,0000*7B
$GPGSA,A,3,21,16,29,18,06,24,22,03,,,,,2.1,1.1,1.8*3D
$GPRMC,080003.000,A,3158.7662,S,11548.9836,E,0.05,215.42,040509,,*1F
```

Each string starts with the identifier “$” followed by the talker identifier, in our case “GP” for GPS and the sentence identifier, here “GGA”, “GSA” and “RMC”. For a detailed list of which information is included in which sentence, see the appendix A.3.

A.3 NMEA-0183 Standard

The National Marine Electronics Association (NMEA) protocol was first released in 1983. It is a standard for the electrical interface and data protocol for communications between marine instrumentation.

The general format is that all data is transmitted in ASCII – sentences. Each sentence starts with a “$” symbol followed by the talker identifier, in our case “GP”. The next three characters are the sentence identifier and after that up to 80 characters with the data delimited by commas. For more information see reference [13]. The used sentence identifiers are:

GGA Global Positioning System Fix Data. Time, Position and fix related data for a GPS receiver

```
          1            2     3 4        5 6 7   8   9  10 11 12 13 14    15
          |            |     | |        | | |   |   |   |  |  |  |  |     |
$--GGA,hhmmss.ss,llll.ll,a,yyyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx*hh
```

1) Time (UTC)
2) Latitude
3) N or S (North or South)
4) Longitude
5) E or W (East or West)
6) GPS Quality Indicator,
   0 - fix not available,
   1 - GPS fix,
   2 - Differential GPS fix
7) Number of satellites in view, 00 - 12
8) Horizontal Dilution of precision
9) Antenna Altitude above/below mean-sea-level (geoid)
10) Units of antenna altitude, meters
11) Geoidal separation, the difference between the WGS-84 earth ellipsoid and mean-sea-level (geoid), "-" means mean-sea-level below ellipsoid
12) Units of geoidal separation, meters
13) Age of differential GPS data, time in seconds since last SC104 type 1 or 9 update, null field when DGPS is not used
14) Differential reference station ID, 0000-1023
15) Checksum

RMC Recommended Minimum Navigation Information

```
          1            2 3   4 5         6 7   8   9    10 11 12
          |            | |   | |         | |   |   |     |   |  |
$--RMC,hhmmss.ss,A,llll.ll,a,yyyyy.yy,a,x.x,x.x,xxxx,x.x,a*hh
```

1) Time (UTC)
2) Status, V = Navigation receiver warning
3) Latitude
4) N or S
5) Longitude
6) E or W
7) Speed over ground, knots
8) Track made good, degrees true
9) Date, ddmmyy
10) Magnetic Variation, degrees
11) E or W
12) Checksum

A.4 Inertial Sensor Calibration

To calibrate the accelerometers one has to turn the IMU perpendicular to the earth`s surface in both directions which each axis. The maximum/minimum value returns the gravity force. And the midpoint is the zero-g value. The Atomic 6DOF IMU has four different sensitivity modes for 1.5g, 2g, 4g and 6g. The IMU has a different calibration for each mode.

Measurements of the 3-Axes Accelerometer (MMA7260Q):

Accelerometer, 1.5g-mode		
axes	min	max
x-axis	244	748
y-axis	231	739
z-axis	277	777

Table A.4.1: Calibration, Accelerometer, 1.5g-mode

Accelerometer, 2g-mode		
axes	min	max
x-axis	307	686
y-axis	298	680
z-axis	333	709

Table A.4.2: Calibration, Accelerometer, 2g-mode

Accelerometer, 4g-mode		
axes	min	max
x-axis	403	593
y-axis	399	591
z-axis	418	607

Table A.4.3: Calibration, Accelerometer, 4g-mode

Accelerometer, 6g-mode		
axes	min	max
x-axis	435	562
y-axis	433	560
z-axis	446	572

Table A.4.4: Calibration, Accelerometer, 6g-mode

With the gyroscopes (LISY300AL) one can measure the offset under zero movement. The angular rate per ADC-count is 0.977°/tick, see the datasheet. There was no possibility to test the IMU under a constant angular rate to proof the resolution. The IMU needs a minimum voltage of 3.4 V, otherwise the raw values of the gyros are incorrect.

Gyros		
axes	min	max
x-axis (roll)	520	522
y-axis (nick)	502	503
z-axis (yaw)	491	493

Table A.4.5: Calibration, Gyros

A.5 Implementation

To save processing time we over-sample the IMU signal as already described in chapter 3.1. If we only reduce the IMU sampling frequency we would lose information, but so we can keep the sensor reading at a high frequency and calculate our navigation equations less often. On the other hand is this increasing the time step of the discrete Kalman filter state space model. But tests in Matlab and on the EyeBot M6 have shown that this is still acceptable.

Since we have to calculate many trigonometric functions we can use lookup tables for sine and cosine functions and get the results after a linear interpolation, which is also saving processing time but at the cost of precision. At the same time we should also mention the computational round off error due to the limited representation of values in the computer arithmetic ([9] Grewal/Andrews).

To use the “math.h” library one has to include the “-lm” flag for the gccarm compiler. For some of the matrix manipulation a library called “matmath.h” is used from the www.media.mit.edu website which is referenced at ([14] Simon). Especially for the matrix inversion the Gauss-Jordan algorithm implementation is very useful.