Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

Philipp Mohr

Development of a Kalman Filter for Multiple-Sensor Embedded Systems in Urban Public Transportation Applications

Fakultät Technik und Informatik
Department Informations- und Elektrotechnik

Faculty of Engineering and Computer Science
Department of Information and Electrical Engineering

Philipp Mohr

Development of a Kalman Filter for Multiple-Sensor Embedded Systems in Urban Public Transportation Applications

Bachelorthesis based on the study regulations
for the Bachelor of Engineering degree programme
Information Engineering
at the Department of Information and Electrical Engineering
of the Faculty of Engineering and Computer Science
of the Hamburg University of Aplied Sciences

Supervising examiner: Prof. Dr. rer. nat. Rasmus Rettig
Second examiner: Prof. Dr. rer. nat. Annabella Rauscher-Scheibe

Day of delivery 24. August 2016

Philipp Mohr

Title of the Bachelorthesis

Development of a Kalman filter for multiple-sensor embedded systems in urban public transportation applications

Keywords

GPS/INS integration, GPS, INS, odometry, vehicle constraint,pressure, barometer,magnetometer, extended Kalman filter, loosely coupled integration, innovation filtering, low-cost sensors, bus

Abstract

This thesis describes a robust and more precise navigation solution with fusion of low-cost inertial, pressure and magnetic-field sensors and a consumer-grade GPS-receiver using the idea of Kalman filtering. The derived algorithm implements the physical properties of each sensor and takes advantage of vehicle movement constraints with odometry assistance. An adaptive solution detects faulty measurement data especially in urban areas with poor GPS-signal quality.

Philipp Mohr

Titel der Arbeit

Entwicklung eines Kalman Filters für eingebettete Multisensorsysteme im öffentlichen Nahverkehr

Stichworte

GPS/INS Integration, GPS, INS, Odometrie, Fahrzeugmodell, Luftdruck, Barometer, Magnetometer, Extended Kalman Filter, Loosely Coupled Integration, Innovation Filter, low-cost Sensoren, Bus

Kurzzusammenfassung

Die Arbeit beschreibt die Entwicklung einer robusten und genauen Navigationslösung mit Hilfe der Fusion von kostengünstigen Trägheits-, Luftdruck- und Magnetfeldsensoren und einem GPS Empfänger unter Verwendung eines Kalman Filters. Der hergeleitete Algorithmus nutzt die physikalischen Eigenschaften der Sensoren und macht Gebrauch von einem Fahrzeugmodell, welches mit einer Odometrieinformation ergänzt wird. Eine adaptive Lösung detektiert gestörte Signale insbesondere in städtischen Gebieten mit schlechter GPS-Signalqualität.

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich bei meiner Bachelorarbeit und im Studium unterstützt haben.
Dieser Dank gilt zuerst der Techniker Krankenkasse, die mir die Möglickeit gegeben hat, ein duales Studium zu absolvieren und diese Bachelorarbeit extern an der Hochschule zu schreiben.
Für den Vorschlag des spannenden Themas und die intensive Betreuung möchte ich mich besonders bei Prof. Dr. rer. Rasmus Rettig bedanken. Auch gilt der Dank Frau Prof. Dr. rer. Annabella Rauscher-Scheibe, die die Aufgabe des Zweitprüfers übernommen und vielfach zu einer guten Ausbildung beigetragen hat.
Abschließend danke ich meiner Familie und Freunden, die diese Arbeit Korrektur gelesen haben und für mich immer ein verlässlicher Rückhalt sind.

Contents

1 Introduction

Everywhere in the world vehicles are constantly moving from one place to another. While doing so one essential information is always needed: the exact location. A breakthrough technical resolution for this challenge has been the well known Global Positioning System (GPS). Nevertheless its unreliability due to local disturbances or interruptions of satellite-signals and technical limitations in capturing high dynamic movements are still restricting the area of usage and accuracy performance.

1.1 Project BEEdeL

The research project BEEdeL (German abbreviation: “Bewertung des Einsatzes von Elektrobussen mit dezentraler Ladeinfrastuktur”) deals with the development of an autonomous system (“data logger”) for recording movement and environmental data of multiple buses for public transportation in Hamburg. The ultimate goal is a feasibility study for electric buses and an efficient placement of charging stations. Since the beginning of year 2015 more than 20000 hours measurement data have been recorded.

1.2 Motivation and Requirements

Figure 1.1 shows a visualised extract of GPS measurement data in the urban area of Hamburg. Obviously the determined positions are inaccurate compared to the expected travel route of a road vehicle. The data logger has also recorded information about the acceleration, angular rate, magnetic field, air pressure and velocity of the odometer. To utilise this potential for improvement describes the purpose of this thesis. Aiming at the elimination of those GPS-errors and therefore optimisation of reliability and of the accuracy through the inclusion of information derived from all sensors. The goal is to determine the lane where the bus is driving, which requires an accuracy of approximately 2 meters.

1.3 Solution Approach

Based on physical laws all measurements are related to the vehicle’s position in some way. For example the information about acceleration and angular rate enables a calculation of

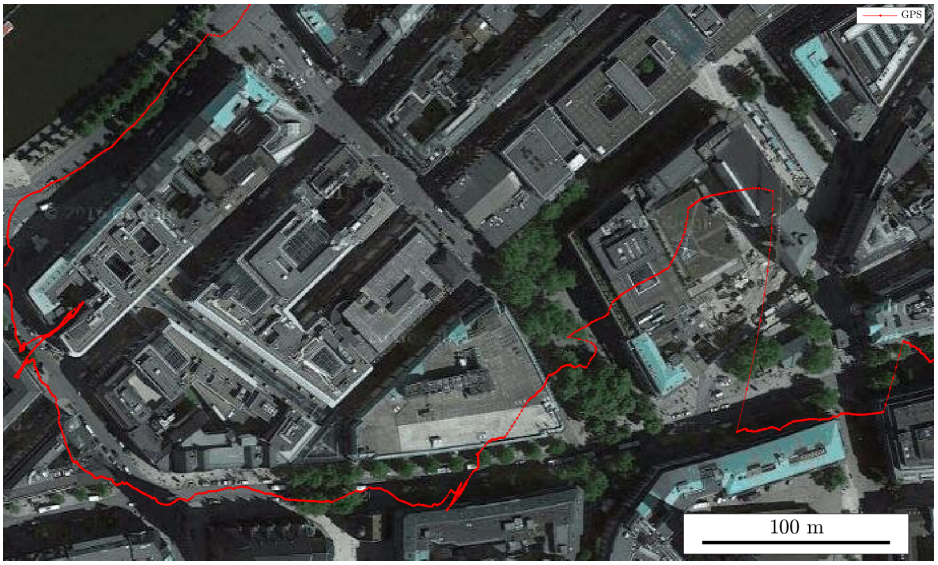


Figure 1.1: Disturbed GPS measurement data

relative distance by applying physical equations of motion. Assuming that these inertial measurements were perfect, this leads to an exact position. In practise technical instruments are not perfect and the inertial position error will grow further gradually. But this is the crucial point to build on. Inertial measurements give a precise relative solution for short time while GPS is more stable in the long term. These complementary error behaviours will restrict each other and eventually result in a smaller position error.

1.4 Structure of Thesis

Chapter 3 presents the mathematical background for processing noisy measurements. It concludes important general definitions and properties of signals providing the theoretical basics for understanding the following concept of Kalman filtering. An example implementation shows the application of the Kalman filter for a simple one dimensional model.

Chapter 4 provides an overview about navigation mathematics in 3-dimensional space and definitions for further coordinate frames.

Chapter 5 describes the technical equipment displaying a detailed overview of the vehicle and all sensors. An outlier detection and a least square temperature compensation algorithm is developed. Furthermore the theory for the vehicle constraints and odometer integration is provided.

Chapter 6 implements the Kalman filter algorithm in three dimensional space. A concep-

tual architecture of an extended Kalman filter is gradually developed. This includes the derivation of the system and measurement models and the configuration of the Kalman filter. A 2D-calibration algorithm for the magnetometer is evolved followed by the innovation filter technique which enables a robust solution for handling outliers in measurement. The chapter concludes with the implementation in software.

Chapter 7 evaluates the results of the algorithm starting with the position solutions which are compared to a reference generated from Google Maps and a height database. Further the innovation filtering and the ability of smoothing outages are illustrated. At last other useful estimates from the algorithm are analysed.

Chapter 8 summarizes the results of this thesis and suggest ideas for further research.

2 List of Abbreviations

BEEdeL German abbreviation: “Bewertung des Einsatzes von Elektrobussen mit dezentraler Ladeinfrastruktur”

CTM coordinate transformation matrix

ECEF Earth centered fixed frame

ECI Earth centered intertial frame

GPS Global Positioning System

HDOP horizontal dilution of precision

IAE innovation based adaptive estimation methods

IMU inertial measurement unit

INS inertial navigation system

LSB least significant bit

MEMS microelectromechanical system

NED North-East-Down-frame

NMEA National Marine Electronics Association

ODR output data rate

PSD power spectral density function

RMS root mean sqaure

RTS Rauch Tung Striebel

SI International System of Units

SPS Standard Positioning Service

VDOP vertical dilution of precision

WGS84 World Geodetic System in 1984

3 Kalman Filter

Information is carried within every single physical interaction or state. For some quantities it is important to be known exactly e.g. for the purpose of generating useful data. Real sensors of technical instruments are always exposed to undesirable quantities like thermal noise or instrument biases. This chapter gives a theoretical background of how to separate the best estimate of true information from those noisy measurements.

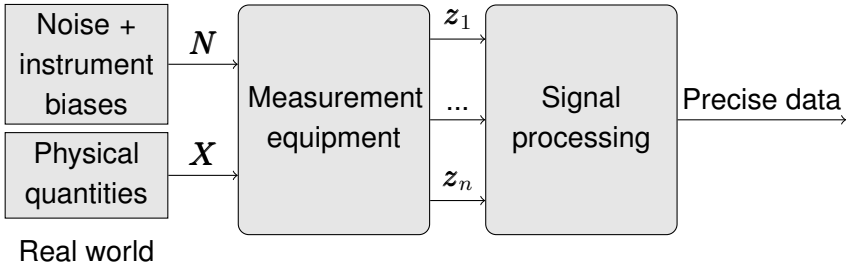


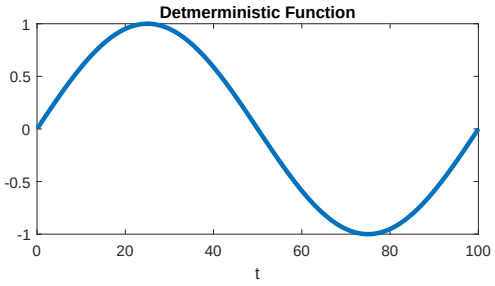
Figure 3.1: Overview measurement processing chain

3.1 Signals

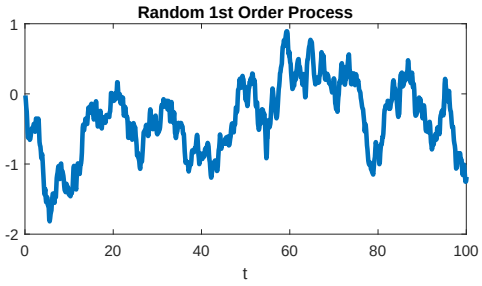
3.1.1 Deterministic and Random Signals

Mathematically two types of signals can be distinguished (Figure 3.2). When a signal is predictable for the timespan of interest it is called deterministic. Consequently a signal is called random or non-deterministic, when its value is not exactly predictable for a given point in time [3, p.57-59].

Physical processes like vehicle movements are of deterministic nature if precise information about all influences is available. In practise signals describing those physical movements are a combination of deterministic signals when the control input is known and random signals including unpredictable effects from winds, vibration, wheel slipping etc.



(a) Sine function: $f(t) = sin(t)$



(b) Random non-deterministic process $X(t)$

Figure 3.2: Types of signals

3.1.2 Properties of Random Signals

Obviously deterministic signals have a big advantage as they are exactly defined with functions for every point in time. This enables analysts to work with them comfortably. Likewise it is desired to achieve a feasible definition for random signals too.

Probability distribution

A way to describe a chance is the concept of random variables: Every physical quantity needs to be assigned to a meaningful rational number for further mathematical processing. By counting the possible outcomes of some random experiment a statistically distribution is created in form of a histogram.

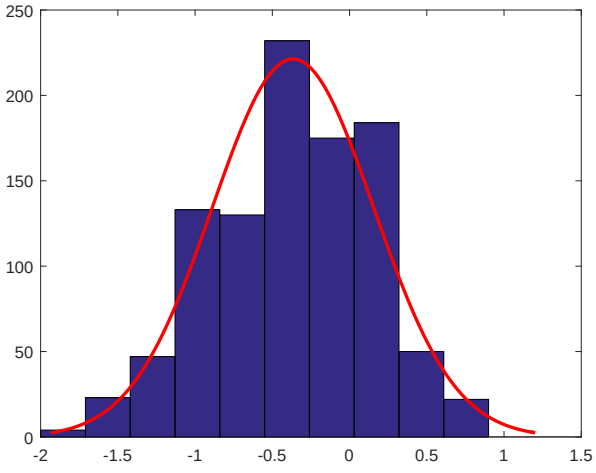


Figure 3.3: Histogram and normal distribution fit

Processes in nature are usually described by a continuous random variable as they have

an infinite number of different outcomes. Then the histogram is created by assigning areas of the outcomes to several discrete values which are called “bins”.

In many situations the profile of the histogram matches to some special probability density functions. This has been the motivation when remembering the lack of a functional definition! One function fitting a lot of natural random experiments is called Gaussian or Normal density function. It is exactly defined by its average value μ and variance σ^2.

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma_X} \cdot \exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right], X \sim N(\mu_X, \sigma_X^2) \tag{3.1}$$

Looking at the prior random 1st-order process all points in the timespan of interest could be defined as a set of random variables $(X_{t_1}, X_{t_2}, ..., X_{t_n})$ with their variate probability densities $f_{X_{t_1}}$, $f_{X_{t_1}X_{t_2}}$, ..., $f_{X_{t_1}X_{t_2}...X_{t_n}}$ in terms of ordinary histograms or matching special functions. Theoretically an infinite number of equal implementations has to be build up and sampled to gather enough information for this general approach. However, this would take infinitely long.

Stationary and Ergodic processes

In order to handle the description of signals in a feasible way, two properties are important.

A process is called strict-sense stationary when all its variate probability densities $f_{X_{t_1}}$, $f_{X_{t_1}X_{t_2}}$, ..., $f_{X_{t_1}X_{t_2}...X_{t_n}}$ are time invariant [3, p.63]. In 3.1.2 a more practical description in form of wide-sense stationarity is proposed.

The definition of ergodicity is dealing with the problem of infinite implementations. A process is said to be ergodic if the average taken at one point of time through these implementations (ensemble average) equals time averaging over one implementation. Fulfilling this condition allows the evaluation of a single signal to determine the distribution characteristics.

An example for a stationary non ergodic process is a random instrument bias, which is constant after initialisation, but changing with every reset [3, p.64].

Mean Value, Mean Square Value and Variance

For a random variable X the mean value (1st moment of X) is defined as follows:

$$E(X) = \int_{-\infty}^{\infty} x f_X(x) dx \tag{3.2}$$

The subsequent properties are valid when the process X is stationary and ergodic. For physical noise processes these conditions can be rarely justified. Thus, heuristic knowledge

is used and simple assumptions are made quite often [3, p.64]. The time average (mean value) for a random signal $X(t)$ is defined as

$$E[X(t)] = \mu_X = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} X(t)dt \tag{3.3}$$

The variance (2nd moment of X) is defined as

$$V[X(t)] = \sigma_X^2 = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} [X(t) - \mu_X]^2 \, dt \tag{3.4}$$

Practically the time for calculating the the mean and variance is restricted. Therefore the calculation of the average from a stationary process is more like a reduction of its variance until a sufficient threshold $\sigma_{\bar{X}}$ is reached. In section 3.1.3 the relation for the variance and the averaging time T is derived.

Correlation

Taking a closer look at the random signal of example 3.2b one can see some dependence between adjoining values. The mean value or variance of the process do not capture this behaviour. A powerful tool for describing the degree of similarity for one signal with itself or another signal at two different times is the correlation function.

$$R_X(t_1, t_2) = E[X(t_1)X(t_2)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_{t_1} x_{t_2} f_{X_{t_1} X_{t_2}}(x_{t_1}, x_{t_2}) dx_{t_1} dx_{t_2} \tag{3.5}$$

A process is wide-sense stationary if the mean-value $E[X(t_1)]$ and variance $E[(X(t_1)-\mu)^2]$ are independent of t_1 and the autocorrelation depends only on the time difference $\tau = t_2 - t_1$ [3, p.68]. In that case

$$R_X(\tau) = E\left[X(t)X(t+\tau)\right] = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} X(t)X(t+\tau)dt = C_X(\tau) + \mu\mu^T \tag{3.6}$$

When evaluating a limited discrete sequence the auto-covariance function is often used to get rid of the mean value [9, Appendix B.42-44]:

$$C_X(\tau) = E[(X(t) - \mu_X)(X(t+\tau) - \mu_X)] = R_X(\tau) - \mu\mu^T \tag{3.7}$$

The cross-correlation function is defined as

$$R_{XY}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} X(t)Y(t+\tau)dt \tag{3.8}$$

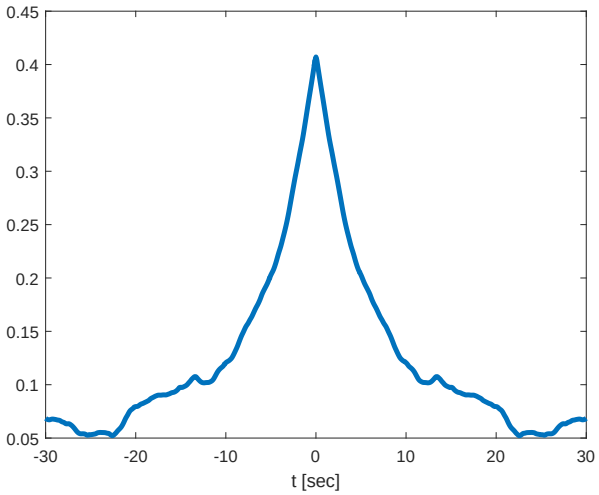


Figure 3.4: Random 1st order process autocorrelation

Power Spectral Density

Obviously the autocorrelation is an indicator for the rate of change. This suggests a promising connection to the frequency domain and it is. The result of applying the Fourier Transformation on the autocorrelation is the power spectral density function (PSD) which is only valid for stationary ergodic signals.

$$\begin{aligned} S_X(j\omega) &= \mathcal{F}\{R_X(\tau)\} = \int_{-\infty}^{\infty} R_X(\tau)\mathrm{e}^{-j\omega\tau}d\tau \\ R_X(\tau) &= \mathcal{F}\{S_X(j\omega)\}^{-1} = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(j\omega)\mathrm{e}^{j\omega\tau}d\omega \end{aligned} \tag{3.9}$$

Within the scope of measurements only positive frequencies can be displayed. Due to energy conservation the single spectrum comprises twice the amplitude of the double sided PSD [9, Appendix B, p.17].

$$S_X^+(j\omega) = \begin{cases} 2S_X(j\omega), & f > 0 \\ S_X(j\omega), & f = 0 \end{cases} \tag{3.10}$$

3.1.3 Gaussian White Noise Process

A fundamental zero mean random stationary process $w(t)$ is white noise. It is totally uncorrelated and by definition jumping around infinitely fast and high [3, p.75]. As a result the autocorrelation is defined by the delta-distribution and the PSD by a constant.

$$R_w(\tau) = E[w_t w_{t+\tau}] = A_S \cdot \delta(\tau) \circ\!\!-\!\!\bullet S_w^+(j\omega) = A_S \cdot 1 \tag{3.11}$$

Gaussian white noise is described by a sequence of zero mean normal distributed random variables with infinite variance. If the spectral amplitude $A_S = 1$, it is called unity white noise w_u.

The whole power is calculated by integrating the PSD-spectrum and results in the auto-correlation value at $\tau = 0$ (Parseval's theorem).

$$P_w = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_X(j\omega) d\omega = R_w(0) = \sigma_w^2 = \infty \tag{3.12}$$

In many data sheets the random characteristic is written down in form of $\frac{[units]}{\sqrt{Hz}}$ describing the square root $\sqrt{A_S} = R_S$ of a constant continuous white-noise PSD-function. In reality no white noise is existing as all physical systems are band limited and no infinite source of power can be provided. Therefore in real applications band limited white noise is assumed.

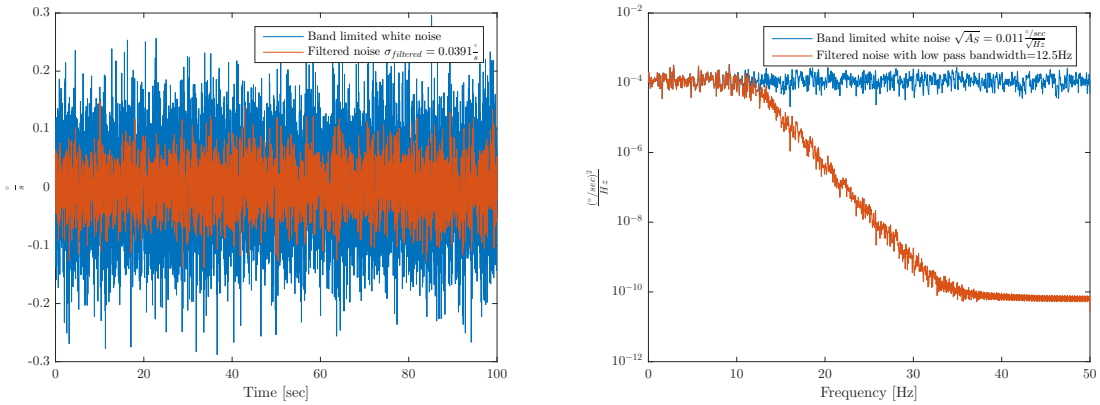
$$S_{wb} = \begin{cases} A_S, & |\omega| \leq \omega_b \\ 0, & |\omega| > \omega_b \end{cases} \tag{3.13}$$

To prevent aliasing effects the sampling-rate has to be more than twice as high as the highest frequency in the signal (Nyquist-Shannon theorem). This can be accomplished with an analogue filter limiting the bandwidth of a signal before the sample and hold circuit [18, p.114/115]. When the bandwidth of a limited white noise signal by far exceeds the cut-off frequency of this filter ($\omega_b > 100 f_{cutoff}$, error < 1%)[3, p.110/111] or the sampling rate [9, Appendix B, p.18] pure white noise can be assumed in further analysis steps. The specification A_s in combination with the bandwidth B in [Hz] of the filter allows an estimated calculation of the standard deviation σ_w^2. From (3.12) the variance is determined by integrating over the whole spectrum. The bandwidth of a filter limits the integration interval for the white noise spectrum. Thus,

$$\hat{\sigma}_w^2 = \frac{1}{2\pi} \int_{-2\pi B}^{2\pi B} S_w(\omega) d\omega = \int_{-B}^{B} S_w(f) df = \int_0^B S_w^+(f) df = A_S \cdot B \tag{3.14}$$

Conversely it is possible to determine A_s from a sampled sequence of white noise which is quite useful for later Kalman filter configuration.

The following example demonstrates the derived theory for the noise of a gyroscope. From the data sheet a noise density $R_S = 0.01 \frac{^\circ/s}{\sqrt{Hz}}$ is given. The noise gets reduced at the output by a configured digital low path filter with cut-off frequency $f_g = B = 12.5Hz$. The PSD and the filtered noise in time domain are illustrated in figure 3.5 using a high order low-pass filter in Matlab. Applying (3.14) leads to an estimated standard deviation of $\hat{\sigma} = R_S \cdot \sqrt{B} = 0.0388 \frac{^\circ}{s}$ which is close to the true standard deviation of $\sigma = 0.0391 \frac{^\circ}{s}$ calculated from the time domain filtered noise signal.



(a) Filtered an unfiltered sampled noise

(b) Single sided power spectral densities

Figure 3.5: Noise in time and frequency domain

Impact of averaging time for white noise variance

In (3.3) the mean value gets calculated by averaging about an infinite amount of time. In reality no infinite data can be provided for this approach. Thus, the averaging time is limited resulting in a not perfect mean in mathematical sense and still having some variance. The following derivation clarifies the relationship between the accuracy of the mean and the averaging time T of averaged white noise $\bar{w}(t)$.

$$\begin{aligned}
\sigma_{\bar{w}}^2 = E[\bar{w}(t)\bar{w}(t)] &= E\left[\frac{1}{T}\int_{-T/2}^{T/2} w(t')dt' \frac{1}{T}\int_{-T/2}^{T/2} w(t'')dt''\right] \\
&= \frac{1}{T^2}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} E[w(t')w(t'')]dt'dt'' \\
&= \frac{1}{T^2}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} A_s\delta(t'-t'')dt'dt'' \\
&= \frac{1}{T^2}\int_{-T/2}^{T/2} A_s dt'' \\
&= \frac{1}{T}A_s
\end{aligned} \tag{3.15}$$

As a result the variance drops by $\frac{1}{T}$ for averaged white noise. This should always be remembered when working with limited means.

3.1.4 Gauss Markov Process

Quite often physical noise is exponentially correlated like in 3.4. This process is then called a Gauss Markov (or first order autoregressive process) [3, p.77].

$$R_X(\tau) = \sigma^2 \mathrm{e}^{-\beta|\tau|} \circ\!\!-\!\!\bullet S_X(j\omega) = \frac{2\sigma^2\beta}{\omega^2+\beta^2} \tag{3.16}$$

$$S_X(j\omega) = \frac{2\sigma^2\beta}{\omega^2+\beta^2} \tag{3.17}$$

A shaping filter is able to create a Gauss Markov process $X(t)$ out of unity white noise $w(t)$. The transfer-function for the filter is determined by the usage of the PSD and the Wiener-Khintchine-theorem [18, p.213]:

$$\begin{aligned} S_X(j\omega) = G(s)\cdot G(-s) &= \frac{2\sigma^2\beta}{\omega^2+\beta^2} \\ G(s) = \frac{X(s)}{W(s)} &= \frac{\sqrt{2\sigma^2\beta}}{s+\beta} \text{ with } \omega_{cutoff} = \beta(4\sigma^2-\beta) \end{aligned} \tag{3.18}$$

The inverse LaPlace transformation gives the differential equation for a Gauss Markov process.

$$\begin{aligned} \dot{x}(t) &= -\beta x(t) + \sqrt{2\sigma^2\beta}w(t) \\ &= -\frac{1}{\tau_c}x(t) + \sigma\sqrt{\frac{2}{\tau_c}}w(t) \end{aligned} \tag{3.19}$$

The correlation time τ_c defines a window of time in which the values of a signal are dependent [8, p.127]. The example noise sequence 3.2b has a correlation time (Figure 3.4) of about 10 seconds.

The Gauss Markov process is stationary and therefore often used for modelling slow varying bounded sensor errors e.g. the change of pressure depending on weather.

3.1.5 Random Walk

The random walk is a sequence of random steps. Continuously these steps will be infinitesimally small mathematically resulting in an integrator driven by white noise [3, p.84].

$$X(t) = \int_0^t w(t')dt' \tag{3.20}$$

The mean of the process is zero but it is non-stationary due to its time dependent variance.

$$\begin{aligned}\sigma^2(t) &= E\left[(X(t)-\mu)^2\right] = E\left[X(t)^2\right] = E\left[\int_0^t w(t')dt' \int_0^t w(t'')dt''\right] \\ &= \int_0^t \int_0^t E[w(t')w(t'')]dt'dt'' = \int_0^t \int_0^t A_S\delta(t'-t'')dt'dt'' \\ &= \int_0^t A_S dt'' = tA_S\end{aligned} \tag{3.21}$$

Figure 3.6 visualizes the development of the standard deviation σ which is analogue to the error in velocity resulting from the noise in an acceleration signal. Also many other processes can be approximately modelled by a random walk with the advantage of tuning only one parameter A_S compared to two of the Gauss Markov process.

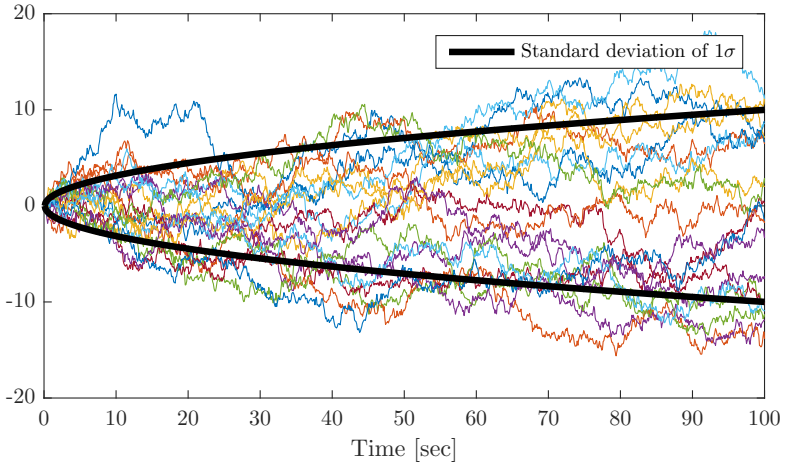


Figure 3.6: Random walk driven by white noise with $A_s = 1$

3.1.6 Random Constant

A random constant is similar to a random walk but without white noise as an input. Instead it has an initial uncertainty σ. The autocorrelation function is constant. Static states can be modelled by this process e.g. a lever arm of a rigid body.

3.2 Derivation of Kalman Filter Equations

In this chapter the general purpose of evaluating noisy measurement data has been introduced. The central problem is how to separate signals containing the information about

specific quantities from additive disturbances like noise or biases [3, p.141]. The Kalman Filter solves this in an optimal way using physical connections between desired information and technical properties of the measurement equipment. It takes advantage of simple Gaussian white noise properties and minimises the mean-squared estimation error (Covariance) from the estimated states (Mean) in a linear dynamic system with white system and white measurement noise [8, p.1]. Especially when using different measurement techniques faulty parts of the signals will be observable and can be removed. It is a Bayesian estimation technique [9, p.82]. Figure 3.7 illustrates the principle of Kalman filtering to simulate the real world systems and measurements with linear system and sensor models.

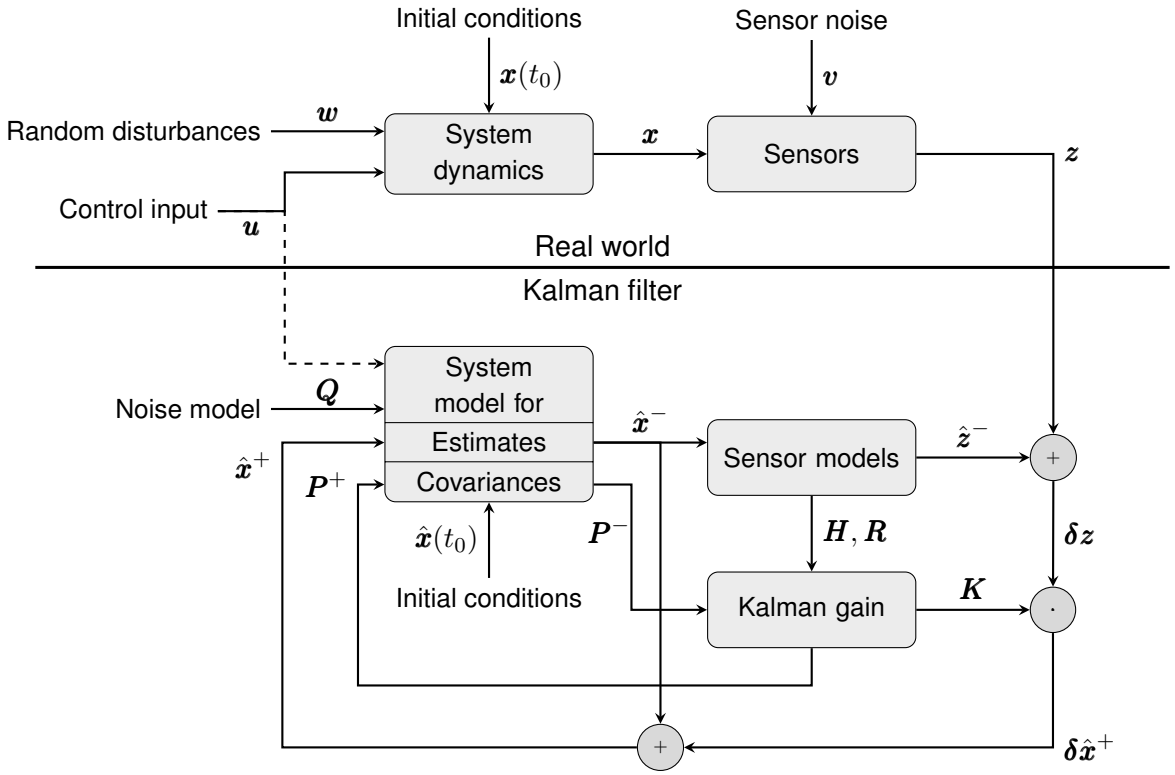


Figure 3.7: Real systems and modelled systems in a Kalman filter (inspired by [8, p.38])

3.2.1 State Space Modeling

All physical states are the result of changed initial conditions. Differential equations express this fact in a mathematical way. A general time varying continuous-time model represents all possible equations.

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{G}(t)\boldsymbol{w}(t) \\ \boldsymbol{z}(t) &= \boldsymbol{h}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{v}(t) \end{aligned} \tag{3.22}$$

where:

$\boldsymbol{x}$: state vector (n x 1)

$\boldsymbol{f}$: non-linear system function

$\boldsymbol{w}$: system noise vector (n x 1)

$\boldsymbol{G}$: system noise distribution matrix (n x n)

$\boldsymbol{z}$: measurement vector (m x 1)

$\boldsymbol{h}$: non-linear measurement function

$\boldsymbol{v}$: measurement noise vector (m x 1)

This kind of dynamic state model can contain non-linear properties which are difficult to handle in analysis and calculations. Therefore further methods make use of a linear time variant state space model. The control input $\boldsymbol{u}$ is used for noiseless system inputs. In this thesis all signals are disturbed by noise and rather handled as measurements. Thus $\boldsymbol{u}$ is neglected in further steps.

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{G}(t)\boldsymbol{w}(t) \\ \boldsymbol{z}(t) &= \boldsymbol{H}(t)\boldsymbol{x}(t) + \boldsymbol{v}(t) \end{aligned} \tag{3.23}$$

where:

$\boldsymbol{x}$: state vector (n x 1)

$\boldsymbol{F}$: system matrix (n x n)

$\boldsymbol{w}$: system noise vector (n x 1)

$\boldsymbol{G}$: system noise distribution matrix (n x n)

$\boldsymbol{z}$: measurement vector (m x 1)

$\boldsymbol{H}$: measurement matrix (m x n)

$\boldsymbol{v}$: measurement noise vector (m x 1)

3.2.2 Discrete System and Measurement Model

Continuous information require an infinite amount of storage. No digital system is able to achieve this. Therefore the continuous model 3.23 is transformed into a discrete system. Many Kalman-Filter books do not derive the following exponential approach [24, p.5], which is fundamentally important for all further steps. The system matrix $\boldsymbol{F}$ and system noise distribution matrix $\boldsymbol{G}$ will be assumed constant during the propagation interval $\tau = t_k - t_{k-1}$.

$$\begin{aligned}
\dot{\boldsymbol{x}}(t) &= \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{G}\boldsymbol{w}(t) \\
\dot{\boldsymbol{x}}(t) - \boldsymbol{F}\boldsymbol{x}(t) &= \boldsymbol{G}\boldsymbol{w}(t) \\
e^{-\boldsymbol{F}t}\dot{\boldsymbol{x}}(t) - e^{-\boldsymbol{F}t}\boldsymbol{F}\boldsymbol{x}(t) &= e^{-\boldsymbol{F}t}\boldsymbol{G}\boldsymbol{w}(t) \\
\frac{d}{dt}e^{-\boldsymbol{F}t}\boldsymbol{x}(t) &= e^{-\boldsymbol{F}t}\boldsymbol{G}\boldsymbol{w}(t) \\
\int_{t_{k-1}}^{t_k} \frac{d}{d\tau}e^{-\boldsymbol{F}\tau}\boldsymbol{x}(\tau)d\tau &= \int_{t_{k-1}}^{t_k} e^{-\boldsymbol{F}\tau}\boldsymbol{G}\boldsymbol{w}(\tau)d\tau \\
e^{-\boldsymbol{F}t_k}\boldsymbol{x}(t_k) - e^{-\boldsymbol{F}t_{k-1}}\boldsymbol{x}(t_{k-1}) &= \int_{t_{k-1}}^{t_k} e^{-\boldsymbol{F}\tau}\boldsymbol{G}\boldsymbol{w}(\tau)d\tau \\
e^{-\boldsymbol{F}t_k}\boldsymbol{x}(t_k) &= e^{-\boldsymbol{F}t_{k-1}}\boldsymbol{x}(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{-\boldsymbol{F}\tau}\boldsymbol{G}\boldsymbol{w}(\tau)d\tau \\
\boldsymbol{x}(t_k) &= e^{\boldsymbol{F}\cdot(t_k - t_{k-1})}\boldsymbol{x}(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{\boldsymbol{F}\cdot(t_k-\tau)}\boldsymbol{G}\boldsymbol{w}(\tau)d\tau
\end{aligned} \tag{3.24}$$

Finally a solution emerges as the sum of an initial condition part and particular solution also called driven response part [3, p.123].

$$\boldsymbol{x}(t_k) = \underbrace{\boldsymbol{\phi}(t_k, t_{k-1})\boldsymbol{x}(t_{k-1})}_{\text{Initial condition solution}} + \underbrace{\int_{t_{k-1}}^{t_k} \boldsymbol{\phi}(t_k, \tau)\boldsymbol{G}(\tau)\boldsymbol{w}(\tau)d\tau}_{\text{Particular solution } \boldsymbol{w}_{k-1}} \tag{3.25}$$

The transition matrix $\boldsymbol{\phi}_{k-1}$ originates from the initial condition part where the system matrix is $\boldsymbol{F} = \boldsymbol{F}(t_k - \tau_s) = \boldsymbol{F}_{k-1}$. The exponential function with matrix argument can be evaluated by its power-series. For small τ_s in most cases a first order approximation (truncation after the second addend) is sufficient and sometimes exact.

$$\boldsymbol{\phi}_{k-1} = e^{\boldsymbol{F}_{k-1}\tau_s} = \sum_{r=0}^{\infty} \frac{\boldsymbol{F}_{k-1}^r \tau_s}{r!} \approx \boldsymbol{I} + \boldsymbol{F}_{k-1}\tau_s \tag{3.26}$$

For the white noise vector w_{k-1} the system noise distribution matrix is defined as $\boldsymbol{G} = \boldsymbol{G}(t_k - \tau_s) = \boldsymbol{G}_{k-1}$.

$$\boldsymbol{w}_{k-1} = \int_{t_k-\tau_s}^{t_k} \boldsymbol{\phi}(t_{k+1}, \tau)\boldsymbol{G}_{k-1}\boldsymbol{w}(\tau)d\tau \tag{3.27}$$

The measurements are sampled so the measurement vector from 3.23 can be directly discretised with $t = t_k$. Then the whole sampled continuous-time system is given by

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{\phi}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1} \\ \boldsymbol{z}_k &= \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k \end{aligned} \tag{3.28}$$

where:

$\boldsymbol{x}_k$: state vector (n x 1)

$\boldsymbol{\phi}_{k-1}$: system matrix (n x n)

$\boldsymbol{w}_{k-1}$: system noise vector (n x 1)

$\boldsymbol{z}_k$: measurement vector (m x 1)

$\boldsymbol{H}_k$: measurement matrix (m x n)

$\boldsymbol{v}_k$: measurement noise vector (m x 1)

3.2.3 Phases of Kalman Filter Algorithm

The algorithm is implemented in a closed loop running through all desired points in time illustrated in figure 3.8. It can be divided into three phases. The process starts with initial conditions $\hat{\boldsymbol{x}}_0^-$ and $\boldsymbol{P}_0^-$.

Kalman Gain

As the Kalman gain represents the fundamental concept of Kalman filtering its full derivation follows in the next steps. The notation originates from two books [3, p.144/145] and [9, p.82-103] combining most of the advantages. Some steps are explained in greater detail.

The basic idea is the assumption of white uncorrelated noise for the system noise $\boldsymbol{w}_k$ and measurement noise $\boldsymbol{v_k}$.

$$\begin{aligned} E[\boldsymbol{w}_k\boldsymbol{w}_i] &= \begin{cases} \boldsymbol{Q}_k, & i = k \\ \boldsymbol{0}, & i \neq k \end{cases} \\ E[\boldsymbol{v}_k\boldsymbol{v}_i] &= \begin{cases} \boldsymbol{R}_k, & i = k \\ \boldsymbol{0}, & i \neq k \end{cases} \\ E[\boldsymbol{w}_k\boldsymbol{v}_i] &= \boldsymbol{0} \; \forall \; i, k \end{aligned} \tag{3.29}$$

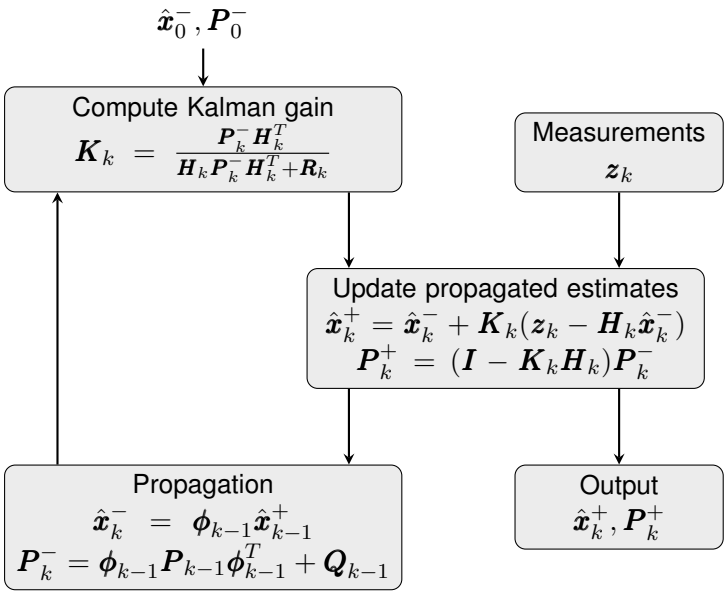


Figure 3.8: Kalman filter loop [3, p.147]

In this section an optimal solution for weighting between propagated system states $\hat{\boldsymbol{x}}_k^-$ and noisy measurements $\boldsymbol{z}_k$ with a factor $\boldsymbol{K}_k$ is derived.

$$\begin{aligned} \hat{\boldsymbol{x}}_k^+ &= \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^-) \\ &= \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \delta \boldsymbol{z}_k^- \end{aligned} \tag{3.30}$$

where:

$\hat{\boldsymbol{x}}_k^+$: updated state vector

$\hat{\boldsymbol{x}}_k^-$: predicted state vector

$\boldsymbol{K}_k$: linear weighting factor of measurements called Kalman gain

$\delta \boldsymbol{z}_k^-$: measure innovation

The updated error covariance $\boldsymbol{P}_k^+$ represents the Gaussian squared standard deviation for the true state $\boldsymbol{x}_k$ around its mean value given by the updated state state vector $\hat{\boldsymbol{x}}_k^+$.

$$\boldsymbol{e}_k^+ = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^+ \tag{3.31}$$

$$\boldsymbol{P}_k^+ = E[\boldsymbol{e}_k \boldsymbol{e}_k^T] = E[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^+)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^+)^T] \tag{3.32}$$

The same applies for the predicted state vector.

$$\boldsymbol{e}_k^- = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^- \tag{3.33}$$

$$\boldsymbol{P}_k^- = E[\boldsymbol{e}_k^- \boldsymbol{e}_k^{-T}] = E[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-)^T] \tag{3.34}$$

Substituting from 3.30 and 3.28 gives

$$\begin{aligned}
\boldsymbol{P}_k^+ &= E\left[\left(\boldsymbol{x}_k - \left(\hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-)\right)\right)\left(\boldsymbol{x}_k - \left(\hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-)\right)\right)^T\right] \\
&= E\left[\left(\boldsymbol{x}_k - \left(\hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-)\right)\right)\left(\boldsymbol{x}_k - \left(\hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-)\right)\right)^T\right] \\
&= E\left[\left(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^- - \boldsymbol{K}_k\boldsymbol{H}_k(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-) - \boldsymbol{K}_k\boldsymbol{v}_k\right)\left(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^- - \boldsymbol{K}_k\boldsymbol{H}_k(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-) - \boldsymbol{K}_k\boldsymbol{v}_k\right)^T\right] \\
&= E\left[\left((\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-) - \boldsymbol{K}_k\boldsymbol{v}_k\right)\left((\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-) - \boldsymbol{K}_k\boldsymbol{v}_k\right)^T\right] \\
&= E\left[\left((\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)(\boldsymbol{e}_k^-) - \boldsymbol{K}_k\boldsymbol{v}_k\right)\left((\boldsymbol{e}_k^-)^T(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T - \boldsymbol{v}_k^T\boldsymbol{K}_k^T\right)\right] \\
&= E\left[(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{e}_k^-\boldsymbol{e}_k^{-T}(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T - (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{e}_k^-(\boldsymbol{v}_k^T\boldsymbol{K}_k^T)\right. \\
&\qquad \left. -\boldsymbol{K}_k\boldsymbol{v}_k\boldsymbol{e}_k^-T(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + \boldsymbol{K}_k\boldsymbol{v}_k\boldsymbol{v}_k^T\boldsymbol{K}_k^T\right]
\end{aligned} \tag{3.35}$$

The predicted state vector error $\boldsymbol{e}_k^-$ is uncorrelated with the measurement noise (3.29). Therefore

$$\begin{aligned}
\boldsymbol{P}_k^+ &= E\left[(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{e}_k^-\boldsymbol{e}_k^{-T}(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T\right] + E\left[\boldsymbol{K}_k\boldsymbol{v}_k\boldsymbol{v}_k^T\boldsymbol{K}_k^T\right] \\
&= (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)E\left[\boldsymbol{e}_k^-\boldsymbol{e}_k^{-T}\right](\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + \boldsymbol{K}_kE\left[\boldsymbol{v}_k\boldsymbol{v}_k^T\right]\boldsymbol{K}_k^T \\
&= (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^-(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + \boldsymbol{K}_kR_k\boldsymbol{K}_k^T \\
&= \boldsymbol{P}_k^- - \boldsymbol{K}_k\boldsymbol{H}_k\boldsymbol{P}_k^- - \boldsymbol{P}_k^-\boldsymbol{H}_k^T\boldsymbol{K}_k^T + \boldsymbol{K}_k\left(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{R}_k\right)\boldsymbol{K}_k^T
\end{aligned} \tag{3.36}$$

The state variance is given by the diagonal of $\boldsymbol{P}_k^+$. A minimization procedure (3.38) of this diagonal will result in the optimal Kalman gain with help of two matrix differentiation formulas.

$$\begin{aligned}
\frac{d}{d\boldsymbol{A}}\left[diag(\boldsymbol{A}\boldsymbol{B}\right] &= \boldsymbol{B}^T \qquad (\boldsymbol{A}, \boldsymbol{B} = n \times n) \\
\frac{d}{d\boldsymbol{A}}\left[diag(\boldsymbol{A}\boldsymbol{C}\boldsymbol{A}^{\boldsymbol{T}}\right] &= 2\boldsymbol{A}\boldsymbol{C} \qquad (\boldsymbol{C} = symmetric)
\end{aligned} \tag{3.37}$$

$$\begin{aligned}
\frac{d}{d\boldsymbol{K}_k}\left[diag(\boldsymbol{P}_k^+)\right] &= 0 \\
\frac{d}{d\boldsymbol{K}_k}\left[diag\left(\boldsymbol{P}_k^- - \boldsymbol{K}_k\boldsymbol{H}_k\boldsymbol{P}_k^- - \boldsymbol{P}_k^-\boldsymbol{H}_k^T\boldsymbol{K}_k^T + \boldsymbol{K}_k\left(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{R}_k\right)\boldsymbol{K}_k^T\right)\right] &= 0 \\
-2(\boldsymbol{H}_k^-\boldsymbol{P}_k)^T - 2\boldsymbol{K}_k(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{R}_k) &= 0
\end{aligned} \tag{3.38}$$

Solving equation 3.38 for the Kalman gain leads to

$$\boldsymbol{K}_k = \frac{\boldsymbol{P}_k^- \boldsymbol{H}_k^T}{\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k} \tag{3.39}$$

Update Step

The update for the state vector has been the first approach for the Kalman Gain calculation. Repeating 3.30

$$\begin{aligned} \hat{\boldsymbol{x}}_k^+ &= \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^-) \\ &= \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \delta \boldsymbol{z}_k^- \end{aligned}$$

Substituting $\boldsymbol{K}_k$ from 3.39 into equation 3.36 gives the update formula for the state-covariance matrix

$$\begin{aligned} \boldsymbol{P}_k^+ &= \boldsymbol{P}_k^- - \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_k^- - \boldsymbol{P}_k^- \boldsymbol{H}_k^T \boldsymbol{K}_k^T + \overbrace{\boldsymbol{K}_k \left(\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k\right)}^{(3.39):\ \boldsymbol{P}_k^- \boldsymbol{H}_k^T = \boldsymbol{K}_k \left(\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k\right)} \boldsymbol{K}_k^T \\ &= \boldsymbol{P}_k^- - \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_k^- - \boldsymbol{P}_k^- \boldsymbol{H}_k^T \boldsymbol{K}_k^T + \boldsymbol{P}_k^- \boldsymbol{H}_k^T \boldsymbol{K}_k^T \\ &= (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_k^- \end{aligned} \tag{3.40}$$

Propagation Step

During the propagation the prior state estimate is predicted by the physical system implemented in the transition matrix for the time interval τ_s. When working with closed loop feedback error states 3.41 can be skipped.

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{\phi}_{k-1} \hat{\boldsymbol{x}}_{k-1}^+ \tag{3.41}$$

The prior error covariance is always predicted [9, p.99].

$$\begin{aligned} \boldsymbol{P}_k^- &= E\left[(\hat{\boldsymbol{x}}_k^- - \boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k^- - \boldsymbol{x}_k)^T\right] \\ &= \boldsymbol{\phi}_{k-1} \boldsymbol{P}_{k-1} \boldsymbol{\phi}_{k-1}^T + \boldsymbol{Q}_{k-1} \end{aligned} \tag{3.42}$$

A pracitcal approach for the state matrix has been proposed in 3.26. The analytical and practical determination of the driven noise covariance matrix $\boldsymbol{Q}_{k-1}$ is even more rough.

By substituting 3.27 for the discrete white noise, the noise covariance matrix is generally

defined as

$$\begin{aligned}\boldsymbol{Q}_{k-1} &= E\left[\boldsymbol{w}_{k-1}\boldsymbol{w}_{k-1}^T\right] \\ &= \int_{t_k-\tau_s}^{t_k}\int_{t_k-\tau_s}^{t_k} \boldsymbol{\phi}(t_{k-1},t')\boldsymbol{G}(t')E[\boldsymbol{w}(t')\boldsymbol{w}^T(t'')]\boldsymbol{G}^T(t'')\boldsymbol{\phi}^T(t_{k-1},t'')dt'dt''\end{aligned} \tag{3.43}$$

If the system noise is white (3.11) applying $E[\boldsymbol{w}(t')\boldsymbol{w}^T(t'')] = \boldsymbol{A_S}\delta(t'-t'')$ gives

$$\boldsymbol{Q}_{k-1} = \int_{t_k-\tau_s}^{t_k} \boldsymbol{\phi}(t_{k-1},t')\boldsymbol{G}(t')\boldsymbol{A_S}\boldsymbol{G}^T(t'')\boldsymbol{\phi}^T(t_{k-1},t'')dt' \tag{3.44}$$

The off diagonal elements of the matrix $\boldsymbol{A_S}$ are zero. As a further simplification the time propagation over the transition matrix can be neglected resulting in the so-called Impulse approximation [9, p.99].

$$\boldsymbol{Q}_{k-1} \approx \boldsymbol{Q}'_{k-1} = \boldsymbol{G}_{k-1}\boldsymbol{A_S}\boldsymbol{G}_{k-1}^T\tau_s \tag{3.45}$$

Another more accurate numerical approach to determine the discrete transition matrix $\boldsymbol{\phi}_k$ and system noise covariance matrix $\boldsymbol{Q}_k$ is the Van Loan method [15].

3.2.4 Backward Smoother

So far only the information of measurements prior to the time of the estimated state vector has been used. Smoothing deals with the optimal estimation problem considering subsequent information in time. Mainly three types of smoothing techniques are applied depending on the degree of real time requirements and purpose [8, p.241].

Fixed Point Smoother optimize the estimate for one point in time by using measurement data before and after t_{fixed} resulting in a function as a predictor for $t < t_{fixed}$,a filter for $t = t_{fixed}$ and a smoother for $t > t_{fixed}$. The application is mainly the estimation of initial conditions for stationary states e.g. the initial fine alignment of a precise inertial navigation system by measuring the Coriolis-force [8, p.266].

Fixed-Interval Smoothers give the best smoothed estimate by evaluating all data in a defined time-interval e.g. using the Rauch Tung Striebel (RTS)-algorithm [11]. They are quite valuable for post-processing applications [8, p.244]. In this thesis an interval smoother is implemented.

Fixed-Lag Smoothers are running in real-time giving an smoothed estimate using all measurements of the time interval $t_{start} \leq t_{meas} \leq t_{est} + \Delta t_{lag}$ [8, p.256]. The RTS-algorithm is adaptable for this purpose. An even better approach is to augment the state vector $[\boldsymbol{x}_k]$ by a tuple of delayed state vector estimates $[\boldsymbol{x}_{k-1}, \boldsymbol{x}_{k-2}, ...\boldsymbol{x}_{k-N}]$.

This leads to a form which can be processed by the previous derived Kalman algorithm giving estimates in real time which will be sequentially optimized with further measurements [3, p.214-216].

The RTS-algorithm starts after a complete run of a Kalman filter which has processed measurements and stored the following vectors and matrices for each iteration: $\boldsymbol{x}_k^+, \boldsymbol{x}_k^-, \boldsymbol{P}_k^+, \boldsymbol{P}_k^-, \boldsymbol{\phi}_k$. When these information are available 3.46, 3.47 and 3.48 are processed backwards from $k_N, k_{N-1}...k_0$ noting that processing 3.48 is not required for the smoothed estimate $\hat{\boldsymbol{x}}_k^s$.

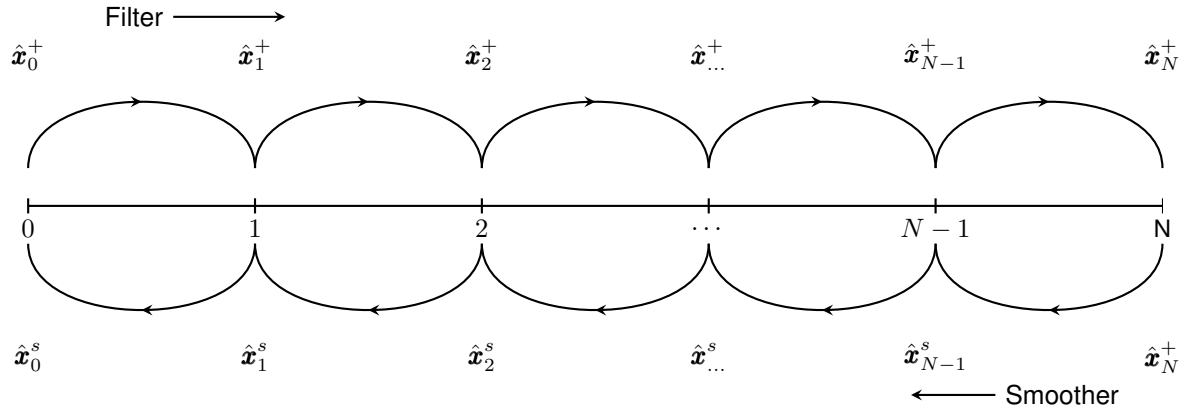


Figure 3.9: Discrete fixed interval smoothing (inspired by [3, p.208])

$$\boldsymbol{A}_k = \frac{\boldsymbol{P}_k^+ \boldsymbol{\phi}_k^T}{\boldsymbol{P}_{k+1}^-} \tag{3.46}$$

$$\hat{\boldsymbol{x}}_k^s = \hat{\boldsymbol{x}}_k^+ + \boldsymbol{A}_k(\hat{\boldsymbol{x}}_{k+1}^s - \hat{\boldsymbol{x}}_{k+1}^-) \tag{3.47}$$

$$\boldsymbol{P}_k^s = \boldsymbol{P}_k^+ + \boldsymbol{A}_k(\boldsymbol{P}_{k+1}^s - \boldsymbol{P}_{k+1}^-)\boldsymbol{A}_k^T \tag{3.48}$$

3.3 Simple Example: One Dimensional Movement

After all these general definitions a simple example will demonstrate the effect of Kalman filtering. Additionally the concept of error states is introduced.

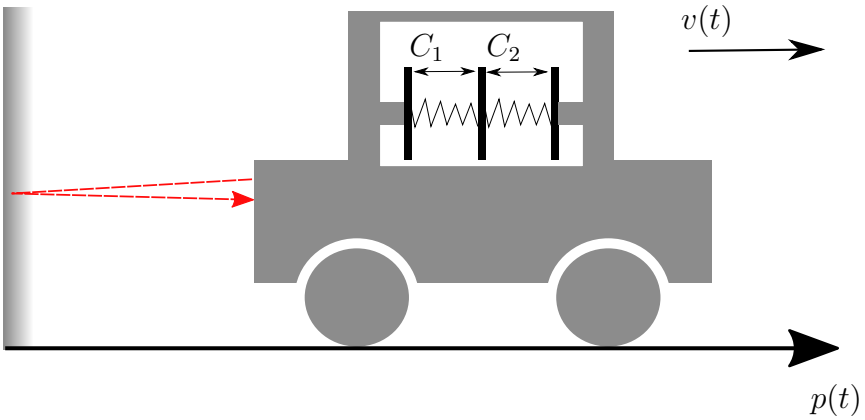


Figure 3.10: One-dimensional rail vehicle with instruments

3.3.1 Setup Description

The movement of a straight forward driving rail vehicle can be described with one dimensional equations, neglecting all kinds of friction.

$$\begin{aligned} \frac{dp(t)}{dt} &= v(t) \\ \frac{dv(t)}{dt} &= a(t) \end{aligned} \tag{3.49}$$

Two complementary low-cost measurement systems are used:

- A Laser sends out a light impulse towards a mirror and starts a stopwatch simultaneously. A light-sensitive receiver detects the reflected signal and evaluates the time. The distance to the mirror is calculated by multiplying the speed of light and the passed time divided by two.
  This technique is similar to the principle of ranging measurements of GPS-Receivers and belongs to the category of position fixing instruments.

- A single accelerometer measures the force on one conducting plate which is mounted with springs between two other plates. When the vehicle accelerates forward the intermediate plate will react inertially to the force defined by Newton's laws of motion. Therefore the gaps between the plates will vary temporally and likewise the capacities C_1, C_2. Electric devices can detect the size of changed capacity which is analogue to the acceleration. Due to temperature variations and fabrication inaccuracies the middle plate is constantly shifted to the left which is called bias. This results in a constant acceleration error.

An accelerometer is an inertial sensor and belongs to the class of Dead Reckoning instruments [9, p.5] as it can only measure the change in position.

3.3.2 Concept of Error State Kalman Filter and Closed Loop Approach

With the knowledge of previous sections one would probably choose the state variables for this example as the position, velocity and acceleration connected by linear differential equations of 3.49.

$$\dot{\boldsymbol{x}}_{total}(t) = \begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{a} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{F}_{total}} \underbrace{\begin{bmatrix} p \\ v \\ a \end{bmatrix}}_{\boldsymbol{x}_{total}} + \boldsymbol{G}_{total}(t)\boldsymbol{w}_{total}(t) \tag{3.50}$$

Additionally states describing the measurement system, in this case the accelerometer bias, have to be modelled. A random walk model with very low noise input w_{bias}, accounting for offset drifts due to temperature changes, is able to model the bias characteristics in the system.

$$\dot{\boldsymbol{x}}_{error}(t) = \begin{bmatrix} \delta\dot{p} \\ \delta\dot{v} \\ \delta\dot{b}_a \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{F}_{error}} \underbrace{\begin{bmatrix} \delta p \\ \delta v \\ \delta b_a \end{bmatrix}}_{\boldsymbol{x}_{error}} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{\boldsymbol{G}_{error}} \underbrace{\begin{bmatrix} w_{bias} \end{bmatrix}}_{\boldsymbol{w}_{error}} \tag{3.51}$$

Equation 3.50 and 3.51 can be combined to

$$\dot{\boldsymbol{x}}_{total}(t) = \begin{bmatrix} \dot{\boldsymbol{x}}_{total} \\ \dot{\boldsymbol{x}}_{error} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_{total} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_{error} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{total} \\ \boldsymbol{x}_{error} \end{bmatrix} + \begin{bmatrix} \boldsymbol{G}_{total} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_{error} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_{total} \\ \boldsymbol{w}_{error} \end{bmatrix} \tag{3.52}$$

Integrating the acceleration and calculating the laser-distance leads to four measurements.

$$\begin{aligned} \boldsymbol{z}_{acc}(t) &= \begin{bmatrix} z_p \\ z_v \\ z_a \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{total} \\ \boldsymbol{x}_{error} \end{bmatrix} + \boldsymbol{v}_{acc} \\ z_{p,laser} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{total} \\ \boldsymbol{x}_{error} \end{bmatrix} + v_{laser} \end{aligned} \tag{3.53}$$

In doing so the movement would be described by a Gaussian random variable. Since its random characteristic is constantly changing a manageable process for vehicle's motion is difficult to find. For this kind of situation the "Go Free" concept is useful: States can be relieved of any assumption about their statistics if measurement redundancy is given (e.g. complementary sensors). This can be done by boosting the noise covariance $\boldsymbol{Q}$ with a

high amount of system noise for the total states [3, p.284-285]. Another approach with the same effect is the Error-State Kalman filter which makes use of the differencing-out idea [3, p.290].

The error system model is almost given by equation 3.51, which is a quarter of the matrix-size compared to the total system. Instead of four there will be one measurement containing only the error quantities.

$$\delta z = z_{p,laser} - \hat{z}_{p,acc} = z_{p,laser} - \hat{p} = p_{true} + \delta p_{laser} - (p_{true} - \delta p) = \delta p + \delta p_{laser} \quad (3.54)$$

Figure 3.11 visualises the new role of the accelerometer as a total state extension algorithm (later called reference system) for the error state Kalman filter. This setting is more efficient and can be more precise in case of non-linear systems compared to a total state Kalman filter. Additionally the accelerometer solution won't drift due to the closed loop corrections.

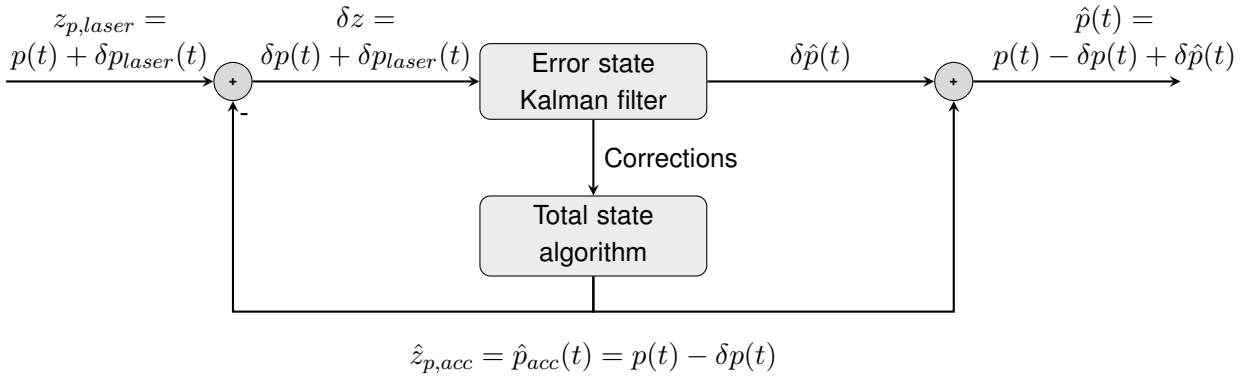


Figure 3.11: Error state Kalman filter integration

The Kalman filter separates the error $\delta p(t)$ of the accelerometer from the residual measurement δz (later called measurement innovation) containing both errors. The system noise must be extended by w_{noise} to consider the missing total state acceleration measurement noise v_{acc} which leads to the final error state system model.

$$\boldsymbol{\delta\dot{x}}(t) = \begin{bmatrix} \delta\dot{p} \\ \delta\dot{v} \\ \delta\dot{b}_a \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{F}} \underbrace{\begin{bmatrix} \delta p \\ \delta v \\ \delta b_a \end{bmatrix}}_{\boldsymbol{\delta x}} + \underbrace{\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\boldsymbol{G}} \underbrace{\begin{bmatrix} w_{noise} \\ w_{bias} \end{bmatrix}}_{\boldsymbol{w}} \quad (3.55)$$

$$z_{p,laser}(t) - \hat{p}^{-}_{acc}(t) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_{\boldsymbol{H}} \boldsymbol{x} + \boldsymbol{v}_{laser} \quad (3.56)$$

3.3.3 Discrete algorithm

The continuous system 3.55 is transformed into its discrete counterpart by applying 3.26.

$$E[\boldsymbol{\delta x}_k] = \boldsymbol{\delta \hat{x}}_k = \begin{bmatrix} \delta p_k \\ \delta v_k \\ \delta b_{b,k} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \tau_s & 0 \\ 0 & 1 & \tau_s \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{\phi}_{k-1}} \underbrace{\begin{bmatrix} \delta p_{k-1} \\ \delta v_{k-1} \\ \delta b_{a,k-1} \end{bmatrix}}_{\boldsymbol{\delta \hat{x}}_{k-1}} \tag{3.57}$$

The laser-range equipment uses a low-cost components. Therefore the noise in the calculated position is large resulting in a standard deviation of $\sigma_p = 100m$.

$$\delta z_k^- = z_{p,k} - \hat{p}_k^- = \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_{\boldsymbol{H}_k} \boldsymbol{\delta x_k} \qquad R_k = \sigma_p^2 \tag{3.58}$$

The noise of the accelerometer from the data sheet is specified by $R_{S,noise} = 220\frac{\mu g}{\sqrt{Hz}} = 0.0022\frac{m/s^2}{\sqrt{Hz}}$. In order to give the bias state some stochastic flexibility a tiny Root-PSD-level $R_{S,bias} = 1\mu\frac{m/s^2}{\sqrt{Hz}}$ is assumed. This results in a very slow growing variance of the bias state with respect to time to account for temperature drifts. With the impulse approximation (3.45) the noise covariance matrix results in

$$\boldsymbol{Q}_{k-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & A_{S,noise} & 0 \\ 0 & 0 & A_{S,bias} \end{bmatrix} \tau_s \qquad A_S = R_S^2 \tag{3.59}$$

A flow chart diagram 3.12 is created combining all previous equations. Afterwards a script processes sub sequentially all steps of the flow chart diagram. In this thesis Matlab is used as the software environment, because it is quite efficient in programming matrices calculations.

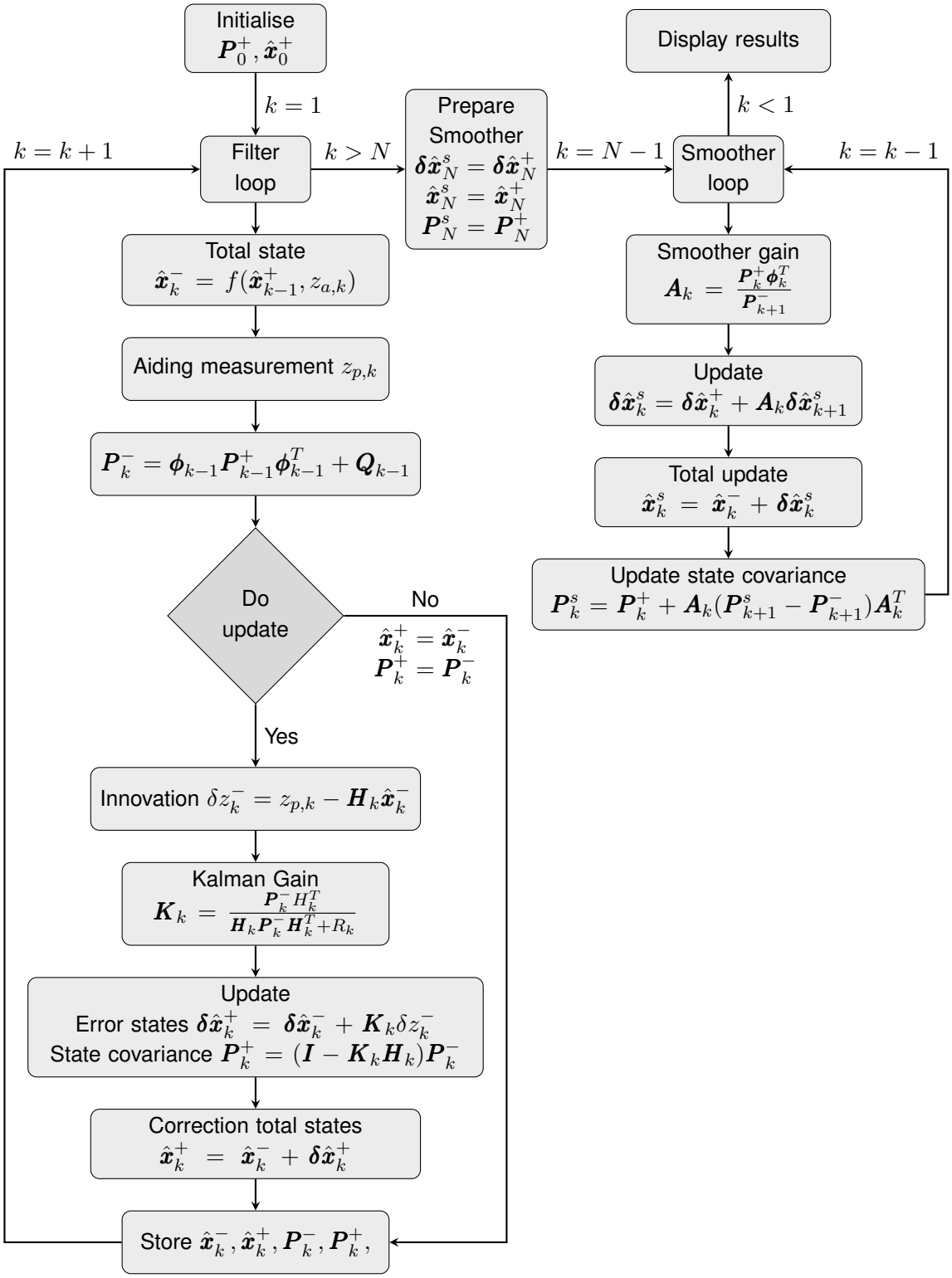


Figure 3.12: Flow chart diagram filter and smoother

3.3.4 Monte-Carlo Simulation

Monte-Carlo simulations are especially helpful in testing stochastic algorithms that are difficult to analyse in their performance with direct mathematical methods. A statistical experiment is artificially created by generating random sequences as inputs for the algorithm [3, p.128].

First of all a deterministic trajectory serves as a reference. Out of theses reference data combined with random sequences artificial measurements can be produced matching the realistic sensor output as good as possible. Figure 3.13 shows the result for the noisy acceleration and laser-range measurements. The standard deviation for generating the Gaussian noise out of unity white $w_{u,k}$ noise is calculated with 3.14.

$$\begin{aligned} z_{acc,k} &= a_{ref,k} + b_a + w_{a,k} \qquad & w_{a,k} &= \sigma_a w_{u,k} = R_{S,noise}\sqrt{B} w_{u,k} \\ z_{laser,k} &= p_{ref,k} + b_a + w_{p,k} \qquad & w_{p,k} &= \sigma_p w_{u,k} \end{aligned} \tag{3.60}$$

The bandwidth from the digital low-pass filter is $B = 12.5Hz$. A bias for the accelerometer is chosen as $b_a = +0.5\ \frac{m}{s^2}$. The iteration frequency of the loop is $f_s = \frac{1}{\tau_s} = 10Hz$. This set-up shall demonstrate the ability of a Kalman filter to generate an optimal result out of significantly noisy position data. In addition the results of the backward smoother are presented in the same plots.

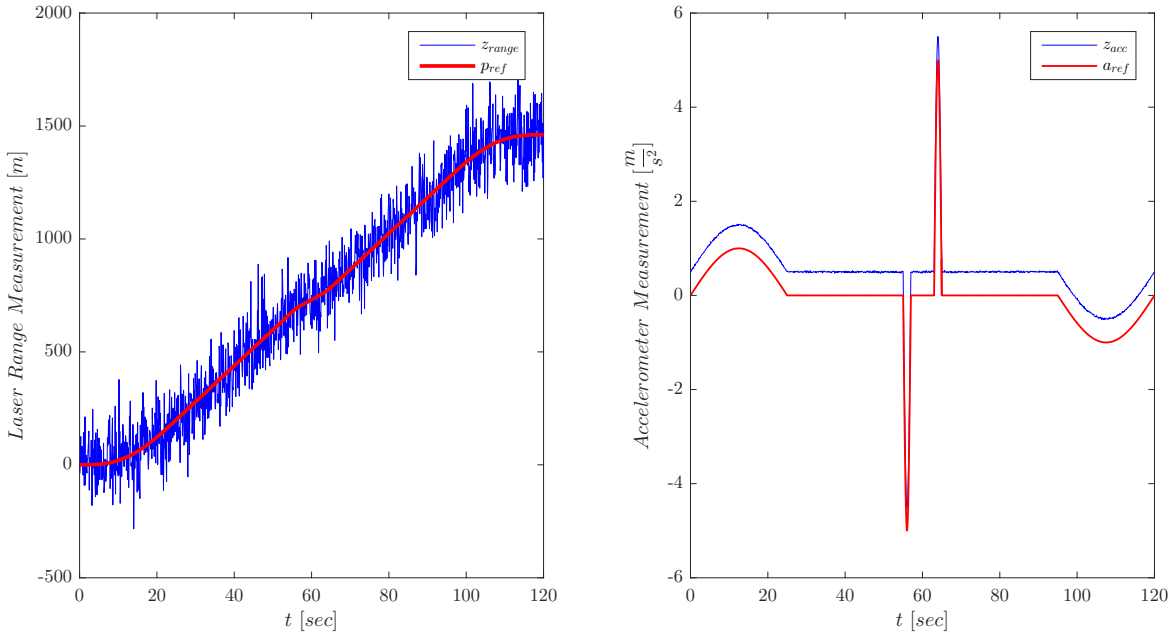


Figure 3.13: Results for laser and acceleration measurements

The Kalman filter and smoothing algorithm is started with an initial state covariance matrix

of

$$\boldsymbol{P}_{initial} = \begin{bmatrix} 10^2 & 0 & 0 \\ 0 & 0.1^2 & 0 \\ 0 & 0 & 0.5^2 \end{bmatrix} \tag{3.61}$$

Theoretically the Kalman filter is searching in the first steps for a rough estimate of the state with large uncertainties. After this transient phase the estimates starts to swing around their true values, but constantly minimizing their standard deviation, which has been the purpose of the Kalman gain derivation. The filter will converge in a steady-state after some time with the optimal achievable precision of the combined sensors if all parameters are tuned correctly [8, p.437]. In cases of bad tuning or erroneous unexpected measurements the filter possibly diverges and the state-estimate drifts away from their true value. Also some states may be unobservable which is tested by the observability matrix for linear time-invariant systems [8, p.62]. An additional advantage of the Kalman filter is to observe states which are not directly accessible for measurement equipment.

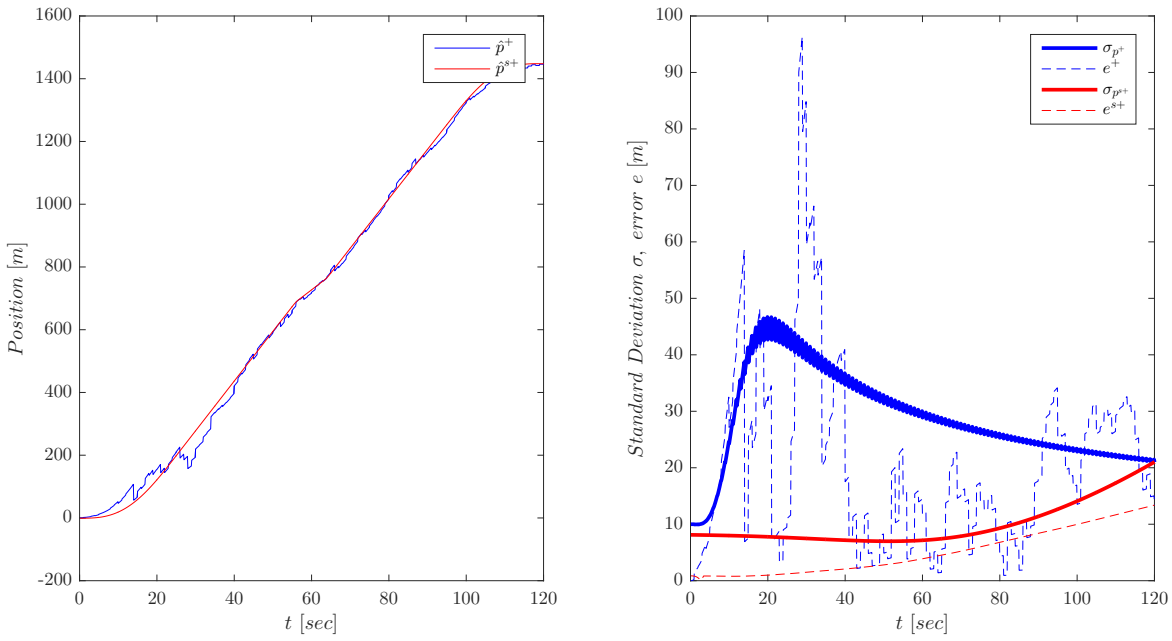


Figure 3.14: Position results for filter and smoother

For this example the results show the theoretical behaviour. The position estimates are clearly better than the measured position (Figure 3.14). The standard deviation reaches a value of 20 meter which is five times better compared to the laser-range with $\sigma = 100\ m$. Laser-range measurements are not able to give information about the short-time dynamics of the vehicle. This lack of ability had been criticised in the introduction for GPS too. The accelerometer's ability to give true information is also limited due to its bias. The Kalman filter does an remarkable job in solving this. The estimate of velocity (Figure 3.15) shows a

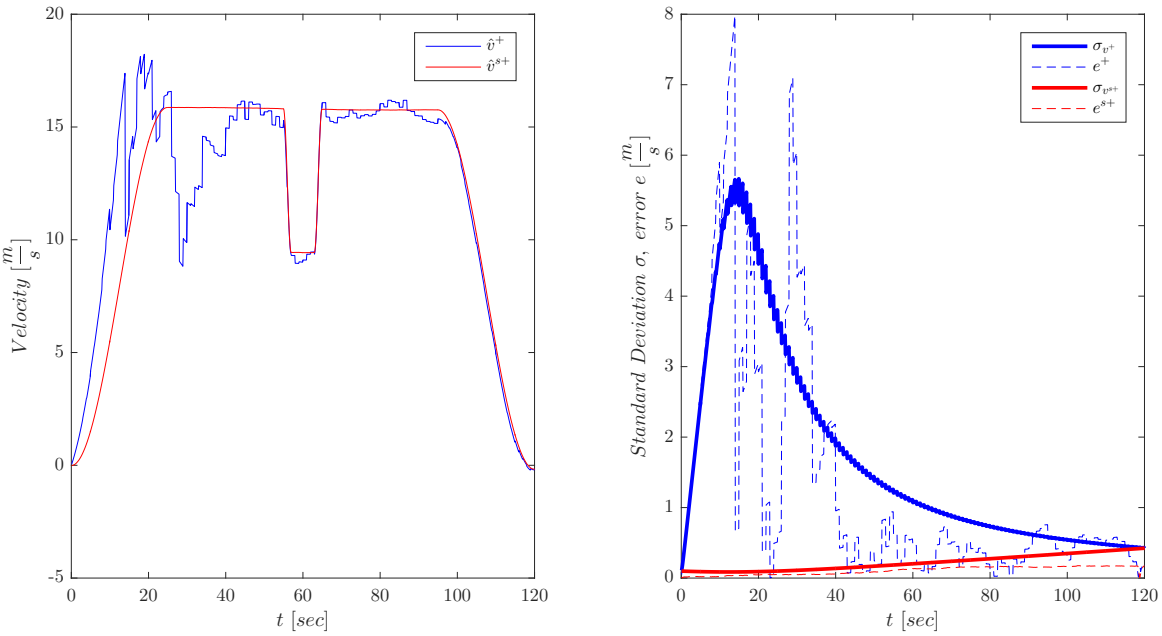


Figure 3.15: Velocity results for filter and smoother

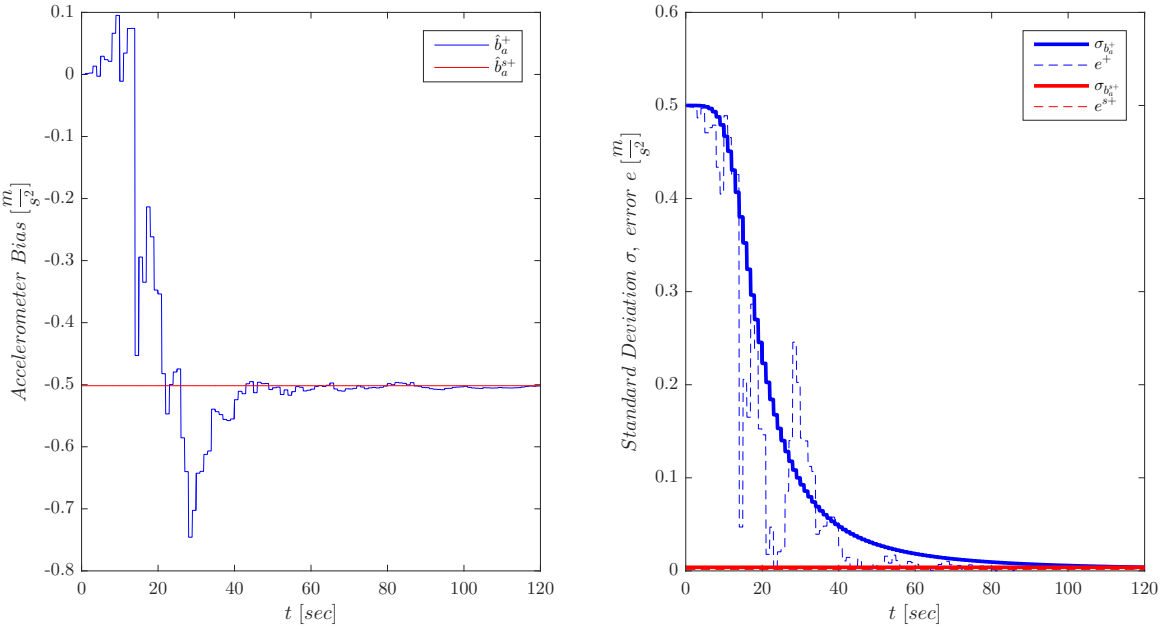


Figure 3.16: Bias results for filter and smoother

good accuracy after the transient phase nearly at the same time when the bias (Figure 3.16) is observed.

The standard deviation matches in most cases the true error marked by the dotted line. Thus, it is another credible and valuable information for the evaluation of data.

For the selected time interval the standard deviation of the bias reaches its steady state and the bias is nearly perfectly determined.

The RTS-smoother is filtering the filtered data backwards again, which should lead to an approximate performance improvement of factor 2. The standard deviations and true errors (red lines) confirm this. Besides the results in time-domain are getting smoother (this is where the name comes from) and correspond better with the real physical movement. Especially in situations of signal outages from the range equipment smoother can dramatically improve performance for those times.

3.3.5 Extended Kalman Filter

The Kalman filter algorithm is only working for linear systems. Since most physical connections are non-linear they have to be linearised [3]. The following derivation uses some ideas of [3, p.250-251] and connects them to the architecture and notation throughout this thesis.

Recalling the general time varying model from 3.22 and neglecting the input control $u(t)$, nearly all physical models fit to this expression.

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(t, \boldsymbol{x}(t)) + \boldsymbol{G}(t)\boldsymbol{w}(t) \\ \boldsymbol{z}(t) &= \boldsymbol{h}(t, \boldsymbol{x}(t)) + \boldsymbol{v}(t) \end{aligned} \tag{3.62}$$

Linearisation is achieved by truncating the Taylor expansion series after the first order.

$$f(x) = T(f(x), a) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \approx f(a) + \left[\frac{df(x)}{dx}\right]_{x=a}(x-a) \tag{3.63}$$

A linearisation point a must be chosen which is further called reference trajectory $\boldsymbol{x}^*(t)$. This reference trajectory may diverge from the true trajectory $\boldsymbol{x}(t)$. Thus,

$$\boldsymbol{x}(t) = \boldsymbol{x}^*(t) + \boldsymbol{\Delta x}(t); \tag{3.64}$$

Substituting the true trajectory $\boldsymbol{x}(t)$ in 3.62 gives

$$\begin{aligned} \dot{\boldsymbol{x}}^*(t) + \boldsymbol{\Delta}\dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(t, \boldsymbol{x}^*(t) + \boldsymbol{\Delta x}(t)) + \boldsymbol{G}(t)\boldsymbol{w}(t) \\ \boldsymbol{z}(t) &= \boldsymbol{h}(t, \boldsymbol{x}^*(t) + \boldsymbol{\Delta x}(t)) + \boldsymbol{v}(t) \end{aligned} \tag{3.65}$$

Applying the Tailor series 3.63 leads to

$$\dot{\boldsymbol{x}}^*(t) + \boldsymbol{\Delta}\dot{\boldsymbol{x}}(t) \approx \boldsymbol{f}(t, \boldsymbol{x}^*(t)) + \left[\frac{\boldsymbol{f}(t, \boldsymbol{x})}{\boldsymbol{dx}}\right]_{\boldsymbol{x}=\boldsymbol{x}^*(t)} \cdot \boldsymbol{\Delta x} + \boldsymbol{G}(t)\boldsymbol{w}(t) \tag{3.66}$$

$$\boldsymbol{z}(t) \approx \boldsymbol{h}(t, \boldsymbol{x}^*(t)) + \left[\frac{\boldsymbol{h}(t, \boldsymbol{x})}{\boldsymbol{dx}}\right]_{\boldsymbol{x}=\boldsymbol{x}^*(t)} \cdot \boldsymbol{\Delta x} + \boldsymbol{v}(t) \tag{3.67}$$

Splitting equation 3.66 results in two differential equations.

$$\dot{\boldsymbol{x}}^*(t) = \boldsymbol{f}(t, \boldsymbol{x}^*(t)) \tag{3.68}$$

$$\underbrace{\boldsymbol{\Delta}\dot{\boldsymbol{x}}(t)}_{\boldsymbol{d}\dot{\boldsymbol{x}}(t)} \approx \underbrace{\left[\frac{\boldsymbol{f}(t, \boldsymbol{x})}{\boldsymbol{\delta x}}\right]_{\boldsymbol{x}=\boldsymbol{x}^*(t)}}_{\boldsymbol{F}(t)} \cdot \underbrace{\boldsymbol{\Delta x}}_{\boldsymbol{\delta x}(t)} + \boldsymbol{G}(t)\boldsymbol{w}(t) \tag{3.69}$$

Compared to the error state Kalman filter these equations look similar which indicates that the differencing out idea and this version of the extended Kalman filter are related. Performing the linearisation in the error state differential equation 3.69 results in the same form as (3.55).

$$\boldsymbol{\delta}\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t) \cdot \boldsymbol{\delta x} + \boldsymbol{G}(t)\boldsymbol{w}(t) \tag{3.70}$$

The extended Kalman filter is the same as the linearized Kalman filter except that the linearisation takes place about the filter's estimated trajectory and not a precompiled reference trajectory [3, p.257]. Thus, the extended Kalman filter takes more of a risk as the linearisation points are influenced by the measurement updates.

The total state equation 3.68 represents the algorithm which is working in the box "total state algorithm" in figure 3.11. The reference system works with non-linear equations although the Kalman filter is linearised. The output of the reference system provides the values for the trajectory $\boldsymbol{x}^*(t)$.

From (3.67) the same residual measurement can be derived as in (3.54) and (3.56).

$$\boldsymbol{\delta z}(t) = \boldsymbol{z}(t) - \boldsymbol{h}(t, \boldsymbol{x}^*(t)) \approx \underbrace{\left[\frac{\boldsymbol{h}(t, \boldsymbol{x})}{\boldsymbol{dx}}\right]_{\boldsymbol{x}=\boldsymbol{x}^*(t)}}_{\boldsymbol{H}(t)} \cdot \boldsymbol{\Delta x} + \boldsymbol{v}(t) \tag{3.71}$$

Taken all together the discrete algorithm has to be extended by a linearised system matrix $\boldsymbol{\phi}_{k-1}$ and a measurement matrix $\boldsymbol{H}_k$.

$$\boldsymbol{F}_{k-1} = \left.\frac{\boldsymbol{df}(t,\boldsymbol{x})}{\boldsymbol{dx}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1}^{+}} \qquad \boldsymbol{\phi}_{k-1} \approx e^{\boldsymbol{F}_{k-1}\tau_s} \tag{3.72}$$

$$\boldsymbol{H}_k = \left.\frac{\boldsymbol{dh}(t,\boldsymbol{x})}{\boldsymbol{dx}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1}^{+}} \tag{3.73}$$

This is valid when the continuous non-linear equations are nearly constant for the propagation interval τ_s. A test is proposed in [9, p.120].

In 3.64 the true state has been defined as $\boldsymbol{x}(t) = \boldsymbol{x}^*(t) + \boldsymbol{\Delta x}(t)$. Therefore in the discrete Kalman filter the predicted total estimate must be corrected as follows:

$$\hat{\boldsymbol{x}}_k^{+} = \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{\delta}\hat{\boldsymbol{x}}_k^{+} \tag{3.74}$$

3.4 Summary

Different measurement equipment outputs noisy signals containing true information about multiple quantities which are connected by physical relationships. A state space model is a way to describe this mathematically while the states represent the true information which have to be determined (3.1).

The Kalman filter assumes Gaussian distributed noise in the measurements and yields an optimal solution for estimating those states. The errors of the estimates are described by covariances. With the transition matrix the states and the covariances are propagated to the time of validity of new measurement data. The Kalman gain leads to a sophisticated ratio of these new measurements and the predictions in the updated states (3.2).

Inertial sensors provide reliable measurements with Gaussian noise and deterministic error characteristics. This makes them suitable for modelling in an error state system model for predicting those errors in the integrated velocity and position solution. The Kalman filter weights the error of this reference system solution against position fixing residual measurements. As a consequence deterministic errors like the accelerometer bias becomes observable and the accuracies of all states grow with time. The total states are outsourced in the total state algorithm leading to a better efficiency (3.3.2).

The extended Kalman filter has been developed for working with non-linear differential equations. Its derivation results in an error state system model where only the system matrix for predicting the uncertainties and the measurement matrices must be linearised. The total states can be processed with non-linear equations resulting in possible higher accuracy (3.3.5).

4 Coordinate Frames

This chapter has been created with help of [9, p.24-78]. An orthogonal euclidean coordinate frame has an origin $\boldsymbol{o}$ and three right handed perpendicular axes x, y, z. Navigation comprises the description of motion, orientation and position of an object 'A' relative to some reference 'B'(Figure 4.1). Therefore a coordinate frame α must be defined with respect to a reference coordinate frame β. The position $\boldsymbol{o}^{\alpha}$ is defined by relative coordinates $\boldsymbol{r}^{\beta}_{\beta\alpha}$ in the reference frame and the axes by a rotation from the reference axes to the object axes in terms of euler rotation angles $\boldsymbol{\psi}$. The rotation is performed right handed where the thumb points in the direction of the rotation axis and the bent fingers in positive rotation direction. This is why an orthogonal frame has six degrees of freedom.

$$\boldsymbol{o}^{\alpha} = \boldsymbol{o}^{\beta} + \boldsymbol{r}^{\beta}_{\beta\alpha} = \boldsymbol{o}^{\beta} + \begin{bmatrix} \Delta x^{\beta} \\ \Delta y^{\beta} \\ \Delta z^{\beta} \end{bmatrix}_{\beta\alpha} \qquad \boldsymbol{\psi} = \begin{bmatrix} \phi_{\beta\alpha} \\ \theta_{\beta\alpha} \\ \psi_{\beta\alpha} \end{bmatrix} \tag{4.1}$$

where the angles $\boldsymbol{\psi}$ are rotated in following sequence:

1. $\psi_{\beta\alpha}$ is the yaw angle describing the rotation around the z^{β}-axis $(0 - 360°)$ into the intermediate $\beta\psi$-frame.

2. $\theta_{\beta\alpha}$ is the pitch angle describing the rotation around the $y^{\beta\psi}$-axis $(0 - 180°)$ into the intermediate $\beta\theta$-frame.

3. $\phi_{\beta\alpha}$ is the roll angle describing the rotation about the $x^{\beta\theta}$-axis $(0 - 360°)$ and results in the alignment of the frame α with respect to frame β.

The navigation quantity x may be expressed with the resolving axes in the coordinate frame γ in superscript and the specification about the relative frame connection $\beta\alpha$ in subscript resulting in $x^{\gamma}_{\beta\alpha}$. As an example the relative velocity v of an object in frame α with respect to the origin of frame β expressed in frame α is $v^{\alpha}_{\beta\alpha}$.

Rotations can be performed with a coordinate transformation matrix (CTM) $\boldsymbol{C}^{\beta}_{\alpha}$ changing the resolving axes of a 3 dimensional quantity $\boldsymbol{x}$ from frame α to frame β.

$$\boldsymbol{x}^{\beta} = \boldsymbol{C}^{\beta}_{\alpha}\boldsymbol{x}^{\alpha} \tag{4.2}$$

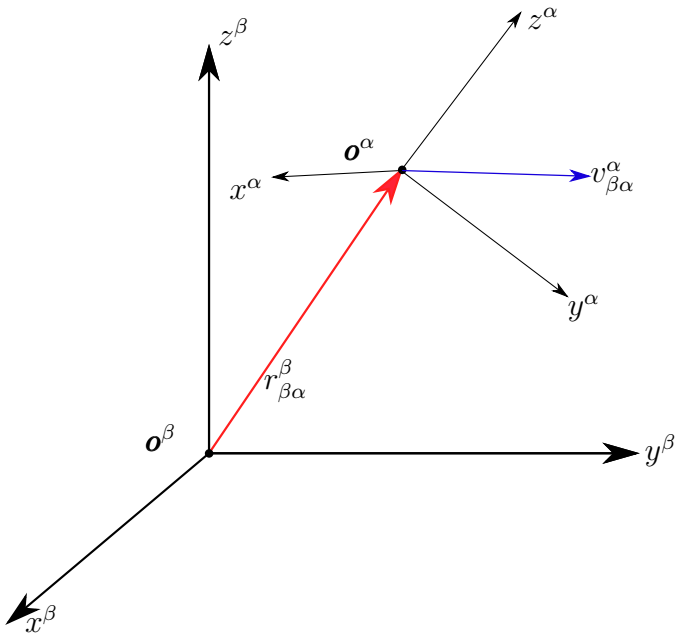


Figure 4.1: Coordinate frames in 3 dimensions

The coordinate transformation matrix is given by the euler angles.

$$\boldsymbol{C}_\alpha^\beta = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi_{\beta\alpha} & sin\phi_{\beta\alpha} \\ 0 & -sin\phi_{\beta\alpha} & cos\phi_{\beta\alpha} \end{bmatrix}}_{\text{Roll Rotation}} \underbrace{\begin{bmatrix} cos\theta_{\beta\alpha} & 0 & -sin\theta_{\beta\alpha} \\ 0 & 1 & 0 \\ sin\theta_{\beta\alpha} & 0 & cos\theta_{\beta\alpha} \end{bmatrix}}_{\text{Pitch Rotation}} \underbrace{\begin{bmatrix} cos\psi_{\beta\alpha} & sin\psi_{\beta\alpha} & 0 \\ -sin\psi_{\beta\alpha} & cos\psi_{\beta\alpha} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Yaw Rotation}} \tag{4.3}$$

When $\theta_{\beta\alpha} = \frac{\pi}{2}$ the rotation about $\psi_{\beta\alpha}$ or $\phi_{\beta\alpha}$ has the same effect, loosing one degree of freedom ("Gimbal Lock"). A solution is the attitude expression with quaternions. Indeed it is quite non-intuitive to work with them as they are four dimensional numbers. Since no street has a slope of 90° a land-vehicle will never be in a situation with an angle of $\theta_{\beta\alpha} = \frac{\pi}{2}$. Thus euler angles are used in this thesis. Some important properties of the CTM are

$$\left(\boldsymbol{C}_\alpha^\beta\right)^T = \boldsymbol{C}_\beta^\alpha \qquad \boldsymbol{C}_\alpha^\gamma = \boldsymbol{C}_\beta^\gamma \boldsymbol{C}_\alpha^\beta \qquad \boldsymbol{C}_\alpha^\beta \boldsymbol{C}_\beta^\alpha = \boldsymbol{I_3} \tag{4.4}$$

Often the small angle approximation is useful where $[\boldsymbol{\psi}_{\beta\alpha}\times]$ is the skew symmetric form of the euler angles in radians.

$$\boldsymbol{C}_\alpha^\beta \approx \begin{bmatrix} 1 & \psi_{\beta\alpha} & -\theta_{\beta\alpha} \\ -\psi_{\beta\alpha} & 1 & \phi_{\beta\alpha} \\ \theta_{\beta\alpha} & -\phi_{\beta\alpha} & 1 \end{bmatrix} = \boldsymbol{I_3} - [\boldsymbol{\psi}_{\beta\alpha}\times] \qquad \boldsymbol{\psi}_{\alpha\beta} \approx -\boldsymbol{\psi}_{\beta\alpha} \tag{4.5}$$

A coordinate frame 'A' is defined relative to another reference frame 'B'. Frame 'B' must

be defined too. The last frame in this definition sequence one could think of is an inertial frame.

4.1 Earth Centered Inertial Frame

A coordinate frame which is not accelerating or rotating with respect to the rest of the universe is called inertial frame. When neglecting the rotation of the Earth around the Sun and the galaxy an Earth centered intertial frame (ECI) with its origin in the center of mass is nearly inertial (Figure 4.2).

1. The z^i-axis is along the Earth's axis of rotation to the true north.
2. The x- and y-axis lie within the equatorial plane, but do not rotate with the Earth.
3. The x^i-axis points towards the vernal equinox. This is the direction to the Sun exactly when the equatorial plane passes the center of the Earth orbital plane (ecliptic) around the Sun which is the spring equinox of the northern hemisphere.
4. The y^i-axis completes an right handed orthogonal frame.

Inertial sensor uses the ECI-frame as the reference frame. It is denoted by the symbol i.

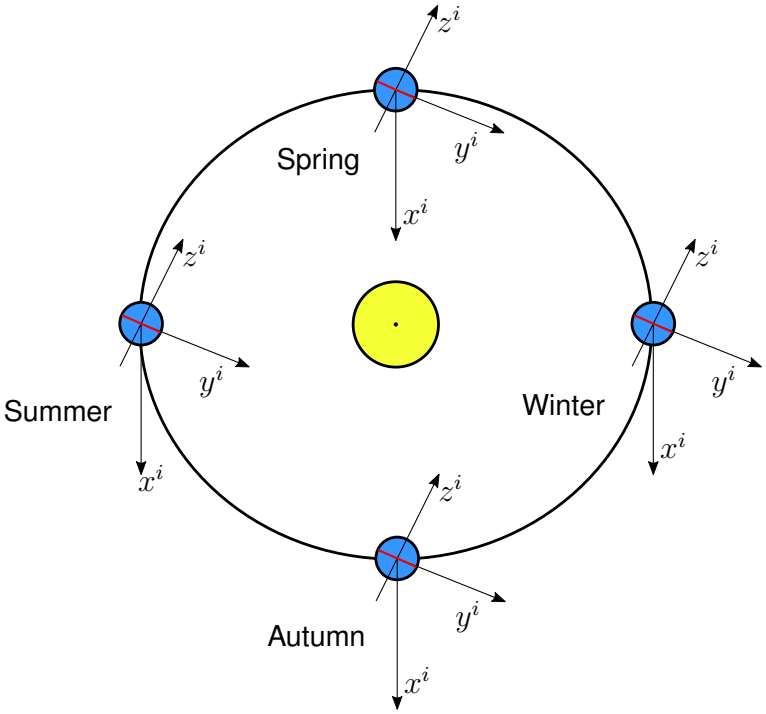


Figure 4.2: Inertial frame definitions with Earth and Sun

4.2 Earth Centered Fixed Frame

An Earth centered fixed frame (ECEF) is quite similar to the ECI-frame, except that it is rotating with the Earth. Therefore

1. The z^e-axis is along the Earth's axis of rotation to the true north.
2. The x and y-axis lie within the equatorial plane rotating with the Earth.
3. The x^e-axis points towards the intersection of the equatorial plane and the zero-meridian at longitude 0°.
4. The y^e-axis completes a right handed orthogonal frame pointing towards the intersection of 90° east meridian with the equator.

The ECEF-frame is important for navigation as most users wants to know their position relative to the Earth. It is denoted by the symbol e.

4.2.1 Cartesian Position

The cartesian position r^e_{eb} (b: body) in an ECEF-frame is defined by values for the components x^e,y^e,z^e. The resolving axis are independent. Thus, adding, substracting and transforming positions is easily accomplished. Unfortunately the numbers are user-unfriendly, e.g. the coordinate for Hamburg is $x^e = 3739.905\ km$, $y^e = 659.021\ km$ and $x^e = 5107.222\ km$ and it is hard to imagine a position with such big numbers. This is why the ECEF-Position is commonly used for navigation in space, but not for the surface of Earth.

4.2.2 Curvilinear Position

The shape of the Earth has nearly the form of an ellipsoid. This is why a standard ellipsoid model has been introduced with the World Geodetic System in 1984 (WGS84). WGS84 was developed for the U.S. military and later used as the reference model in most GPS-applications. An elliposid can be characterized by four parameters which are the equatorial radius R_0, the polar radius R_P, the flattening factor f and the eccentricity e. For the WGS84 they are

$$\begin{aligned} R_0 &= 6378.137km \qquad & R_P &= 6356.75231425km \\ f &= 1/298.257223563 \qquad & e &= 0.0818191908425 \end{aligned} \tag{4.6}$$

Using this model as the reference a position on Earth is described by three coordinates $\boldsymbol{p}_b = [L_b, \lambda_b, h_b]$ (b: body) aligned with the axes of the navigation frame.

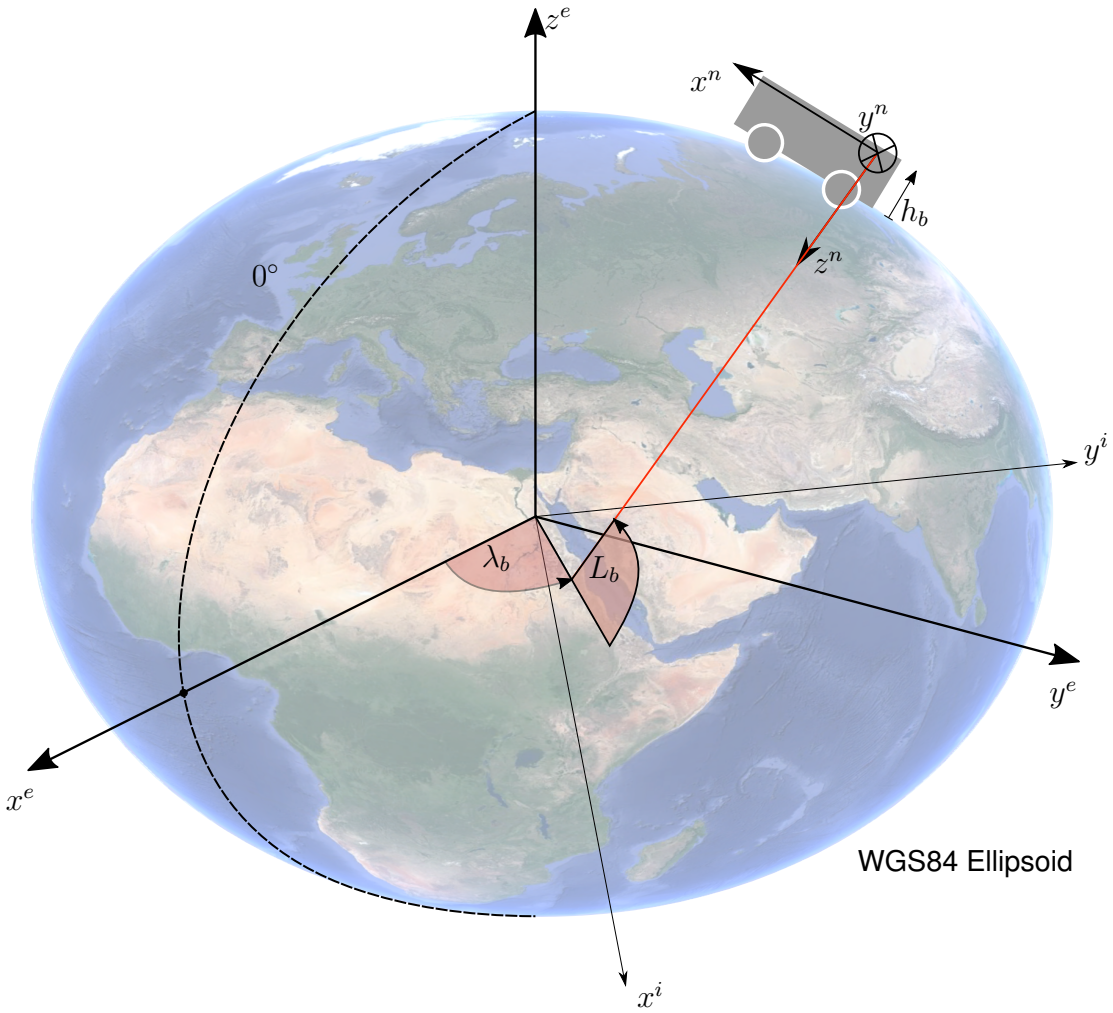


Figure 4.3: Inertial frame, Earth frame and navigation frame

1. L_b is the geodetic latitude defined by the angle of intersection of the normal from a body to the ellipsoid with the equatorial plane. Every point on this normal line has the same geodetic latitude.

2. λ_b is the longitude. When projecting the ellipsoid on the equatorial plane all points have the same longitude lying on a radius line defined by the longitude angle. This angle starts at $0°$ defined by the zero meridian at Greenwich (London) and is positive in the eastern hemisphere and negative in the western hemisphere.

3. h_b is the geodetic height above the WGS84 ellipsoid and the mean sea level. Note that sea level varies on the Earth due to gravitational anomalies. Therefore some GPS-Receiver use local sea level of a simple geoid model as zero height for h_b.

The curvilinear position in decimal degrees and an accuracy of one meter for Hamburg is $\boldsymbol{p}_b = [53.55108°,\ 9.99368°, 6\ m]$ which is more user-friendly than the Cartesian-position. This is why the curvilinear position is chosen throughout the next sections. Calculations are always done with angles in radians.

4.2.3 Navigation Frame

The origin of the navigation frame is the location defined by the curvilinear position p^b of the inertial measurement unit in the vehicle.

1. The z^n-axis is aligned with the vector of gravity, which is a combination of gravitational force and centrifugal force. As a result the enhanced vector does not pass the center of Earth.
2. The x^n-axis is the line orthogonal from the z-axis to the north pole.
3. The y^n-axis completes a right handed orthogonal frame pointing to the east.

With its directions it is often abbreviated as North-East-Down-frame (NED). For local navigation the information about angles with respect to the navigation frame axes are especially important. This is why angles in further Kalman filter modelling uses the NED-frame as the reference-frame. It is denoted by the symbol n.

4.3 Sensor Frame and Body Frame

Both frames have the same origin as the navigation frame and are visualised in figure 4.4. The sensor frame axes, denoted by superscript s, are defined by the measurement axes of the inertial measurement unit (IMU) (Figure 5.3). The axes are aligned with the body frame but the x,y,z-labels are interchanged. The body frame labels are commonly chosen as follows:

1. The x^b-axis points forward (roll) and equals the negative z^s-axis.
2. The y^b-axis points to the right (pitch) and equals the negative x^s-axis.
3. The z^b-axis points down (yaw) and equals the y^s-axis.

The body frame is fundamental for the navigation as microelectromechanical system (MEMS)-sensors have a fixed alignment with its axes. It is denoted by the symbol b.

4.4 Rear Frame

The wheelspeed is measured at the rear-axis of the vehicle.

1. The y^r-axis is along the vehicle axis to the right.
2. The x^r axis points forward.
3. The z^r axis completes a right handed orthogonal frame pointing downwards.

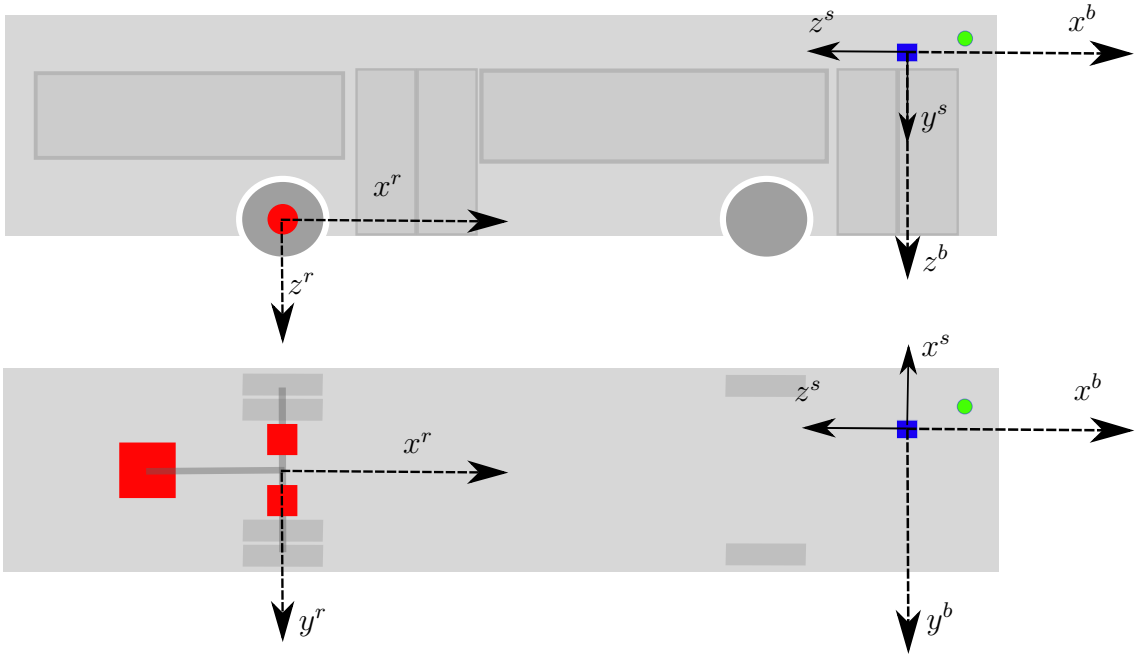


Figure 4.4: Sensor frame, body frame and rear frame (Top: Side view, Bottom: Plan view)

5 Experimental Setup

The datalogger has been designed [33] and build by project BEEdeL and is already operating in the city of Hamburg. This chapter will present the specific information required for developing the Kalman filter algorithm with its architecture and parameters.

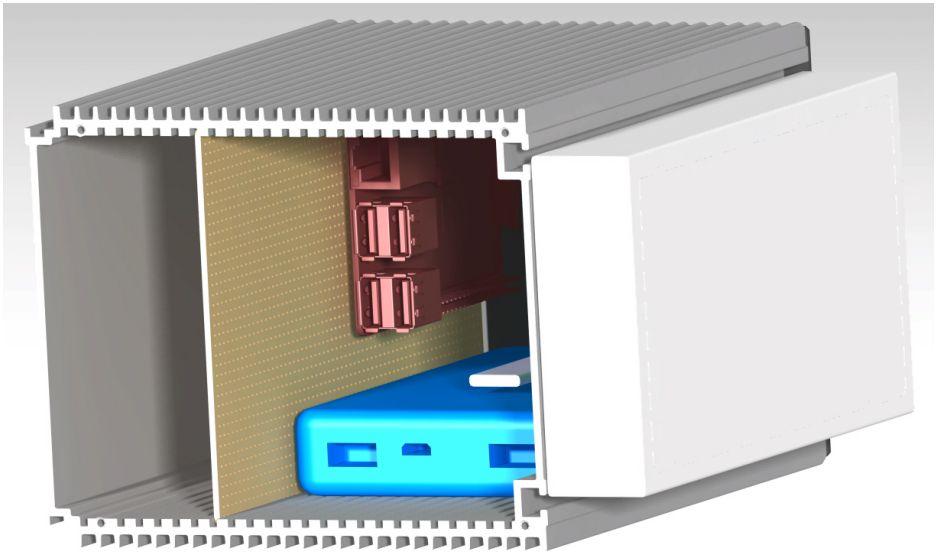


Figure 5.1: Data logger box [13]

5.1 Vehicle Description

In Hamburg three different types of buses serve for public transportation. Their employment depends primarily on the amount of people to transport. In this thesis data is only available from the normal bus and the articulated bus. In urban areas the larger vehicles are commonly chosen especially in times of high requests.

In 5.2 the red box indicates the place of the engine and differential. The blue box shows the location where the data logger is installed. The green dot is the place of of the GPS-antenna where the satellite-signals are received. The difference in position between the logger and the antenna is accounted for with a lever arm of $l_{ab}^{b} \approx [-0.5, 0, 0]m$. The bus size data is taken from [17, p.5].

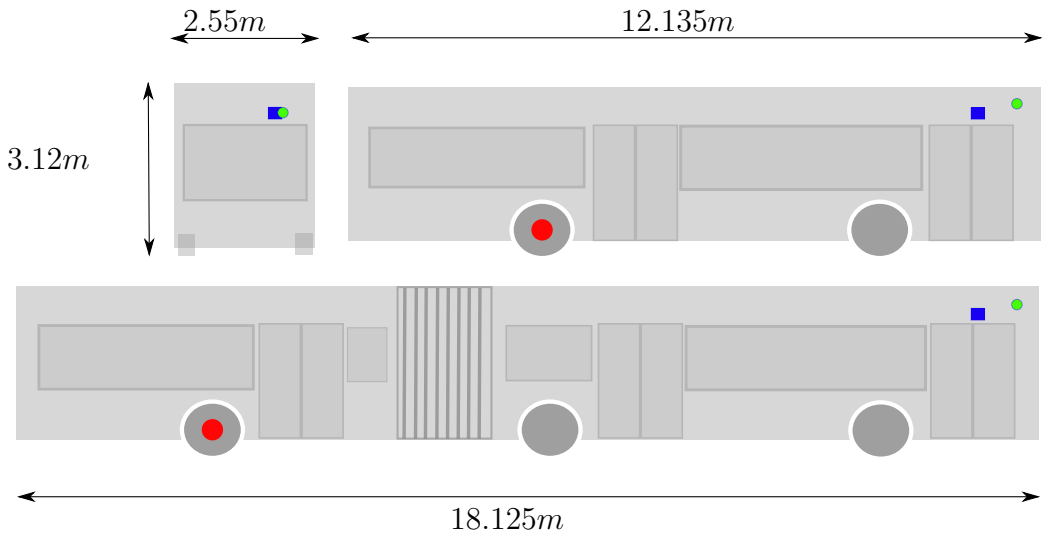


Figure 5.2: Normal bus and articulated bus

5.2 GPS-Receiver

After performance tests of three different modules on a urban route through hamburg the SparkFun Venus GPS Module has been chosen [33, p.45-47]. The module delivers its data with an UART interface sending two different National Marine Electronics Association (NMEA)-messages ‘GGA’ and ‘RMC’ summarized in table 5.1.

The module evaluates multiple satellite range measurements internally(similar to the one dimensional example) and generates a position information with at least 4 valid signals. An internal Kalman filter is always enabled in selectable pedestrian or car mode smoothing the position stream [28].

Table 5.1: NMEA information[29]

Information	Description
Position	Longitude, latitude, height
GPS-Quality indicator	0 position fix unavailable 1 valid position fix, SPS mode 2 valid position fix, differential GPS mode 6 Estimated (dead reckoning) mode
Satellites used	Number of satellites in use (0 - 12)
HDOP	Indicator for quality of satellite geometry for horizontal positioning (0.8 (best) - 99.9)
Time and date	Universal Time Coordinated
Speed over ground	Absolute speed in knots
Course over ground	Clockwise angle between vector of north and vector of travel direction (0 - 359.9°)
Mode	N Data not valid A Autonomous mode D Differential mode E Estimated (dead reckoning) mode
Hemisphere	Latitude hemisphere (north, south) Longitude hemisphere (east, west)

5.3 Inertial Measurement Unit

The module Adafruit 10-DOF IMU Breakout (5.3) is used as the inertial measurement unit (IMU) in the datalogger providing sensors for acceleration, angular rate, magnetic field, air pressure and temperature measurements [33, p.52]. The following sections describe the characteristics of each sensor.

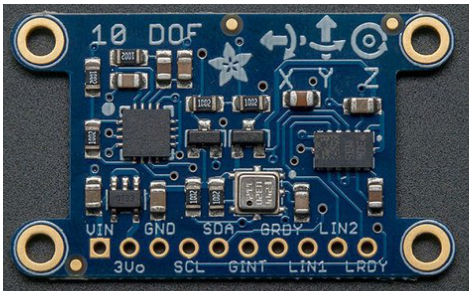


Figure 5.3: Adafruit 10-DOF IMU Breakout Board [30]

5.3.1 Accelerometer

The 3D-MEMS capacitive accelerometer 'LSM303DLHC' is a system-in-package including a magnetometer which is analysed in section 5.3.3. It is manufactured by using specialised micro-machining processes and capable of measuring linear acceleration along 3 perpendicular axes in units of [g] [26, p.1,15], where 1 g equals 9.80665 $\frac{m}{s^2}$. The device belongs to the pendulous type of MEMS(Micro-Electronic-Mechanical-System)-accelerometers. Alternative technologies like vibrating beam or resonant sensors provide a higher accuracy [5, p.206]. The configuration which has been done in [32] leads to a measurement range of $\pm$4g, a sensitivity of 2 mg per least significant bit (LSB) and the output data rate (ODR) is 25 Hz.

Table 5.2: Accelerometer sensor characteristics

Characteristic	**Data sheet**	**Laboratory**	
		Setting Data Sheet	Setting Data Logger
Sensitivity	$\pm 2\ mg$	$\pm 2\ mg$	$\pm 2\ mg$
Range	$\pm 2\ g$	$\pm 2\ g$	$\pm 4\ g$
ODR	$1344\ Hz$	$1344\ Hz$	$25\ Hz$
Bandwidth B	$\frac{ODR}{9} = 150\ Hz$	$\frac{ODR}{9} = 150\ Hz$	$\frac{ODR}{9} \approx 3\ Hz$
Noise R_S	$220\ \frac{\mu g}{\sqrt{Hz}}$	$325\ \frac{\mu g}{\sqrt{Hz}}$	$2484\ \frac{\mu g}{\sqrt{Hz}}$
Bias σ_b	$\pm 60\ mg$		$< 60\ mg^{(1)}$
Temperature bias	$\pm 0.5\ mg/°C$		$x:\ \pm 0.142\ mg/°C^{(1)}$
			$y:\ \pm 0.184\ mg/°C^{(1)}$
			$z:\ \pm 1.914\ mg/°C^{(1)}$

(1) ODR: 10 Hz

Due to technical limitations all accelerometer measurements are influenced by different error sources. In MEMS-systems the bias $\boldsymbol{b}_a$ consists of a static initial offset and a dynamic temperature dependent bias. Another source of error (up to 10% of the absolute measured acceleration [9, p.155]) is caused by scale errors and misalignments of the axes described with the scale- and cross coupling matrix $\boldsymbol{M}_a$. Electrical noise of the sensor matches approximately white noise $\boldsymbol{w}_a$. From the datasheet the noise density is 220 $\frac{\mu g}{\sqrt{Hz}}$ measured at a configured range of $\pm$2 mg and data rate of 1344 Hz noting that the noise density should be independent of the data rate. The following equation shows how different sensor error sources contribute mathematically to the measurements [9, p.160].

$$\tilde{\boldsymbol{f}}_{ib}^{b} = \boldsymbol{b}_a + (\boldsymbol{I}_3 + \boldsymbol{M}_a)\boldsymbol{f}_{ib}^{b} + \boldsymbol{w}_a \tag{5.1}$$

Laboratory tests

Lab experiments [14] can confirm the data sheet noise values for the same configuration in table 5.2. The noise density is calculated with $R_S = \frac{\sigma_w}{\sqrt{B}}$ (3.14). When configuring the setting used in the data loggers the noise density is 10 times higher than expected. This is probably due to the greater range of $\pm 4mg$ or the internal filter does not limit the bandwidth to 3 Hz. The analysis is part of further research because in this thesis only already recorded data is processed.

The MEMS accelerometer is very sensitive for vibration induced errors. The vibration of the temperature cabinet used for temperature experiments increases the noise by a factor of 7.

5.3.2 Gyroscope

The three axes angular rate sensor 'L3GD20H' is also manufactured with micro machining processes. It measures the Coriolis force applied on moving mirco-meachanical elements when rotating the sensor frame with respect to an inertial frame. The amount of force is linearly related to the angular rate which the sensor outputs in units of dps (°/s). In the data logger the gyrsocope has been configured [32] with a range of ± 245 dps, a sensitivity of 8.75 mdps an output data rate of 94,7 Hz which is cut off by a low pass filter at 12.5 Hz.

Table 5.3: Gyroscope sensor characteristics

Characteristic	Data sheet	Laboratory	
		Setting Data Sheet	Setting Data Logger
Sensitivity	$\pm 8.75\ mdps$	$\pm 8.75\ mdps$	$\pm 8.75\ mdps$
Range	$\pm 245\ dps$	$\pm 245\ dps$	$\pm 245\ dps$
ODR	$400\ Hz$	$400\ Hz$	$94.7\ Hz$
Bandwidth B	$50\ Hz$	$50\ Hz$	$12.5\ Hz$
Noise R_S	$0.011\ \frac{dps}{\sqrt{Hz}}$	$0.029\ \frac{dps}{\sqrt{Hz}}$	$0.03\ \frac{dps}{\sqrt{Hz}}$
Bias σ_b	$\pm 25\ dps^{(1)}$		$< 25\ dps^{(2)}$
Temperature bias	$\pm 0.04\ dps/°C^{(1)}$		$\pm 0.027\ dps/°C^{(2)}$
	[(1)] Range: 2000 dps		[(2)] ODR: 12.5 Hz

The gyroscope exhibits also similar errors like the accelerometer. From the data sheet [27] the static bias $\boldsymbol{b}_{gs}$ can be up to ±25 dps at a range of 2000 dps. The dynamic bias $\boldsymbol{b}_{gd}$ is ±0.04 dps/°C. For the noise $\boldsymbol{w}_g$ a density of 0.011 dps/$\sqrt{Hz}$ is given. Scale factor errors and misalignments can occur in the same way as in MEMS accelerometers described by $\boldsymbol{M}_g$. Additionally a force-dependent error is introduced with $\boldsymbol{G}_g$ [9, p.160].

$$\tilde{\boldsymbol{\omega}}_{ib}^{b} = \boldsymbol{b}_g + (\boldsymbol{I}_3 + \boldsymbol{M_g})\boldsymbol{\omega}_{ib}^{b} + \boldsymbol{G}_g \boldsymbol{f}_{ib}^{b} + \boldsymbol{w}_g \tag{5.2}$$

Laboratory tests

In lab experiments [14] twice the noise density has been measured compared to the data sheet noise values in table 5.3. The higher amount of noise is caused by outliers which are handled in section 5.3.6. In contrast to the accelerometer the noise density results are independent of the data rate which confirms that the internal filters are working properly. Additionally the MEMS gyroscope is more robust in vibration environment. The vibration of the temperature cabinet induces only additional noise of factor 1.5.

5.3.3 Magnetometer

Moving electrical conducting fluids in the outer core of the Earth generates a magnetic field (Dynamo theory). This field is vertical at the magnetic poles and is inclined at nearly 10 $^\circ$ to the rotational axis of the Earth. The SI unit of the magnetic flux density B is Tesla $[T]$. The Earth's field density varies from 30 μT at the equator to 60 μT at the poles [9, p.218]. Due to this small amount magnetometers often measure the density in units of Gauss $[G]$ while $1\ G$ equals $0.1\ mT$.

The outputs of a magnetometer are influenced by the factors in equation (5.3) [9, p.221].

$$\tilde{\boldsymbol{m}}^b_{mb} = \boldsymbol{b}_m + (\boldsymbol{I}_3 + \boldsymbol{M}_m)\, \boldsymbol{C}^b_n\, (\boldsymbol{m}^n_E + \boldsymbol{m}^n_A) + \boldsymbol{w}_m \tag{5.3}$$

where

$\boldsymbol{b}_m$ is the hard iron flux density from the magnetic structure of the vehicle including the sensor offset.

$\boldsymbol{M}_m$ is the scaling factor matrix for soft-iron effects produced by nearby materials that distort the magnetic field. It is including the cross coupling sensor errors.

$\boldsymbol{C}^b_n$ is the CTM from navigation frame to body frame (4.3).

$\boldsymbol{m}^n_E$ is the geomagnetic flux density.

$\boldsymbol{m}^n_A$ is the flux density from local magnetic anomalies.

$\boldsymbol{w}_m$ is the random sensor noise.

With help of the measurements and knowledge about roll ϕ_{nb} and pitch θ_{nb} the yaw angle ψ_{mb} can be calculated with (5.4).

$$\tilde{\boldsymbol{\psi}}_{mb} = \arctan_2 \begin{bmatrix} -\tilde{m}^b_{m,y} \cos\hat{\phi}_{nb} + \tilde{m}^b_{m,z} \sin\hat{\phi}_{nb}, \\ \tilde{m}^b_{m,x} \cos\hat{\theta}_{nb} + \tilde{m}^b_{m,y} \sin\hat{\phi}_{nb} \sin\hat{\theta}_{nb} + \tilde{m}^b_{m,z} \cos\hat{\phi}_{nb} \sin\hat{\theta}_{nb} \end{bmatrix} \tag{5.4}$$

The vector of magnetic flux density is not exactly aligned with the north vector due to the inclination phenomena. Equation 5.5 corrects the angle for the navigation frame. The size of the declination angle α_{nE} depends on date and location (Table 5.4).

Table 5.4: Declination angle α_{nE} calculated with the world magnetic model [6]

Location	**Time-dependent declination angle $\boldsymbol{\alpha}_{nE}$**		
	28.07.2015	28.07.2016	28.07.2017
London	-0.8 $\pm$ 0.37 $^\circ$	-0.65 $\pm$ 0.37 $^\circ$	-0.5 $\pm$ 0.37 $^\circ$
Hamburg	2.46 $\pm$ 0.38 $^\circ$	2.6 $\pm$ 0.38 $^\circ$	2.74 $\pm$ 0.38 $^\circ$
Reykjavik, Island	-14.39 $\pm$ 0.48 $^\circ$	-14.12 $\pm$ 0.48 $^\circ$	-13.86 $\pm$ 0.48 $^\circ$

$$\tilde{\boldsymbol{\psi}}_{nb} = \tilde{\boldsymbol{\psi}}_{mb} + \alpha_{nE} \tag{5.5}$$

5.3.4 Barometer

The barometric chip on the board is the 'BMP180' from Bosch. The characteristics are listed in table 5.5. In [25] an algorithm for correcting the height measurements using the barometer

Table 5.5: Barometer BMP180 sensor characteristics

Characteristic	**Data sheet**
Resolution output	$0.01\ hPa$
Absolute accuracy	$2\ hPa\ (0-65°C)$
Relative accuracy	$\pm\ 0.12\ hPa \hat{=}\ \pm 1\ m\ (25-40°C)$
RMS noise in ultra high resolution setting (conversion time 25.5 ms)	$0.03hPa \hat{=} 0.25m$

has been developed with the purpose of generating a height profile with respect to the height of a startpoint. This height has been set to zero or obtained from a height database [25, p.51,61]. In this thesis the purpose is to find the absolute offset of the pressure sensor without external help by using information from all sensors in the Kalman filter algorithm. The quality of GPS-height is be crucial for absolute information. In a multipath environment an innovation filter (Section 6.3.7) is able to detect outliers of the GPS measurements.

For barometric pressure measurements in units of pascal [Pa] there is a relationship to

the absolute height defined with the international barometric height formula [4].

$$h_b = \frac{T_s}{k_T}\left[1 - \left(\frac{p_b}{p_s}\right)^{\frac{Rk_T}{g_0}}\right] + h_s = \frac{T_s}{6.5 \cdot 10^{-3} K m^{-1}}\left[1 - \left(\frac{p_b}{p_s}\right)^{\frac{1}{5.261}}\right] + h_s \tag{5.6}$$

where

h_b is the Geodetic height.

h_s is the Geoid height.

p_b is the measured air pressure.

T_s is the temperature on the surface of the ellipsoid.

$k_T = 6.5 \cdot 10^{-3} K m^{-1}$ is the atmospheric temperature gradient.

$p_s = 101.325 kPa$ is the surface pressure for mean sea level.

$R = 287.1 J k g^{-1} K^{-1}$ is the gas constant.

$g_0 = 9.80665 m s^{-2}$ is the average surface acceleration due to gravity.

Figures 5.4 visualizes the variation of pressure with respect to the height. Especially for heights up to 1000m the function is approximately linear. Several models of Earth describe

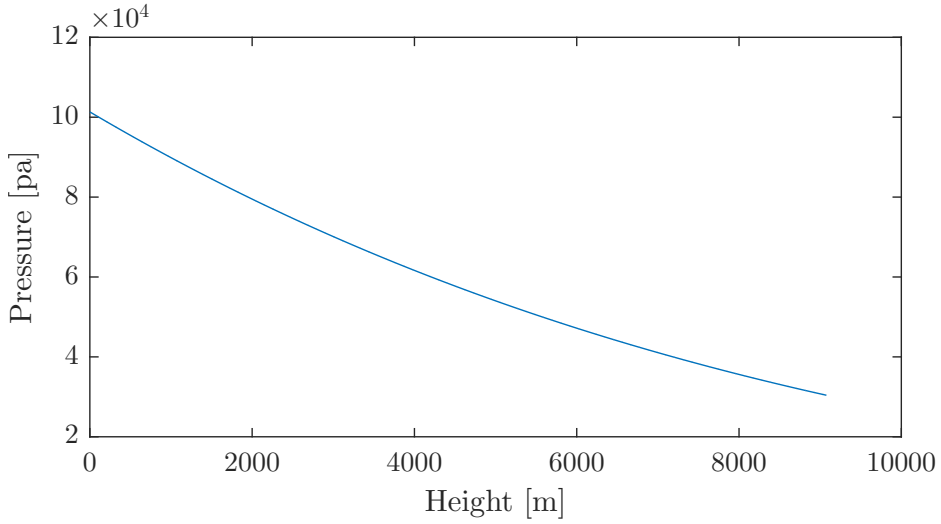


Figure 5.4: Variation of pressure with height

its shape. The model which includes all structures is called terrain. Because structures are changing constantly this a volatile definition. Therefore the geoid model has been introduced.

Substituting all structures with water but keeping the amount of gravity the water will distribute to places with more gravity. This results in the shape of the geoid. The WGS84 geoid model consists of 4.730.400 coefficients defining the geoid height. As a further simplification the WGS84 ellipsoid model has been developed which is defined by only four parameters (4.6). The geoid and elliposid height can differ up to 100m [9, p.64]. Figure 5.5 illustrates these different kinds of height.

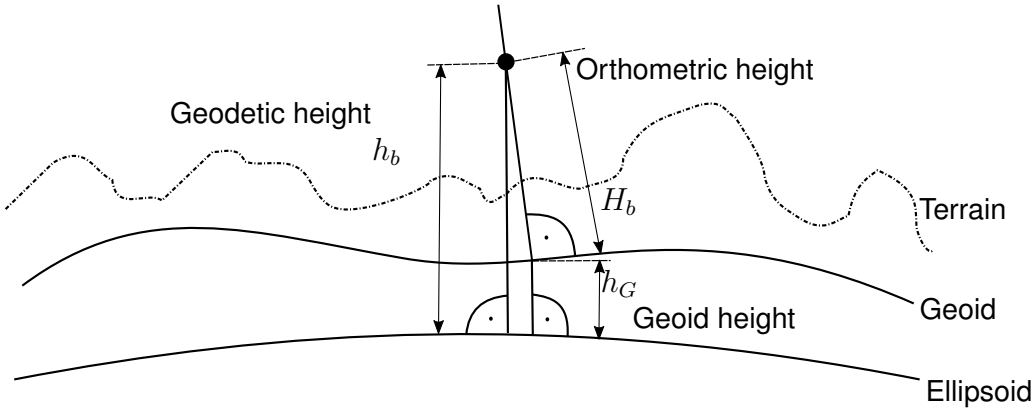


Figure 5.5: Height definitions, redrawn from [9, p.65]

5.3.5 Temperature

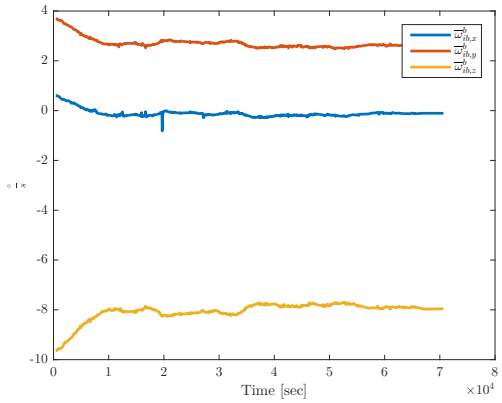
The barometer includes a temperature sensor with following characteristics:

- Resolution output 0.1 °C
- Absolute accuracy 2 °C (0-65°C)

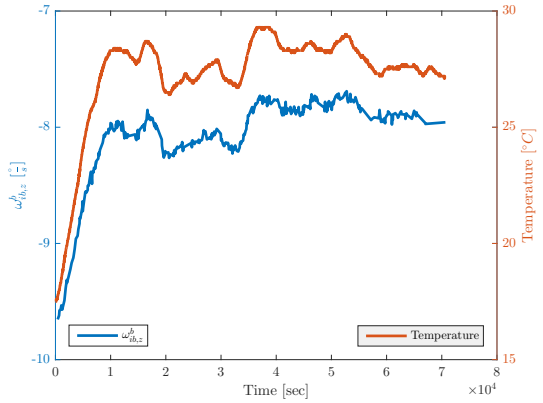
The bias of the gyroscope and accelerometer is drifting mainly due to changes in temperature. To get an idea about the impact, time intervals has been selected where the vehicle is stationary and the door closed to reduce errors from moving passengers or bus tilts. For each time interval the average values of the inertial sensors are calculated and stored in an array. The results are illustrated in figure 5.6a and 5.7a. The figures on the right show a direct comparison for the curves from temperature and one selected sensor axis.

In 5.6b the gyroscope z-axis is a clearly correlated with the temperature. This gives some indication of possibility to compensate the drift by inversely scaling the gyroscope measurements with the temperature curves.

The averaged accelerometer measurements have a obviously noisier character as the vehicle is aligned slightly differently at each stop due to the variation of the street's slope and tilts of the bus. In this form the raw data is too noisy to show a correlation with the temperature. Therefore the unequally time-spaced averages are resampled with Matlab to create



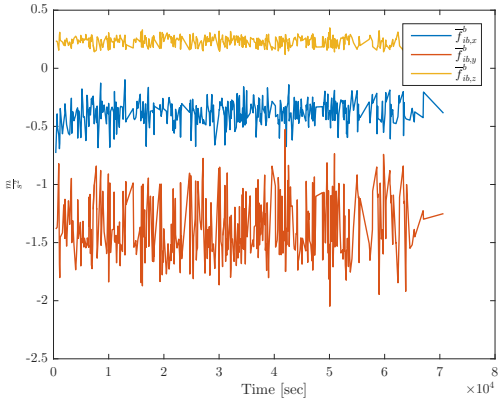
(a) all axes in body frame



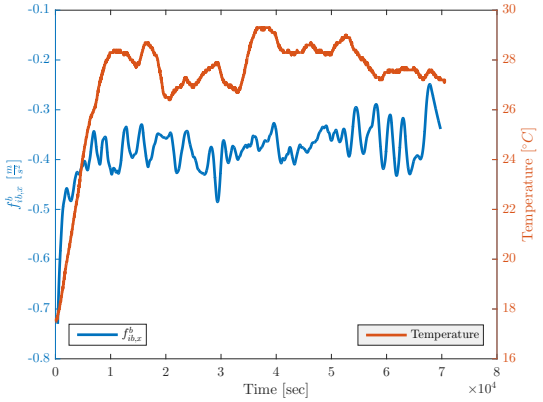
(b) z-axis (body frame) vs temperature

Figure 5.6: Averaged gyroscope data when stationary (Body frame)

equally spaced data. Now these data can be smoothed out with an average filter which leads to the curves in figure 5.7b. The accelerometer x-axis shows a weaker correlation to temperature due to the mentioned disturbances but has also the potential for compensation.



(a) all axes, z corrected for gravity



(b) Resampled and filtered x-axis vs temperature

Figure 5.7: Averaged accelerometer data when stationary (Body frame)

Based on the observations a linear relationship of the sensor bias drifts b and the temperature T can be assumed. Thus,

$$b(T) = c_1 + c_2 T \tag{5.7}$$

where

c_1 is the offset between bias and temperature.

c_2 is the scaling factor from the temperature curve to the bias.

A way to fit coefficients of a linear equation to two datasets is a least square fit described in [21, p.13-15]. It minimizes the distance between the calculated $b(T)$ and the measured biases $\tilde{b}$ which is called residual. For all measurements the linear connection is described by the overdetermined system in (5.8).

$$\begin{bmatrix} T_1 & 1 \\ T_2 & 1 \\ \vdots & \vdots \\ T_n & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \approx \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_n \end{bmatrix} \tag{5.8}$$
$$\boldsymbol{A} \cdot \boldsymbol{c} \approx \boldsymbol{y}$$

The residual is $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{c}$. To get rid of different signed residuals the squared residual has to be minimised. This leads to an optimal solution (5.9) for the coefficients in least square sense [21, p.14].

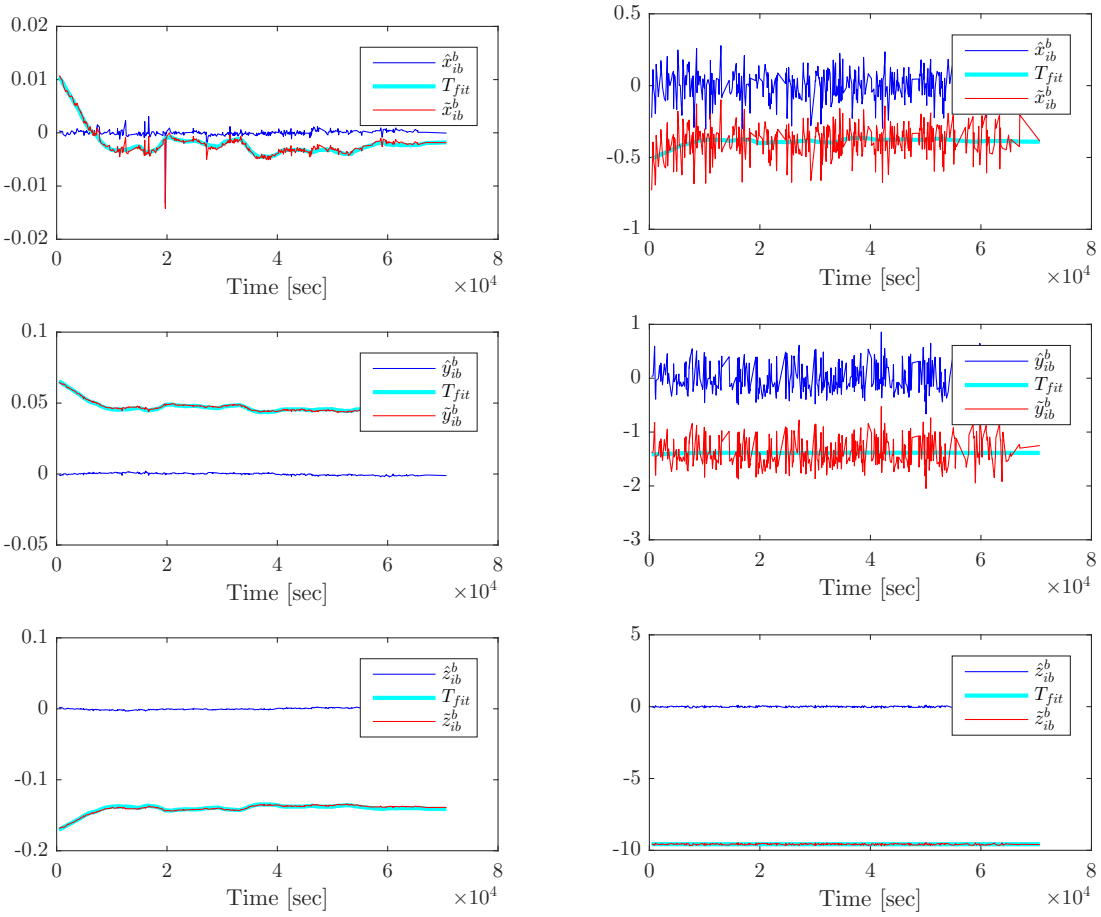
$$\begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \end{bmatrix} = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{y} \tag{5.9}$$

The offset c_2 does not provide useful information as the absolute temperature offset and the absolute sensor offset are totally independent. Therefore the inertial measurements $\tilde{\boldsymbol{y}}$ are only corrected in a relative way with respect to the temperature.

$$\hat{\boldsymbol{y}} = \tilde{\boldsymbol{y}} - \hat{c}_2 \boldsymbol{T} \tag{5.10}$$

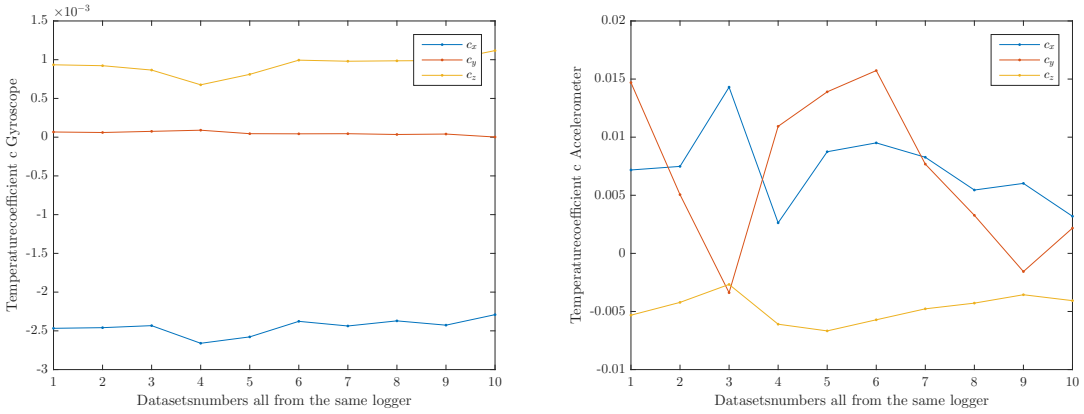
Figure 5.8 shows the results of the compensation algorithm noting that the offset has been added to achieve a better readability. The compensation for the gyroscope works excellent. The accelerometer is also corrected but less accurate due to the mentioned disturbances.

Figure 5.9 sums up the coefficients to a series of datasets. Obviously the technique generates more stable results for the gyroscope.



(a) Gyroscope (b) Accelerometer

Figure 5.8: Temperature compensated measurements (Body frame)



(a) Gyroscope (b) Accelerometer

Figure 5.9: Temperature coefficients from 10 datasets of one logger (Body frame)

5.3.6 Outlier Detection and Rejection Algorithm

In datasets and also laboratory tests [14] the inertial sensor data stream contains high peaks. In a stream of one axis the magnitude of these peaks is nearly always the same. The peaks do not appear consecutively most likely. With this information an non-linear outlier-detection and rejection algorithm is developed in Matlab, which filters for unrealistic slopes in the signal. When the derived signal contains a negative followed by a positive peak or a positive peak followed by a negative peak this is one of those outliers.

In figure 5.10 the red dots mark the points which are detected as outliers. The angular rate for these points gets interpolated by averaging about the two adjacent values for each case.

The algorithm is only implemented for the gyroscope. Outliers of the accelerometer are difficult to distinguish from the noise peaks, because the noise is comparatively higher.

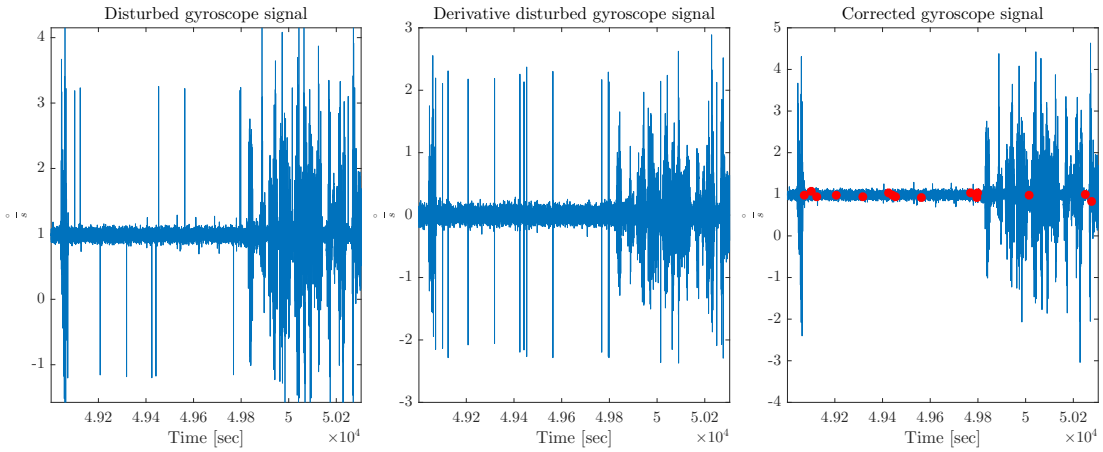


Figure 5.10: Outlier of gyroscope

5.4 Odometry and Vehicle Constraint

The drive train delivers the power from the engine to the driving wheels. Therefore a transmission converts the rotational speed of the engine to the average speed of all wheels on the drive shaft. As the buses have a rear-wheel drive the average speed of the drive shaft is adjusted by the differential for each of the rear wheels. This is especially important in curves where rear-wheels are rotating at different rates [23, p.101-102].

In the bus a high precision rotational speed sensor (Model: KITAS 2170/2171) provides a defined amount of pulses for each rotation of the drive shaft. An internal board computer processes those signals and passes a prepared signal to the data logger. Four pulses are generated for one meter distance [33, p.79]. The derivative of these odometry pulses with

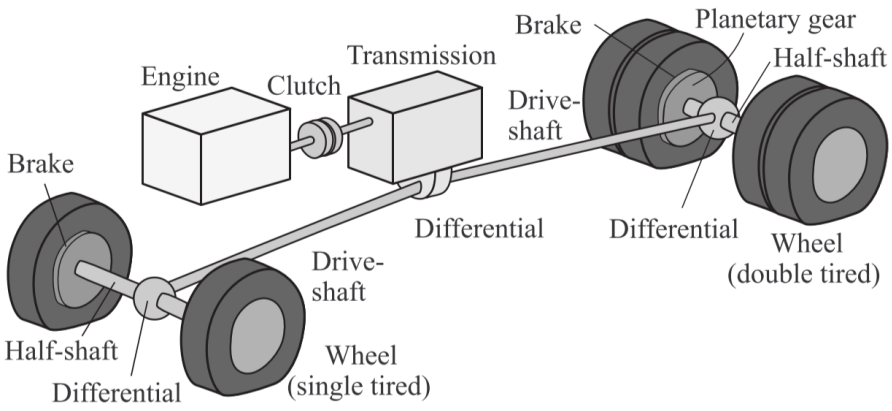


Figure 5.11: Drive train, drawing from [23, p.102]

respect to time gives the speed at the center of the rear axes. For a sampling time of $\tau_s = 0.1s$ the measurement has a maximum error of $\epsilon_O = 1.25\frac{m}{s}$.

The x-component of the rear velocity vector $\boldsymbol{v^r}$ in figure 5.13 represents the odometry-speed v_{odo}. By assuming that the wheels do not slip or jump the y- and z-component of $\boldsymbol{v^r}$ are both zero which is known as land vehicle constraint [9, p.641-642]. In the next chapter this property will be used as a velocity measurement.

To connect these measurements with the others the odometry velocity has to be known at the place of the IMU. From physics the velocity $\boldsymbol{v}$ at a point $\boldsymbol{r}$ from the centre of a rotating rigid body with angular velocity $\boldsymbol{\omega}$ is defined by

$$\boldsymbol{v} = \boldsymbol{\omega} \times \boldsymbol{r} \tag{5.11}$$

In figure 5.13 this radius $\boldsymbol{r}$ is the lever arm $\boldsymbol{l}^b_{rb}$ as the distance vector from the rear to the position of the IMU. Therefore

$$\begin{aligned} \boldsymbol{v}^b_{eb} - \boldsymbol{v}^b_{er} &= \boldsymbol{\omega}^b_{eb} \times \boldsymbol{l}^b_{rb} \\ \boldsymbol{v}^b_{er} &= \boldsymbol{v}^b_{eb} - \boldsymbol{\omega}^b_{eb} \times \boldsymbol{l}^b_{rb} \end{aligned} \tag{5.12}$$

The superscript b indicates the frame of the IMU called body frame. It has to be considered that the rear frame and the body frame are aligned differently. A coordinate frame transformation (4.2) of 5.12 will result in the measured velocity $\boldsymbol{v}^r_{er}$. The rotation rate of the Earth

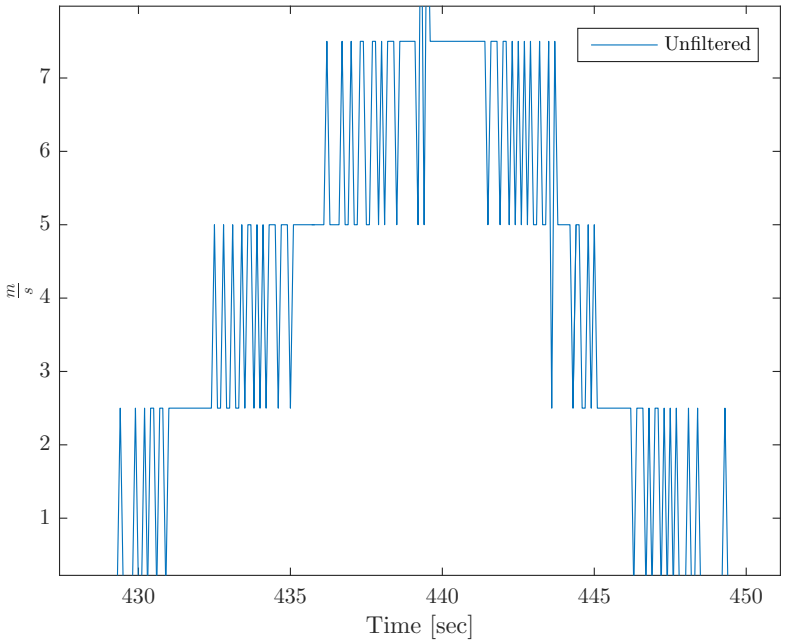


Figure 5.12: Raw derivative of odometry signal

can be neglected.

$$\begin{aligned} \boldsymbol{v}_{er}^{r} = \begin{bmatrix} v_{odo} \\ 0 \\ 0 \end{bmatrix} &= \boldsymbol{C}_{b}^{r} \left(\boldsymbol{v}_{eb}^{b} - \boldsymbol{\omega}_{ib}^{b} \times \boldsymbol{l}_{rb}^{b} \right) \\ &= \boldsymbol{C}_{b}^{r} \left(\boldsymbol{C}_{n}^{b} \boldsymbol{v}_{eb}^{n} + \boldsymbol{\omega}_{ib}^{b} \times \boldsymbol{l}_{br}^{b} \right) \end{aligned} \tag{5.13}$$

These equations are valid for the bi-articulated and normal bus.

The single-articulated bus measures the velocity at the trailor-axes. In this case equation

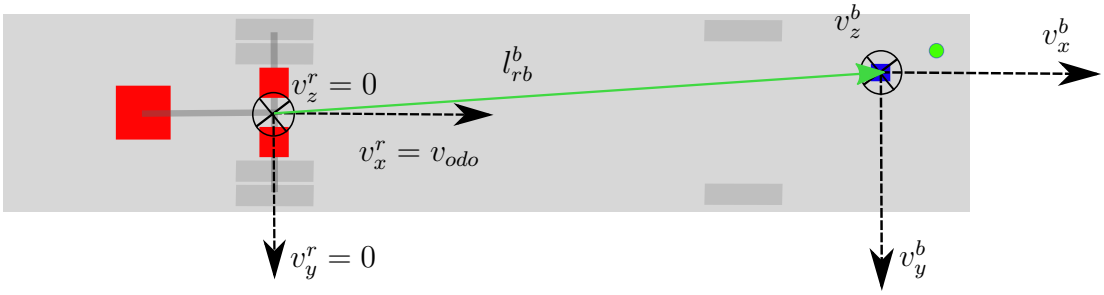


Figure 5.13: Odometry with vehicle constraint

5.13 is only valid for the velocity constraint components $v^r_{y,er}$ and $v^r_{z,er}$. In curves the assumption of a rigid body prevents the usage of v_{odo}. A trailer model in [23, p.293-298] may be used to derive the relevant equations.

5.5 Door Signal

The state of the bus door (open or closed) is recorded. With this information a bus station can be distinguished from other stops. It may be helpful in evaluating the pressure which is changing when opening or closing the door.

6 Implementation

The one-dimensional example 3.3 illustrates the performance of Kalman filtering in a simple way. Now the algorithm with its advantages is extended to our three-dimensional world for a multi-sensor system to get the finest position accuracy based on all information. Considering that the description of all sensor properties in Chapter 5 is used to build up a Kalman filter model similar to the one dimensional case. On top the theoretical concept of the linearisation technique from section 3.3.5 is applied.

6.1 Architecture

Architecture 6.1 enhances the idea of error states from the one dimensional example (Figure 3.11).

- The total state algorithm is replaced by a reference system called inertial navigation system (INS). It consists of an accelerometer and a gyroscope. They give the information of the earlier discussed definition of coordinate frames with their six degrees of freedom. A strapdown algorithm processes those measurements and describes the change of the body frame with respect to a navigation frame. Out of this the change in velocity and position in three-dimensional space can be determined for short times. The involved physical equations giving an exact solution are originally non-linear and therefore have to be linearised.

- Instead of one, there are now three measurements from the GPS-Receiver, the barometer and velocity constraint combined with odometer information. The equations restricting the movement with information about the vehicle model are also non-linear thus they must be linearised as well.

The GPS receiver generates its position autonomously. Including this as a position measurement results in an architecture which belongs to the type Loosely Coupled Integration. If the GPS pseudo range measurements (distances to each satellite) were used it would be called Tightly Coupled Integration [5, p.412-415].

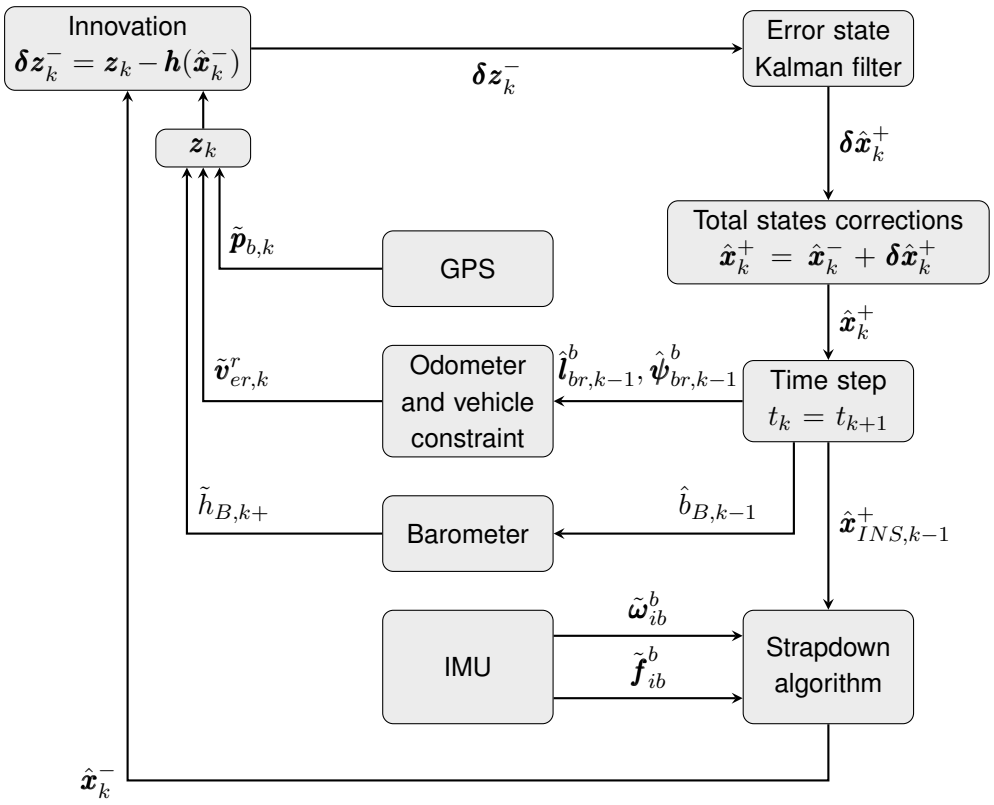


Figure 6.1: Architecture

6.2 Strapdown Algorithm

This section is based on the great work of [9, p.163-216]. The idea is to transform the specific force $\boldsymbol{f}_{ib}^b$ measured in the body frame into the reference frame by using the CTM $\boldsymbol{C}_b^n$ determined from the gyroscope data $\boldsymbol{\omega}_{ib}^b$. After this the transformed $\boldsymbol{f}_{ib}^n$ updates the old velocity and position information for the time interval τ_s, by applying physical equations of motion. Four steps of figure 6.2 are required to perform this and each will be discussed separately:

1. Alignment update
2. Transformation of the specific force
3. Velocity update
4. Position update

The approach is always to derive the differential equation and evaluating them discretised in an optimal way.

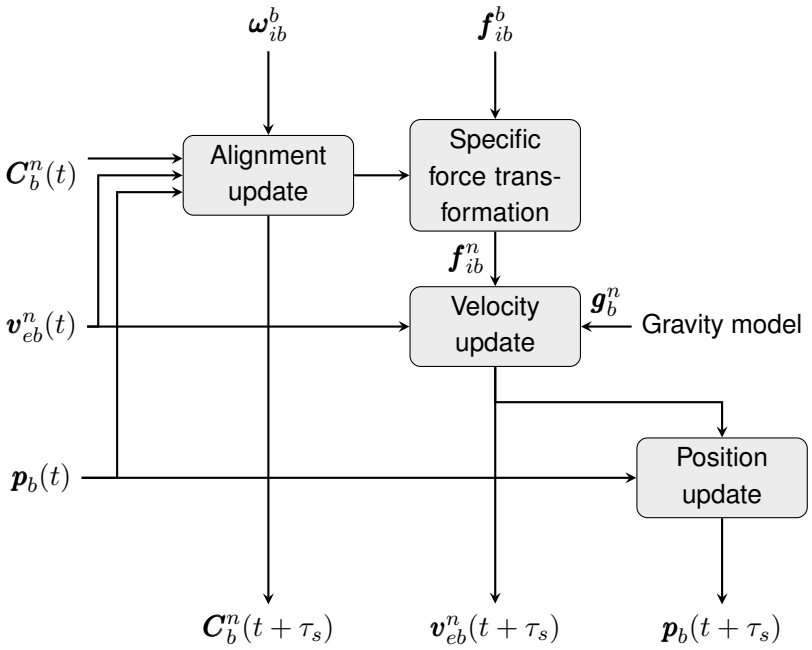


Figure 6.2: Block diagram describing local-navigation-frame equations [9, p.176]

6.2.1 Alignment update

The time derivative of the of the CTM is [9, p.45-46]

$$\dot{\boldsymbol{C}}_b^n = \boldsymbol{C}_b^n \boldsymbol{\Omega}_{nb}^b \qquad \boldsymbol{\Omega}_{nb}^b = \boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{ie}^b - \boldsymbol{\Omega}_{en}^b \tag{6.1}$$

The skew symmetric matrix form of the inertial angular velocity $[\boldsymbol{\omega}_{ib}^b\times]$ is calculated as follows:

$$\boldsymbol{\Omega}_{ib}^b = [\boldsymbol{\omega}_{ib}^b\times] = [\boldsymbol{\omega}_{ib}^b\times] = \begin{bmatrix} 0 & -\omega_{ib,z}^b & \omega_{ib,y}^b \\ \omega_{ib,z}^b & 0 & -\omega_{ib,x}^b \\ -\omega_{ib,y}^b & \omega_{ib,x}^b & 0 \end{bmatrix} \tag{6.2}$$

The Earth rate $\boldsymbol{\Omega}_{ie}^n$ describes the rotation of the Earth frame with respect to the inertial frame resolved in navigation frame.

$$\boldsymbol{\Omega}_{ie}^n = \omega_{ie} \begin{bmatrix} 0 & sinL_b & 0 \\ -sinL_b & 0 & -cosL_b \\ 0 & cosL_b & 0 \end{bmatrix} \qquad \omega_{ie} \approx 7.29 \cdot 10^{-5}\, rad/s \tag{6.3}$$

The transport rate $\boldsymbol{\Omega}_{en}^{n}$ is caused by movements of the vehicle on Earth's surface.

$$\boldsymbol{\Omega}_{en}^{n} = [\boldsymbol{\omega}_{en}^{n}\times] \qquad \boldsymbol{\omega}_{en}^{n} = \begin{bmatrix} v_{eb,E}^{n}/(R_E(L_b)+h_b) \\ -v_{eb,N}^{n}/(R_N(L_b)+h_b) \\ -v_{eb,E}^{n}\tan(L_b)/(R_E(L_b)+h_b) \end{bmatrix} \tag{6.4}$$

$$R_N(L_b) = \frac{R_0(1-e^2)}{(1-e^2 sin^2 L_b)^{3/2}} \tag{6.5}$$

$$R_E(L_b) = \frac{R_0}{\sqrt{1-e^2 sin^2 L_b}} \tag{6.6}$$

where:

$R_E(L_b)$: Geocentric radius of ellipsoid curvature

$R_N(L_b)$: Transverse radius of ellipsoid curvature

R_0: Equatorial radius of ellipsoid WGS84 (4.6)

e: Eccentricity of ellipsoid WGS84 (4.6)

Integrating this by using an exponential approach (3.26) and truncating the power-series expansion after the first order leads to [9, p.178]

$$\boldsymbol{C}_{b}^{n}(t+\tau_s) = \boldsymbol{C}_{b}^{n}(t)e^{\boldsymbol{\Omega}_{nb}^{b}\tau_s} \approx \boldsymbol{C}_{b}^{n}(t)(\boldsymbol{I}_3 + \boldsymbol{\Omega}_{ib}^{b}\tau_s) - (\boldsymbol{\Omega}_{ie}^{n}(t) + \boldsymbol{\Omega}_{en}^{n}(t))\boldsymbol{C}_{b}^{n}(t)\tau_s \tag{6.7}$$

In empiric tests the first order approximation has been insufficient, due to the fact that the linearisation error is integrated twice for the position update: An update rate of 100 Hz with a permanent angular rate of 280 °/s would cause an angle drift error of $1720\ \frac{^\circ}{hour}$ [9, p.184]. The error is only significant for the inertial angular rate $\boldsymbol{\omega}_{ib}^{b}$. Therefore a higher order truncation and substituting $\boldsymbol{\Omega}_{ib}^{b}\tau_s$ with the skew symmetric matrix of the increment angle $[\boldsymbol{\alpha}_{i}^{b}\times]$ results in

$$e^{\boldsymbol{\Omega}_{ib}^{b}\tau_s} = e^{[\boldsymbol{\alpha}_{i}^{b}\times]} = \sum_{r=0}^{\infty}\frac{[\boldsymbol{\alpha}_{i}^{b}\times]^r}{r!} \tag{6.8}$$

This can be effectively solved with the Rodrigues' formula using similarities to the power expansion series of sine and cosine [9, p.184].

$$e^{[\boldsymbol{\alpha}_{i}^{b}\times]} = \boldsymbol{I}_3 + \frac{\sin|\boldsymbol{\alpha}_{ib}^{b}|^2}{|\boldsymbol{\alpha}_{ib}^{b}|}[\boldsymbol{\alpha}_{ib}^{b}\times] + \frac{1-\cos|\boldsymbol{\alpha}_{ib}^{b}|}{|\boldsymbol{\alpha}_{ib}^{b}|^2}[\boldsymbol{\alpha}_{ib}^{b}\times]^2 \tag{6.9}$$

Together with the first order approximation for the Earth rate and transport rate, a more precise alignment update is

$$\boldsymbol{C}_b^n(t+\tau_s) \approx \boldsymbol{C}_b^n(t) e^{[\boldsymbol{\alpha}_i^b \times]} - (\boldsymbol{\Omega}_{ie}^n(t) + \boldsymbol{\Omega}_{en}^n(t)) \boldsymbol{C}_b^n(t) \tau_s \tag{6.10}$$

When $\boldsymbol{\alpha}_{ib}^b$ is very small ($< 1 \cdot 10^{-8}\, rad/s$), equation (6.7) is applied to avoid a division by zero. The Earth rate and transport rate can be neglected as they are smaller than the sensitivity of the angular rate sensor. For further applications of the algorithm with more precise inertial sensors the equations are still implemented.

6.2.2 Transformation of Specific Force

The specific force transformation into the navigation frame is

$$\boldsymbol{f}_{ib}^n = \boldsymbol{C}_b^n(t) \boldsymbol{f}_{ib}^b \tag{6.11}$$

The specific force measurement is an average over the time τ_s. The CTM should be averaged too [9, p.171].

$$\boldsymbol{f}_{ib}^n \approx \frac{1}{2} \left(\boldsymbol{C}_b^n(-) + \boldsymbol{C}_b^n(+) \right) \boldsymbol{f}_{ib}^b \tag{6.12}$$

6.2.3 Velocity Update

Differential equations of motion describe the velocity. Inertial acceleration is obtained by adding an acceleration term due to gravitation to the specific force [9, p.171].

$$\dot{\boldsymbol{v}}_{ib}^i = \boldsymbol{a}_{ib}^i = \boldsymbol{f}_{ib}^i + \boldsymbol{\gamma}_{ib}^i(r_{ib}^i) \tag{6.13}$$

The term results in the velocity resolved in ECI-frame. In vehicle navigation the velocity with respect to the surface is of great interest. Thus the inertial velocity has to be transformed into the rotating ECEF-frame, which adds Coriolis acceleration when moving on Earth, and a centrifugal acceleration. The force of gravitation combined with the centrifugal force adds up the amount of gravity.

$$\dot{\boldsymbol{v}}_{eb}^e = \boldsymbol{f}_{ib}^e + \underbrace{\boldsymbol{\gamma}_{ib}^i(r_{ib}^i) - \boldsymbol{\Omega}_{ie}^e \boldsymbol{\Omega}_{ie}^e \boldsymbol{r}_{eb}^e}_{\text{gravity: } \boldsymbol{g}_b^e(r_{eb}^e)} - 2\boldsymbol{\Omega}_{ie}^e \boldsymbol{v}_{eb}^e \tag{6.14}$$

As the chosen frame of reference is the navigation frame, the velocity has to be transformed again. A transport rate term is added describing the rotational rate of the navigation frame

with respect to the ECEF-frame.

$$\dot{\boldsymbol{v}}^n_{eb} = \boldsymbol{f}^n_{ib} + \boldsymbol{g}^n_b(\boldsymbol{r}^n_{eb}) - (\boldsymbol{\Omega}^n_{en} + 2\boldsymbol{\Omega}^n_{ie})\,\boldsymbol{v}^n_{eb} \tag{6.15}$$

For heights less than 10 km the gravity $\boldsymbol{g}^n_b(\boldsymbol{r}^n_{eb})$ can be approximated by [12, p.188-189]

$$\begin{aligned} \boldsymbol{g}^n_b(\boldsymbol{r}^n_{eb}) &= \boldsymbol{g}^n_b(L_b, h_b) \\ g^n_{b,N} &= -8.08 \cdot 10^{-9} h_b \sin 2L_b \\ g^n_{b,E} &= 0 \\ g^n_{b,D} &= g_0(L_b)\left(1 - \frac{2}{R_0}\left[1 + f(1 - 2\sin^2 L_b) + \frac{\omega^2_{ie} R^2_0 R_p}{\mu}\right] h_b + \frac{3}{R^2_0} h^2_b\right) \end{aligned} \tag{6.16}$$

where

$\mu = 3.986004418 \cdot 10^{14} \frac{m^3}{s^2}$ is the Earth's gravitational constant with its WGS84 value.

$g_0(L_b)$ is the gravity at specific latitude on the ellipsoid's surface and is approximated by the WGS84 model [9, p.70]

$$g_0(L_b) \approx 9.7803253359 \frac{1 + 0.001931853 \sin L^2}{\sqrt{1 - e^2 \sin L^2}} \frac{m}{s^2} \tag{6.17}$$

R_P, R_0, f are defined in (4.6) and ω_{ie} in (6.3)

The velocity update is then given through integrating (6.15).

$$\boldsymbol{v}^n_{eb}(t + \tau_s) \approx \boldsymbol{v}^n_{eb}(t) + [\boldsymbol{f}^n_{ib} + \boldsymbol{g}^n_b(L_b(t), h_b(t)) - (\boldsymbol{\Omega}^n_{en}(t) + 2\boldsymbol{\Omega}^n_{ie}(t))\,\boldsymbol{v}^n_{eb}(t)]\,\tau_s \tag{6.18}$$

6.2.4 Position Update

The origin of a navigation frame is defined by curvilinear coordinates for the purpose of easier readability. The derivative of those coordinates can be written as a function of velocity [9, p.61]. Thus,

$$\begin{aligned} \dot{L}_b &= \frac{v^n_{eb,N}}{R_N(L_b) + h_b} \\ \dot{\lambda}_b &= \frac{v^n_{eb,E}}{(R_E(L_b) + h_b)\cos L_b} \\ \dot{h}_b &= -v^n_{eb,D} \end{aligned} \tag{6.19}$$

Integrating 6.19 and neglecting the variations in $R_N(L_b)$, $R_E(L_b)$ and $\boldsymbol{v}_{eb}^n$ leads to

$$\begin{aligned}
h_b(t+\tau_s) &= h_b(t) - \frac{\tau_s}{2}(v_{eb,D}^n(t) + v_{eb,D}^n(t+\tau_s)) \\
L_b(t+\tau_s) &= L_b(t) + \frac{\tau_s}{2}\left(\frac{v_{eb,N}^n(t)}{R_N(L_b(t)) + h_b(t)} + \frac{v_{eb,N}^n(t+\tau_s)}{R_N(L_b(t)) + h_b(t+\tau_s)}\right) \\
\lambda_b(t+\tau_s) &= \lambda_b(t) + \frac{\tau_s}{2}\left(\frac{v_{eb,E}^n(t)}{[R_E(L_b(t)) + h_b(t)]\cos L_b(t)}\right. \\
&\qquad \left. + \frac{v_{eb,E}^n(t+\tau_s)}{[R_E(L_b(t+\tau_s)) + h_b(t+\tau_s)]\cos L_b(t+\tau_s)}\right)
\end{aligned} \tag{6.20}$$

6.3 Extended Kalman Filter Implementation

The extended Kalman Filter is able to apply the advantages of linear Kalman filtering on non-linear systems and measurements.

6.3.1 Reference System

The reference system processes measurements from an accelerometer (5.1) and a gyroscope (5.2). The main sources of error in a consumer-grade INS system are the static bias and the system noise. Therefore a suitable Kalman filter is supposed to model these errors. This can be done by selecting 15 error states.

$$\boldsymbol{x}_{INS}^n = \begin{bmatrix} \boldsymbol{\delta\psi}_{nb}^n \\ \boldsymbol{\delta v}_{eb}^n \\ \boldsymbol{\delta p}_b \\ \boldsymbol{\delta b}_a \\ \boldsymbol{\delta b}_g \end{bmatrix} \tag{6.21}$$

In the strapdown algorithm the alignment update is performed with a coordinate transformation matrices rather than directly working with euler angles. Therefore a differential equation for the error $\boldsymbol{\delta\psi}_{nb}^n$ has to be considered. The derivative of the angle is the angular rate. Taking the mean value from (5.2) is necessary to get rid of the noise. This gives the connection of the modelled error state $\boldsymbol{b}_g$ to the sensor measurement.

$$\dot{\boldsymbol{\psi}}_{nb}^n = \boldsymbol{C}_b^n E\left[\tilde{\boldsymbol{\omega}}_{ib}^b\right] = \boldsymbol{C}_b^n(\boldsymbol{b}_g + \boldsymbol{\omega}_{ib}^b) \tag{6.22}$$

From (3.72) the partial derivatives with respect to the states lead to the linear error state system matrix. Therefore

$$\left.\frac{d\dot{\boldsymbol{\psi}}^n_{nb}}{d\boldsymbol{b}_g}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \hat{\boldsymbol{C}}^n_b \tag{6.23}$$

From (6.15) the derivative of velocity is given. Effects resulting from the transport rate, Coriolis force and Earth rate are neglected, as they are below the sensitivity of the sensors: The gyroscope provides a resolution of 8.75mdps which exceeds Earth's angular rate of 4.2mdps. When moving at a speed 20m/s the Coriolis force applies an acceleration of up to 0.297mg which is much smaller than the resolution of the accelerometer with 2mg.

$$\begin{aligned}\dot{\boldsymbol{v}}^n_{eb} &= \boldsymbol{f}^n_{ib} + \boldsymbol{g}^n_b(L_b, h_b) \\ &= \boldsymbol{C}^n_b \boldsymbol{f}^b_{ib} + \boldsymbol{g}^n_b(L_b, h_b)\end{aligned} \tag{6.24}$$

The partial derivative of $\dot{\boldsymbol{v}}^n_{eb}$ with respect to $\boldsymbol{b}_a$ ic calculated using 5.1.

$$\left.\frac{d\dot{\boldsymbol{v}}^n_{eb}}{d\boldsymbol{b}_a}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \left.\frac{d\boldsymbol{C}^n_b(\boldsymbol{b_a} + \boldsymbol{f}^b_{ib})}{d\boldsymbol{b}_a}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \hat{\boldsymbol{C}}^n_b \tag{6.25}$$

For the derivative with respect to $\boldsymbol{\psi}^n_{nb}$ the true matrix $\boldsymbol{C}^n_b$ is defined analogously to the true state $\boldsymbol{x} = \hat{\boldsymbol{x}} + \boldsymbol{\delta x}$ (3.74).

$$\boldsymbol{C}^n_b = \boldsymbol{\delta C}^n_b \hat{\boldsymbol{C}}^n_b \tag{6.26}$$

$$\begin{aligned}\left.\frac{d\dot{\boldsymbol{v}}^n_{eb}}{d\boldsymbol{\psi}^n_{nb}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} &= \left.\frac{d\boldsymbol{\delta C}^n_b \hat{\boldsymbol{C}}^n_b \boldsymbol{f}^b_{ib}}{d\boldsymbol{\psi}^n_{nb}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \left.\frac{d(\boldsymbol{I}_3 - [\boldsymbol{\delta\psi}^n_{bn}\times])\hat{\boldsymbol{C}}^n_b \boldsymbol{f}^b_{ib}}{d\boldsymbol{\psi}^n_{nb}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} \\ &= -\left.\frac{d[\hat{\boldsymbol{C}}^n_b \boldsymbol{f}^b_{ib}\times]\boldsymbol{\delta\psi}^n_{nb}}{d\boldsymbol{\psi}^n_{nb}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = -[\hat{\boldsymbol{C}}^n_b \hat{\boldsymbol{f}}^b_{ib}\times]\end{aligned} \tag{6.27}$$

For the position state the differential equation is given by (6.19). The partial derivatives for $\dot{\boldsymbol{p}}_b$ having a significant influence are

$$\left.\frac{d\dot{L}_b}{dv^n_{eb,N}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \frac{1}{R_N(\hat{L}_b)+\hat{h}_b} \tag{6.28}$$

$$\left.\frac{d\dot{\lambda}_b}{dv^n_{eb,E}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \frac{1}{(R_E(\hat{L}_b)+\hat{h}_b)\cos\hat{L}_b} \tag{6.29}$$

$$\left.\frac{d\dot{h}_b}{dv^n_{eb,D}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = -1 \tag{6.30}$$

Finally the biases of the INS have to be modelled. In (5.8) an algorithm has been proposed to get rid of the temperature dependent dynamic bias. Therefore only the static bias has to be estimated which can described by a random constant.

$$\dot{\boldsymbol{b_a}} = \boldsymbol{0} \tag{6.31}$$

$$\dot{\boldsymbol{b_g}} = \boldsymbol{0} \tag{6.32}$$

For some small inaccuracies of the temperature compensation algorithm and vibration induced biases a small amount of noise is added to the constant at every iteration. This results in a random walk driven by white noise.

These complex equations for the system matrix can be written in a compact form.

$$\boldsymbol{F}_{INS} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \hat{\boldsymbol{C}}^n_b \\ -[\hat{\boldsymbol{C}}^n_b \hat{\boldsymbol{f}}^b_{ib}\times] & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \hat{\boldsymbol{C}}^n_b & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \hat{\boldsymbol{T}}^p_{r^n} & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} \qquad \hat{\boldsymbol{T}}^p_{r^n} = \begin{bmatrix} \frac{1}{R_N(\hat{L}_b)+\hat{h}_b} & 0 & 0 \\ 0 & \frac{1}{(R_E(\hat{L}_b)+\hat{h}_b)\cos\hat{L}_b} & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{6.33}$$

The matrix $\hat{\boldsymbol{T}}^p_{r^n}$ is also useful for converting other small perturbations into curvilinear coordinates e.g. standard deviations.

$$\boldsymbol{\delta p}_b = \hat{\boldsymbol{T}}^p_{r^n} \boldsymbol{\delta r}^n_{eb} \tag{6.34}$$

6.3.2 Aiding Sensors

The reference system is supported by measurements. The next sections present the connections from the measured quantities to the states. Additionally some measurements may

have unknown parameters in the measurements matrix which can be estimated by augmenting the system matrix with additional states.

GPS

The GPS-measurement is linearly connected to the position states. The lever arm of the GPS sensor is neglected in the system model as it has a very weak influence [9, p.600]. Therefore,

$$\boldsymbol{H}_{G,k} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{6.35}$$

Barometer

From the barometer measurement absolute height is calculated using 5.6. Assuming the function as linear for changes in height the bias of the barometer b_B and the noise v_B are defined in units of meters rather than pascal. Therefore the measurement equation is

$$\tilde{h}_B = h + b_B + w_B \tag{6.36}$$

The bias is unknown and has to be modelled in the Kalman filter as the state b_B. The initial uncertainty is given by the accuracy of the GPS-height while further changes depend mainly on the weather. Figure shows the error in height calculated by the sensor after perfect initialisation from the height database [31].

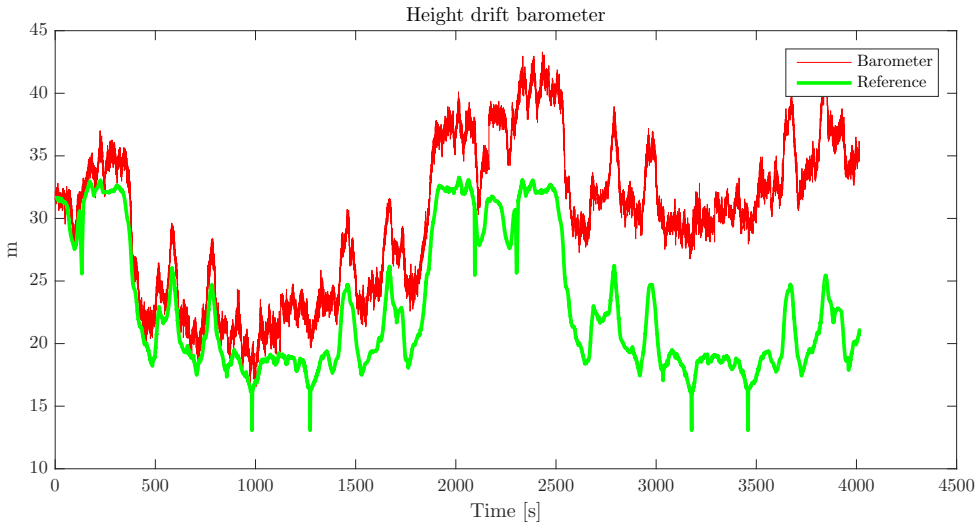


Figure 6.3: Height drift of the barometer solution due to changing weather conditions

Two datasets provided by the meteorological institute in Hamburg[10] are evaluated containing the pressure data for a static position of about 3 months in summer and winter. The

auto-covariance in figures 6.4 and 6.5 is obviously exponentially correlated. After fitting equation (3.16) using the least square fit function 'lsqcurvefit' in Matlab, the parameters for a Gauss Markov process can be determined.

$$\dot{b}_B = -\frac{1}{\tau_{c,B}} b_B + \sigma_B \sqrt{\frac{2}{\tau_{c,B}}} w_u(t) \tag{6.37}$$

Obviously the standard deviations and the correlation times are changing between the seasons. For the Kalman filter configuration the parameters are chosen as $\tau_c = 20000s$ and $\sigma = 60m$ because they are likely to perform very well in empiric tests.

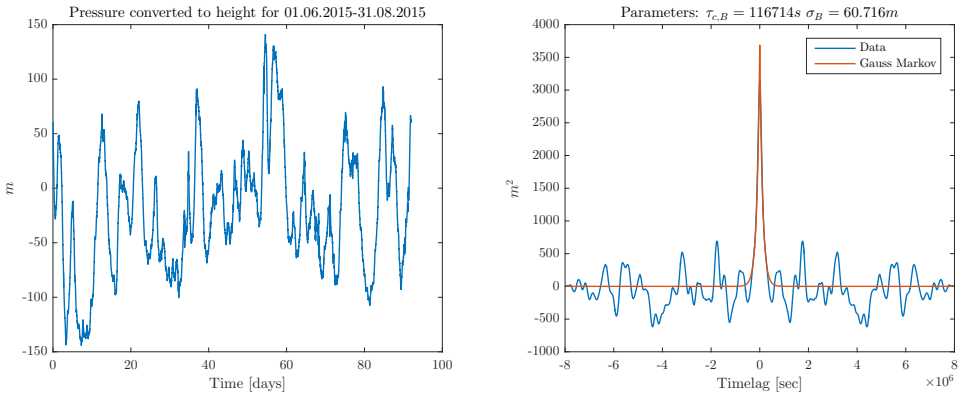


Figure 6.4: Gauss Markov parameters for height of pressure (Summer 2015)

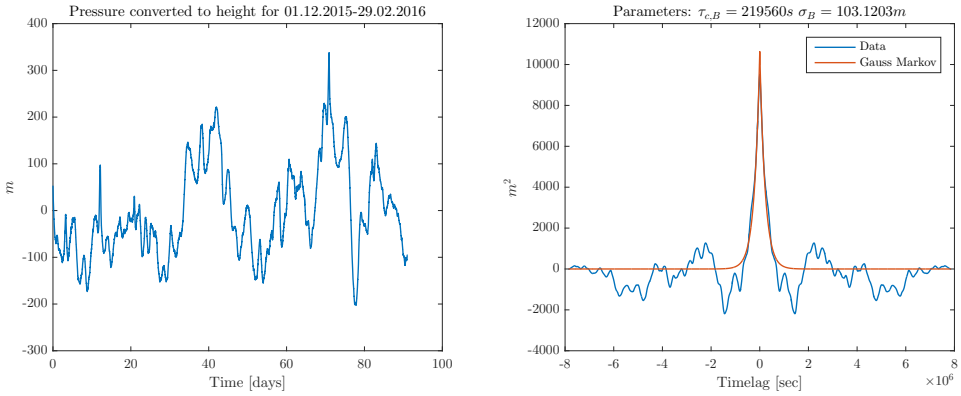


Figure 6.5: Gauss Markov parameters for height of pressure (Winter 2015/16)

Due to the linear connection the measurement matrix can be easily written down.

$$\boldsymbol{H}_{B,k} = \begin{bmatrix} 0_8 & 1 & 0_6 & 1 \end{bmatrix} \tag{6.38}$$

The measurement equipment has to be corrected by rearranging equation 6.36.

$$\hat{h}_B = \tilde{h}_B - b_B \tag{6.39}$$

Odometry and Velocity Constraint

Equation 5.13 describes the connection between the velocity of the body frame and the rear frame.

$$\boldsymbol{v}_{er}^{r} = \boldsymbol{C}_{b}^{r} \left(\boldsymbol{C}_{n}^{b} \boldsymbol{v}_{eb}^{n} + \boldsymbol{\omega}_{ib}^{b} \times \boldsymbol{l}_{br}^{b} \right)$$

Since the lever arm $\boldsymbol{l}_{br}^{b}$ and the alignment $\boldsymbol{C}_{b}^{r}(\boldsymbol{\psi}_{br}^{b})$ are not exactly known this uncertainty must be modelled in the Kalman filter with additional error states.

$$\boldsymbol{x}_O = \begin{bmatrix} \boldsymbol{\delta\psi}_{br}^{b} \\ \boldsymbol{\delta l}_{br}^{b} \end{bmatrix} \tag{6.40}$$

The processes are modelled as a random constant. Thus, the system matrix becomes zero.

$$\boldsymbol{F}_O = \begin{bmatrix} \boldsymbol{0}_3 \\ \boldsymbol{0}_3 \end{bmatrix} \tag{6.41}$$

By defining $\boldsymbol{C}_n^b = \boldsymbol{\delta C}_n^b \hat{\boldsymbol{C}}_n^b$ (6.26) and using (4.5) the derivatives with respect to all significant states are obtained.

$$\begin{aligned}
\left.\frac{d\boldsymbol{v}_{er}^r}{d\boldsymbol{\psi}_{nb}^n}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} &= \left.\frac{d\boldsymbol{C}_b^r\left(\boldsymbol{C}_n^b\boldsymbol{v}_{eb}^n + \boldsymbol{\omega}_{ib}^b \times \boldsymbol{l}_{br}^b\right)}{d\boldsymbol{\psi}_{nb}^n}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \left.\frac{d\boldsymbol{C}_b^r\left(\boldsymbol{\delta C}_n^b\hat{\boldsymbol{C}}_n^b\boldsymbol{v}_{eb}^n\right)}{d\boldsymbol{\psi}_{nb}^n}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} \\
&= \left.\frac{d\boldsymbol{C}_b^r\left((\boldsymbol{I}_3 - [\boldsymbol{\delta\psi}_{nb}^n\times])\hat{\boldsymbol{C}}_n^b\boldsymbol{v}_{eb}^n\right)}{d\boldsymbol{\psi}_{nb}^n}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \left.\frac{d\boldsymbol{C}_b^r\left(-[\boldsymbol{\delta\psi}_{nb}^n\times])\hat{\boldsymbol{C}}_n^b\boldsymbol{v}_{eb}^n\right)}{d\boldsymbol{\psi}_{nb}^n}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} \\
&= \left.\frac{d\boldsymbol{C}_b^r\left([\hat{\boldsymbol{C}}_n^b\boldsymbol{v}_{eb}^n\times]\boldsymbol{\delta\psi}_{nb}^n)\right)}{d\boldsymbol{\psi}_{nb}^n}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \hat{\boldsymbol{C}}_b^r[\hat{\boldsymbol{C}}_n^b\hat{\boldsymbol{v}}_{eb}^n\times]
\end{aligned} \tag{6.42}$$

$$\left.\frac{d\boldsymbol{v}_{er}^r}{d\boldsymbol{v}_{eb}^n}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \left.\frac{d\boldsymbol{C}_b^r\left(\boldsymbol{C}_n^b\boldsymbol{v}_{eb}^n + \boldsymbol{\omega}_{ib}^b \times \boldsymbol{l}_{br}^b\right)}{d\boldsymbol{v}_{eb}^n}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \hat{\boldsymbol{C}}_b^r\hat{\boldsymbol{C}}_n^b \tag{6.43}$$

$$\begin{aligned}
\left.\frac{d\boldsymbol{v}_{er}^r}{d\boldsymbol{b}_g}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} &= \left.\frac{d\boldsymbol{\omega}_{ib}^b \times \boldsymbol{l}_{br}^b}{d\boldsymbol{b}_g}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = -\left.\frac{d[\boldsymbol{l}_{br}^b\times]\boldsymbol{\omega}_{ib}^b}{d\boldsymbol{b}_g}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = -\left.\frac{d[\boldsymbol{l}_{br}^b\times](\hat{\boldsymbol{\omega}}_{ib}^b + \boldsymbol{\delta b}_g)}{d\boldsymbol{b}_g}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} \\
&= -[\hat{\boldsymbol{l}}_{br}^b\times]
\end{aligned} \tag{6.44}$$

$$\left.\frac{d\boldsymbol{v}_{er}^r}{d\boldsymbol{l}_{br}^b}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \hat{\boldsymbol{C}}_b^r[\hat{\boldsymbol{\omega}}_{ib}^b\times] \tag{6.45}$$

$$\begin{aligned}
\left.\frac{d\boldsymbol{v}_{er}^r}{d\boldsymbol{\psi}_{br}^b}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} &= \left.\frac{d\boldsymbol{C}_b^r\boldsymbol{v}_{er}^b}{d\boldsymbol{\psi}_{br}^b}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \left.\frac{d\boldsymbol{\delta C}_b^r\hat{\boldsymbol{C}}_b^r\boldsymbol{v}_{er}^b}{d\boldsymbol{\psi}_{br}^b}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \left.\frac{d(\boldsymbol{I}_3 - [\boldsymbol{\delta\psi}_{br}^b\times])\hat{\boldsymbol{C}}_b^r\boldsymbol{v}_{er}^b}{d\boldsymbol{\psi}_{br}^b}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} \\
&= \left.\frac{d([\hat{\boldsymbol{C}}_b^r\boldsymbol{v}_{er}^b\times])\boldsymbol{\delta\psi}_{br}^b}{d\boldsymbol{\psi}_{br}^b}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = [\hat{\boldsymbol{C}}_b^r\hat{\boldsymbol{v}}_{er}^b\times]
\end{aligned} \tag{6.46}$$

$$\boldsymbol{H}_{O,k} = \begin{bmatrix}\hat{\boldsymbol{C}}_b^r[\hat{\boldsymbol{C}}_n^b\hat{\boldsymbol{v}}_{eb}^n\times] & \hat{\boldsymbol{C}}_b^r\hat{\boldsymbol{C}}_n^b & 0_6 & -[\hat{\boldsymbol{l}}_{br}^b\times] & 0 & \hat{\boldsymbol{C}}_b^r[\hat{\boldsymbol{\omega}}_{ib}^b\times] & [\hat{\boldsymbol{C}}_b^r\hat{\boldsymbol{v}}_{er}^b\times]\end{bmatrix} \tag{6.47}$$

In [1] a non-linear observability analysis shows that the roll angle of $\boldsymbol{C}_b^r$ is unobservable. Therefore the state ϕ_b^r has to be neglected in state model or initialised with a very small uncertainty. The latter is done in this thesis.

Zero Angular Rate Update

When the vehicle is standing still (stationary phase) no inertial rotation affects the gyroscope neglecting effects from Earth's rotation. Therefore the bias is directly observable. The Zero

angular rate update is linear connected to the bias state b_g. Thus,

$$\boldsymbol{H}_Z = \begin{bmatrix} \mathbf{0}_{12} & \boldsymbol{I}_3 \end{bmatrix} \tag{6.48}$$

For post-processing an alternative is the application of a temperature compensation algorithm (5.3.5). In this case stationary phases can be skipped completely. This has shown more stable results than updates in stationary phases. These have sometimes disturbed the state covariance matrix possibly leading to unstable results. Also the smoother algorithm performs more accurate when the state covariance matrix does not change in stationary phases. For real time applications an additional covariance matrix for stationary phases may be considered.

6.3.3 Discrete system

The continuous matrix $\boldsymbol{F}$ is given by the reference system $\boldsymbol{F}_{INS}$ and the extended states for the measurement equipment $\boldsymbol{F}_O$ and $\boldsymbol{F}_B$ leading to the full system. The average from the derivative $\dot{\boldsymbol{x}}$ eliminates the noise-vector in following equation.

$$E\left[\dot{\boldsymbol{x}}\right] = E\begin{bmatrix} \dot{\boldsymbol{\delta\psi}}^n_{nb} \\ \dot{\boldsymbol{\delta v}}^n_{eb} \\ \dot{\boldsymbol{\delta p}}_b \\ \dot{\boldsymbol{\delta b}}_a \\ \dot{\boldsymbol{\delta b}}_g \\ \dot{\boldsymbol{\delta\psi}}^b_{br} \\ \dot{\boldsymbol{\delta l}}^b_{br} \\ \dot{\delta b}_B \end{bmatrix} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \hat{\boldsymbol{C}}^n_b & \mathbf{0}_6 & 0 \\ -[\hat{\boldsymbol{C}}^n_b \hat{\boldsymbol{f}}^b_{ib}\times] & \mathbf{0}_3 & \mathbf{0}_3 & \hat{\boldsymbol{C}}^n_b & \mathbf{0}_3 & \mathbf{0}_6 & 0 \\ \mathbf{0}_3 & \hat{\boldsymbol{T}}^p_{r^n} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_6 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_6 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_6 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_6 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_6 & 0 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_6 & -\frac{1}{\tau_{c,B}} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\delta\psi}^n_{nb} \\ \boldsymbol{\delta v}^n_{eb} \\ \boldsymbol{\delta p}_b \\ \boldsymbol{\delta b}_a \\ \boldsymbol{\delta b}_g \\ \boldsymbol{\delta\psi}^b_{br} \\ \boldsymbol{\delta l}^b_{br} \\ \delta b_B \end{bmatrix} \tag{6.49}$$

Equation (3.26) is used to transform the continuous system matrix $\boldsymbol{F}$ to the discrete transition matrix $\boldsymbol{\phi}_{k-1}$.

$$\boldsymbol{\phi}_{k-1} = e^{\boldsymbol{F}_{k-1}\tau_s} \approx \boldsymbol{I} + \boldsymbol{F}_{k-1}\tau_s$$

Applying (3.26) on (6.49) gives the discrete system matrix.

$$\hat{\boldsymbol{x}}_{\boldsymbol{k}} = \begin{bmatrix} \boldsymbol{\delta\psi}^n_{nb} \\ \boldsymbol{\delta v}^n_{eb} \\ \boldsymbol{\delta p}_b \\ \boldsymbol{\delta b}_a \\ \boldsymbol{\delta b}_g \\ \boldsymbol{\delta\psi}^b_{br} \\ \boldsymbol{\delta l}^b_{br} \\ \delta b_B \end{bmatrix}_k = \underbrace{\begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \hat{\boldsymbol{C}}^n_b\tau_s & \boldsymbol{0}_3 & \boldsymbol{0}_3 & 0 \\ -[\hat{\boldsymbol{C}}^n_b\hat{\boldsymbol{f}}^b_{ib}\times]\tau_s & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \hat{\boldsymbol{C}}^n_b\tau_s & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & 0 \\ \boldsymbol{0}_3 & \hat{\boldsymbol{T}}^p_{r^n}\tau_s & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & 0 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & 0 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & 0 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & 0 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & 0 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & e^{-\frac{1}{\tau_{c,B}}\tau_s} \end{bmatrix}}_{\boldsymbol{\phi}_{k-1}} \cdot \begin{bmatrix} \boldsymbol{\delta\psi}^n_{nb} \\ \boldsymbol{\delta v}^n_{eb} \\ \boldsymbol{\delta p}_b \\ \boldsymbol{\delta b}_a \\ \boldsymbol{\delta b}_g \\ \boldsymbol{\delta\psi}^b_{br} \\ \boldsymbol{\delta l}^b_{br} \\ \delta b_B \end{bmatrix}_{k-1} \tag{6.50}$$

6.3.4 Initialisation

The initialisation process is restricted for stationary phases with a minimum resting time T_S. These phases can be found while evaluating the velocity of the odometer.

- The initial position is determined by averaging the GPS-results during the stationary phase over time T_S.
- The initial velocity is zero as the vehicle is detected stationary.
- The initial bias of the gyroscope b_g is determined through averaging the angular rate results. All non-zero values arise from the sensor offset or noise as the vehicle is stationary. The noise is suppressed by averaging.
- The initial alignment of roll and pitch is achieved by evaluating the vector of gravity measured by the accelerometer [9, p.198].

$$\begin{aligned} \phi_{nb} &= \arctan_2\left[-f^b_{ib,y}, -f^b_{ib,z}\right] \\ \theta_{nb} &= \arctan\left[\frac{f^b_{ib,x}}{\sqrt{{f^b_{ib,y}}^2 + {f^b_{ib,z}}^2}}\right] \end{aligned} \tag{6.51}$$

The accelerometer error $\boldsymbol{\delta f}^b_{ib}$ is included in this calculation. As a consequence the roll

and pitch uncertainty becomes [9, p.198]

$$\begin{aligned}\sigma_{\phi_{nb}} &= \frac{f^b_{ib,z}\delta f^b_{ib,y} - f^b_{ib,y}\delta f^b_{ib,z}}{{f^b_{ib,y}}^2 + {f^b_{ib,z}}^2} \\ \sigma_{\theta_{nb}} &= \frac{\left({f^b_{ib,y}}^2 + {f^b_{ib,z}}^2\right)\delta f^b_{ib,x} - f^b_{ib,x}f^b_{ib,y}\delta f^b_{ib,y} - f^b_{ib,x}f^b_{ib,z}\delta f^b_{ib,z}}{\left({f^b_{ib,x}}^2 + {f^b_{ib,y}}^2 + {f^b_{ib,z}}^2\right)\sqrt{{f^b_{ib,y}}^2 + {f^b_{ib,z}}^2}}\end{aligned} \tag{6.52}$$

Most of $\boldsymbol{\delta f}^b_{ib}$ originates from the bias uncertainty σ_{b_a}. From the data sheet $\sigma_{b_a} = 60\ mg$ [26, p.11] and therefore the angle uncertainty for total horizontal alignment is $\sigma_{\theta_{nb}} = \sigma_{\phi_{nb}} = 3.4°$.

- The initial accelerometer bias b_a can not be calculated due to the unknown alignment. It is only observable during dynamical movement phases [3, p.344-345].
- The initial misalignment of the body frame to the rear frame accounts for an ideal installation zero and therefore $\boldsymbol{\psi}^b_{br} = [0; 0; 0]°$.
- Section (5.1) leads to an initial lever arm from the body to the rear frame of approximately $\boldsymbol{l}^b_{br} = [-7.5; 1; 2]m$. Tests have shown that the Kalman filter converges to these values when initialised to zero. The difference in size of the leverarm between normal and articulated bus is therefore irrelevant for initial values.
- The bias of the barometer is initialised with the averaged height of the GPS signal of the stationary phase.

For a determination error greater than $\pm 2°$ the small angle approximation may become invalid and the whole system unstable [9, p.595,619-620]. This situation is possible as $\sigma_{\theta_{nb}} = \sigma_{\phi_{nb}} = 3.4°$. Even more crucial is the initialisation of the yaw-angle which will be done using the magnetometer.

Magnetometer Alignment

The output of the magnetometer is constantly disturbed by a soft iron scale factor M_m and a hard iron offset b_m(5.3). These errors may exceed the strength of the Earth's magnetic field ($0.3\ G$). Therefore a calibration procedure is required to eliminate this error. The algorithm proposed in [19] fits a sphere into multiple data points. Drawbacks are that this technique estimates only the hard iron offset and needs distributed data points in all directions. A land-vehicle performs only turns in horizontal alignment. Thus, this is not a stable approach as tests with multiple datasets have shown.

In order to obtain a stable and accurate initial alignment a two-dimensional technique for magnetometer calibration is developed in further steps. The idea is that a perfect calibrated

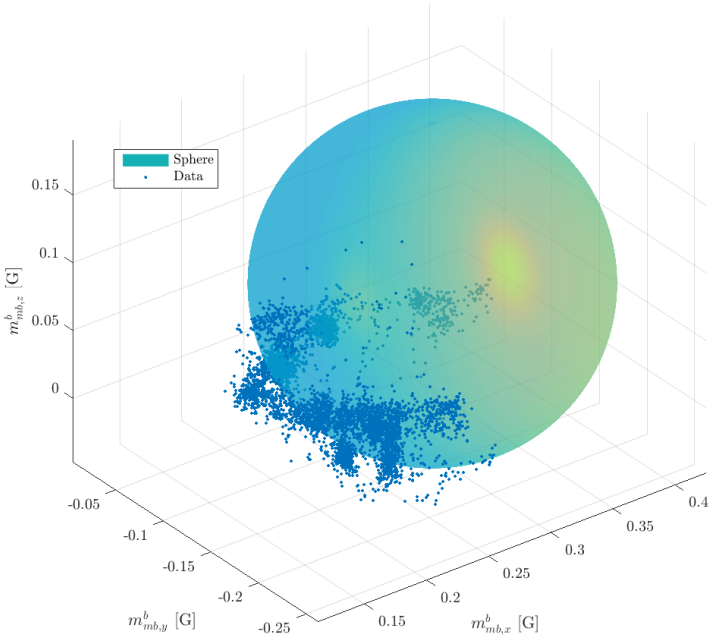


Figure 6.6: Sphere fit of 5000 data points of magnetometer

magnetometer gives data points in form of an ideal circle centred in the origin of the sensor frame when rotating about the z-axis (yaw). The dataset can be filtered for data points which exceeds a yaw-rate threshold. This leads to a set of data points with equally distributed magnetometer measurements illustrated in figure 6.7a. Obviously the form is rather an ellipse than a circle which indicates a soft iron effect. The fact that it is not centred in the origin of the xy-plane which indicates a hard iron offset. An ellipse can be described mathematically by five parameters in parametric form[16, p.11].

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} cos(t) \\ sin(t) \end{bmatrix} \tag{6.53}$$

Another way to describe ellipses is the conic (Cartesian) form 6.54. It is faster to minimize with linear least square methods than fitting directly to the parameter form [16, p.52].

$$0 = Ax^2 + Bxy + Cy^2 + Dx + Ey + F \tag{6.54}$$

In this thesis a Matlab script [7] fits data to the conic representation with least square methods and converts the conic to the parameter form (Figure 6.7b). A direct least square method for fitting data to an ellipse is presented in [20]. The calibration is done in following steps:

1. Correcting the offset $\boldsymbol{b}_{m,xy} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$ illustrated in figure 6.7c.
2. Rotating the data into the elliptic frame by applying a 2D Rotation with $\boldsymbol{R}(\phi) = \begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix}$ illustrated in figure 6.7d.
3. Scale elliptic data to circle data with scaling matrix $S = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$ illustrated in figure 6.7e.
4. Rotating the data back into the body frame with $\boldsymbol{R}(\phi)^T$ illustrated in figure 6.7f.

From (5.3) the measurement resolved in a xy-plane is

$$\tilde{\boldsymbol{m}}^b_{mb,xy} = \boldsymbol{b}_{m,xy} + (\boldsymbol{I}_2 + \boldsymbol{M}_m)\boldsymbol{m}^b_{mb,xy} \tag{6.55}$$

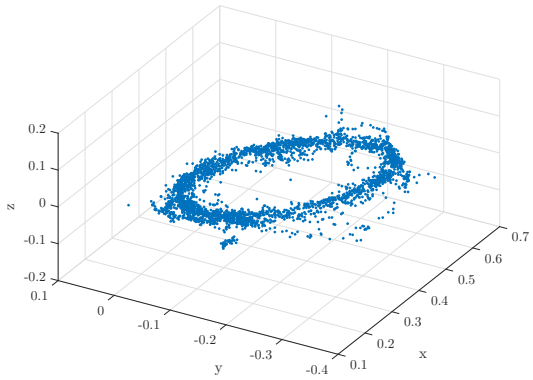
$$\tag{6.56}$$

Rearranging leads to the corrected magnetometer measurements.

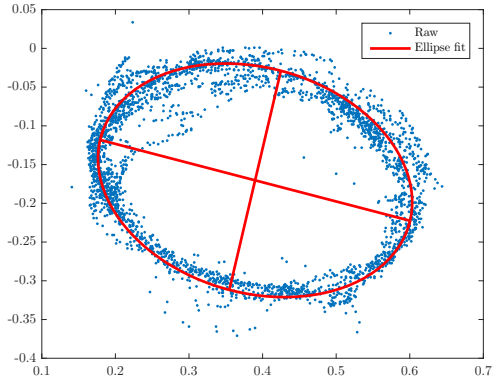
$$\hat{\boldsymbol{m}}^b_{mb,xy} = (\boldsymbol{I}_2 + \boldsymbol{M}_m)^{-1}(\tilde{\boldsymbol{m}}^b_{mb,xy} - \boldsymbol{b}_{m,xy}) \tag{6.57}$$

$$= \boldsymbol{R}(\phi)^T \boldsymbol{S} \boldsymbol{R}(\phi)(\tilde{\boldsymbol{m}}^b_{mb,xy} - \boldsymbol{b}_{m,xy}) \tag{6.58}$$

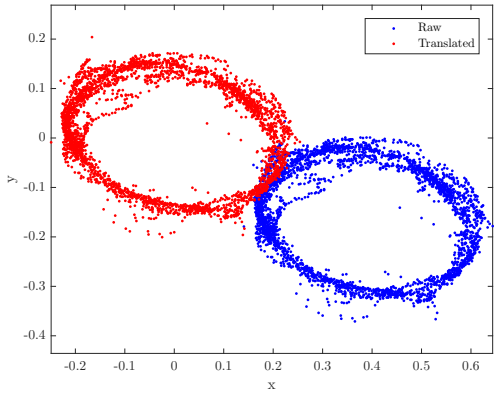
This calibration method does not correct the absolute values of the magnetometer measurement. However, when calculating the yaw-angle from the measurement the absolute value cancels out. Applying (5.4) and (5.5) leads to the tilt compensated yaw angle, noting that only the x- and y-axis are calibrated by the elliptic technique. In the initialisation phase all sampled magnetometer measurements are median averaged to eliminate the sensor noise and possible outliers.



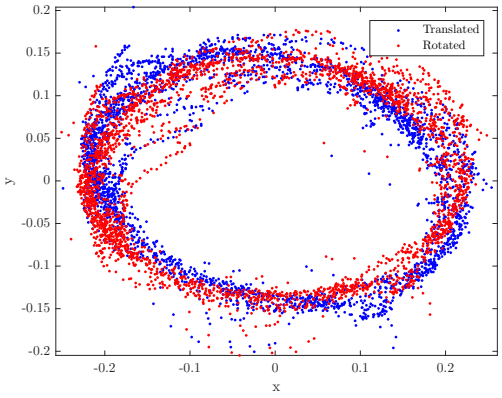
(a) Raw data 3D



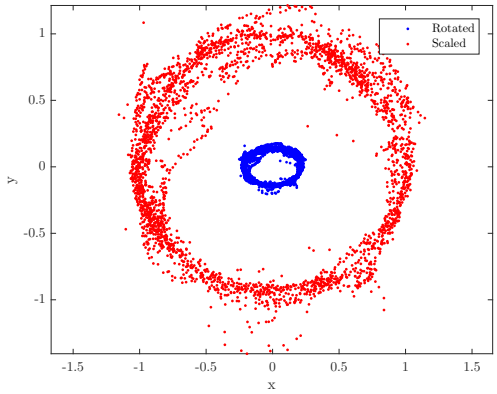
(b) 2D elliptic fit



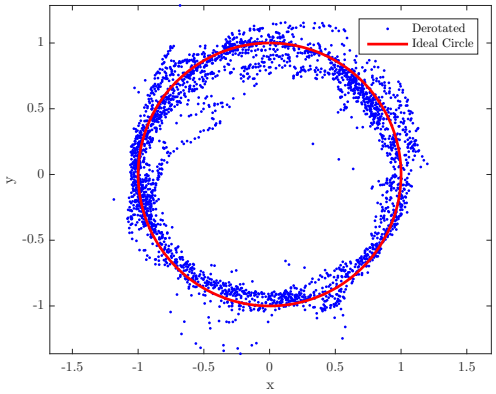
(c) Translation



(d) Rotation



(e) Scaling



(f) Derotation

Figure 6.7: Magnetormeter elliptic calibration

Initial System Covariance

The choice for the initial uncertainties is critical to get the extended Kalman filter started. A large $\boldsymbol{P}_0^-$ combined with low measurement noise leads the state covariance matrix to fall from a very large to a small value which could result in numerical problems. A way to prevent this is a choosing a higher noise in the measurement-matrix $\boldsymbol{R}_k$ for the first steps [3, p.259-260]. In this thesis the phase is called calibration phase. The choice for the initial uncertainties is found in the characteristics of the sensor equipment and the vehicle.

- $\boldsymbol{P}_0^-[\boldsymbol{\psi}_{nb}^n]$: The uncertainty for roll ϕ_{nb}^n and pitch θ_{nb}^n angles originates from the accelerometer offset which leads to the standard deviation $\sigma_{\theta_{nb}} = \sigma_{\phi_{nb}} = 3.4^\circ$ (6.52). The uncertainty for the yaw angle ψ_{nb}^n depends on the quality of magnetometer measurements. From comparisons with the direction of streets and calibrated magnetometer the uncertainty results in $\sigma_{\psi_{nb}^n} = 3^\circ$.

- $\boldsymbol{P}_0^-[\boldsymbol{v}_{nb}^n]$: The vehicle is for sure stationary due to the accurate odometry signal. For stability reasons $\sigma_{\boldsymbol{v}_{nb}^n} = 0.1\ \frac{m}{s}$.

- $\boldsymbol{P}_0^-[\boldsymbol{p}^b]$: The GPS-quality determines the initial uncertainty for the position. To give the filter enough flexibility the uncertainties are chosen as $\sigma_{lat,long} = 15m$ and $\sigma_{height} = 20m$.

- $\boldsymbol{P}_0^-[\boldsymbol{b_a}]$: As visible on the datasheet the accelerometer has an initial static bias uncertainty of $\sigma_{\boldsymbol{b}_a} = 60mg \approx 0.5\frac{m}{s^2}$ [26]. The converged biases in tests confirm this value.

- $\boldsymbol{P}_0^-[\boldsymbol{b_g}]$: Due to the stationary phase the gyroscope bias is directly observable. Therefore a small uncertainty $\sigma_{\boldsymbol{b_g}} = 0.5\frac{^\circ}{s}$ is chosen accounting for zero-angular-rate errors due to bus tilts or passenger movement.

- $\boldsymbol{P}_0^-[\boldsymbol{\psi}_{br}^b]$: The angle between body and rear frame has an estimated uncertainty of $\sigma_{\boldsymbol{\psi}_{br}^b} = 3^\circ$. Tests with multiple datasets have shown that the yaw angle ψ_{br}^b does not converge. To prevent instabilities the initial system covariance is set to zero. The same applies for the unobservable roll angle ϕ_{br}^b.

- $\boldsymbol{P}_0^-[\boldsymbol{l}_{br}^r]$: The lever arm has been estimated referring to the size of the vehicle in 5.1. To account for errors in this estimation the uncertainty is chosen as $\sigma_{\boldsymbol{l}_{br}^r} = [0.5; 0.2; 0.2]m$.

- $\boldsymbol{P}_0^-[b_B]$: The uncertainty of the barometer is the same as the GPS-height uncertainty resulting in $\sigma_{b_B} = 20m$.

6.3.5 System Noise

The system noise parameters are crucial for the development of error state covariances $\boldsymbol{P}$. With every iteration a specific amount of noise represented by the Q-Matrix is added to the P-Matrix. A greater variance in the states leads the Kalman filter to more weight in updates to the measurements than the reference navigation solution. In the following sections the parameters for the discrete system noise covariance matrix Q_{k-1} (3.44) have to be considered.

Gyroscope and Accelerometer Noise

The white system noise represents an inertial sensor noise and is the random input for the system. Due to the equations of motion the sensor white noise gets integrated several times similar to the random walk model of 3.1.5. With respect to time the variance in position determined by the INS has a quadratic slope for the accelerometer noise and a cubic slope for the gyroscope noise. An algorithm in Matlab calculates the standard deviation of the sensors for each stationary phase with closed doors and takes the average. Applying (3.14) with $B_g = 12.5Hz$ for the gyroscope and $B_a = 3Hz$ for the accelerometer leads to the power spectral densities A_S.

The results for several datasets of the same logger are illustrated in figure 6.8. The noise in the Kalman filter is increased by a factor of 2.5 to account for vibrations caused by the movement on the road and by the engine of the vehicle. The root noise densities R_S describing the inertial sensor noise, must be converted from the body to the navigation frame as the resolving axis of the states are in the navigation frame.

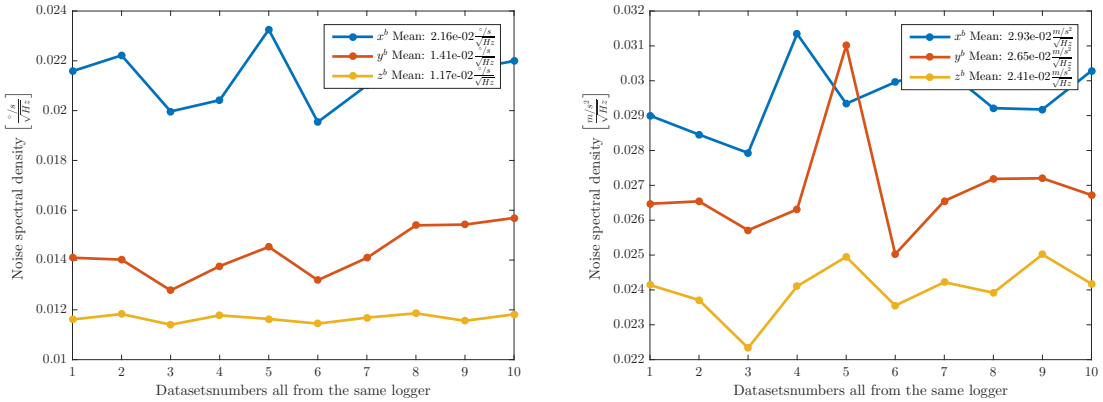
$$\begin{bmatrix} R_{S,x}^n \\ R_{S,y}^n \\ R_{S,z}^n \end{bmatrix} = \boldsymbol{C}_b^n \begin{bmatrix} R_{S,x} \\ R_{S,y} \\ R_{S,z} \end{bmatrix}^b \tag{6.59}$$

Applying the impulse approximation and choosing the system noise distribution matrix as $\boldsymbol{G}_{k-1} = \boldsymbol{I}_3$ leads to the discrete system noise covariance for the velocity-state $\boldsymbol{\delta v}_{nb}^n$ and the alignment-state $\boldsymbol{\delta \psi}_{nb}^n$.

$$Q_{k-1,\psi_{nb}/v} = \begin{bmatrix} R_{S,x}^n & 0 & 0 \\ 0 & R_{S,y}^n & 0 \\ 0 & 0 & R_{S,z}^n \end{bmatrix}^2 \cdot \tau_s \tag{6.60}$$

Gyroscope and Accelerometer Bias

Since the temperature drift of the bias is compensated, the states b_g and b_a are theoretically constant. To account for errors in the calibration algorithm and to ensure the system's stability



(a) Gyroscope Noise

(b) Accelerometer Noise

Figure 6.8: Average noise statistic determined from 10 datasets

some noise is added each iteration resulting in a random walk. The amount for this noise is determined empirically to

- $A^n_{S,xyz}[b_a] \approx 1 \cdot 10^{-8} \frac{(m/s^2)^2}{Hz}$ for the accelerometer bias.
- $A^n_{S,xyz}[b_g] \approx 1 \cdot 10^{-10} \frac{(rad/s)^2}{Hz}$ for the gyroscope bias.

Applying the impulse approximation and choosing the system noise distribution matrix as $\boldsymbol{G}_{k-1} = \boldsymbol{I}_3$ leads to the discrete system noise covariance for the gyroscope bias-state $\delta \boldsymbol{b}_g$ and the accelerometer bias-state $\delta \boldsymbol{b}_a$.

$$Q_{k-1,b_g/b_a} = \begin{bmatrix} A^n_{S,x} & 0 & 0 \\ 0 & A^n_{S,y} & 0 \\ 0 & 0 & A^n_{S,z} \end{bmatrix} \cdot \tau_s \tag{6.61}$$

Lever arm and Body-Rear-Angles

The lever arm $\boldsymbol{l}^b_{br}$ and the body-rear-angles $\boldsymbol{\psi}^b_{br}$ are both considered as random constants. For stability reasons the power spectral densities are

- $A_{S,xyz}[\boldsymbol{l}^b_{br}] = 1 \cdot 10^{-10} \frac{m^2}{Hz}$ for the estimated lever arm
- $A_{S,xyz}[\boldsymbol{\psi}^b_{br}] = 1 \cdot 10^{-10} \frac{rad^2}{Hz}$ for the estimated body-rear-angle

Applying the impulse approximation and choosing the system noise distribution matrix as $\boldsymbol{G}_{k-1} = \boldsymbol{I}_3$ leads to the discrete system noise covariance for the lever arm state $\delta \boldsymbol{l}_b r^b$ and

the body-rear-alignment state $\delta\boldsymbol{\psi}_{br}^{b}$.

$$Q_{k-1,l/\psi_{br}} = \begin{bmatrix} A_{S,x}^{n} & 0 & 0 \\ 0 & A_{S,y}^{n} & 0 \\ 0 & 0 & A_{S,z}^{n} \end{bmatrix} \cdot \tau_s \tag{6.62}$$

Barometer

In section 6.3.2 a Gauss Markov process describes the dynamics of barometer bias. A weather model enabled the estimation of the standard deviation to $\sigma_B = 60m$ with a correlation time of $\tau_{c,B} = 20000s$.

$$\dot{b}_B = \underbrace{-\frac{1}{\tau_{c,B}}}_{F} b_B + \underbrace{\sigma_B \sqrt{\frac{2}{\tau_{c,B}}}}_{G(t)} w_u(t)$$

The system noise covariance matrix is analytically determined without approximations by evaluating (3.44)[3, p.125]. From (3.24) the transition matrix is $\phi(t_{k-1}, t') = e^{-F \cdot t'}$.

$$\begin{aligned} Q_{k-1,B} &= \int_{t_k}^{t_k+\tau_s} \phi(t_{k-1}, t') G(t') A_S G(t'') \phi(t_{k-1}, t'') dt' \\ &= \int_{0}^{\tau_s} \sigma_B^2 \frac{2}{\tau_{c,B}} e^{-\frac{2}{\tau_{c,B}} t'} dt' \\ &= -\sigma_B^2 e^{\frac{2}{\tau_{c,B}} t'} \Big|_0^{\tau_s} = \sigma_B^2 \left(1 - e^{\frac{2}{\tau_{c,B}} \tau_s}\right) \end{aligned} \tag{6.63}$$

6.3.6 Measurement Noise

Measurement noise is the counterpart to system noise in the Kalman gain calculation. It is represented by the measurement error covariance matrix $\boldsymbol{R}$. A smaller value for $\boldsymbol{R}$ will bias the updates in favour of the measurements.

GPS

The signal from the GPS-sensor is internally pre-processed by a Kalman filter. This results in a correlated output which has to be accounted for with a lower update rate. In loosely coupled integration intervals of 10 seconds are common for this kind equipment. An alternative for such a cascaded filter strategy is a single-epoch position receiver without carrier smoothing of the pseudo ranges [9, p.567]. Empirical experiments in this thesis have shown that this update rate yields the best performance in combination with a smoother and innovation filtering.

Additionally the quality of the GPS signal depends on the number of satellites and the geometry indicated by the HDOP-factor. The HDOP-value varies between 0.8-99 and specifies the multiplicative effect on positional measurement precision. From the data sheet [29, p.2] an accuracy of 2.5m is given. Tests with a map as reference have shown, that this precision is only achievable with a high number of satellites (>10), good geometry values (<1.2) and no multipath effects. As a result a linear function (6.9) has been developed returning a standard deviation depending on the mentioned conditions. For the noisier height measurements twice this standard deviation is considered. The NMEA protocol of the Venus-module provides information about the vertical dilution of precision (VDOP) in the GSA-String but was not available in the recorded datasets.

$$\sigma_{GPS}(q, s, \sigma_{max}, \sigma_{min}) = q((\sigma_m in - \sigma_m ax)s) \tag{6.64}$$

where

q is the HDOP.

s is the number of satellite signals received by the GPS sensor

σ_{max} is the standard deviation when 4 satellites are available and $q = 1$ and empirically determined to 15m.

σ_{min} is the standard deviation when 12 satellites are available and $q = 1$ and empirically determined to 1.5m.

Odometry

According to figure 5.12 the signal $\tilde{v}(t)$ is a combination of its true velocity and quantisation noise. As the Kalman filter expects Gaussian white noise $w(t)$ the quantisation noise has to be classified in that way. “A good analogy is that you can fit a square peg into a round hole if you make the hole sufficiently large” [9, p.88]. Its maximum error gives therefore a rough estimate for a Gaussian standard deviation of $\sigma_{O,\tau_s} = \epsilon_O = 1.25\ \frac{m}{s}$ at a bandwidth of $B = 10Hz$. Applying (3.14) leads to a PSD of $A_{S,O} = 0.15625\frac{(m/s)^2}{Hz}$.

Practical experiments have shown that an odometry measurement update rate equal to the system (inertial sensors) rate can result in unstable behaviour. This is justified with the large quantisation steps. One solution to get rid of theses quantisation steps is a filter. By making the assumption that $\tilde{v}(t)$ is ergodic for a short time interval $\tau_c = 0.5\ sec$, time averaging gives the same result as ensemble averaging.

Average filtering for this short time is achieved by the convolution of the measurement signal with respect to a suitable averaging window $h(t)$ of length τ_c. In Matlab the function

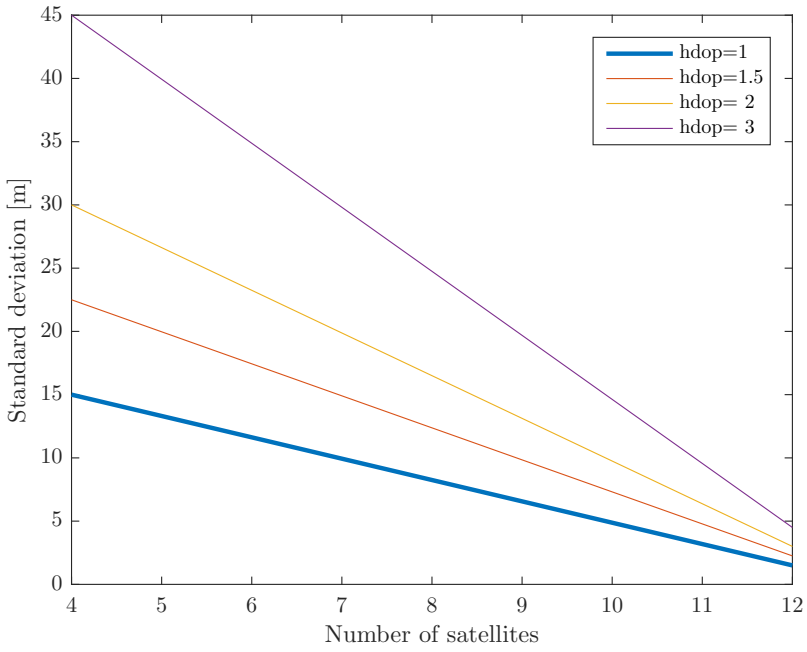


Figure 6.9: GPS standard deviation depending on satellite geometry and number of satellites

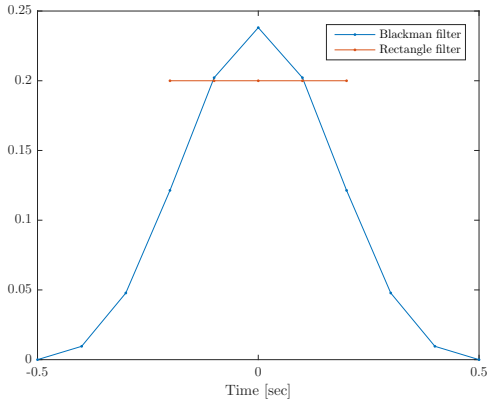
'filtfilt' is used for zero-phase-shift filtering.

$$\text{Continuous: } y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) \cdot h(t-\tau) d\tau \tag{6.65}$$

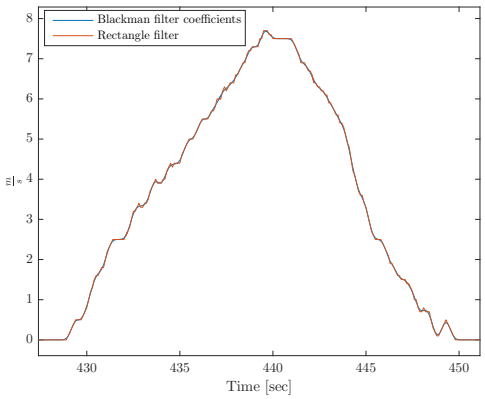
$$\text{Discrete: } y_n = x_n * h_n = \sum_{k=-\infty}^{\infty} x_k h_{n-k} \tag{6.66}$$

For this thesis two windows have been tested in figure 6.10. The Blackman window adds more weight to the actual point in time. That way the window can be increased without being affected of a higher correlation time τ_c. Practically no difference in Kalman filter performance is detectable, thus for an easier analysis the rectangle is used. To fulfil the Kalman filter condition of uncorrelated measurements the update interval must be at least equal to the correlation time τ_c. The noise is also affected by the filter. The convolution of the continuous white noise $w(t)$ with the filter function $h(t)$ results in

$$\begin{aligned} w_{\tau_c}(t) = w(t) * h(t) &= \int_{-\infty}^{\infty} w(\tau) \cdot h(t-\tau) d\tau = \\ &= \int_{-\infty}^{\infty} w(\tau) \cdot \frac{1}{\tau_c} rect(\frac{t-\tau}{\tau_c}) d\tau = \frac{1}{\tau_c} \int_{t-\frac{\tau_c}{2}}^{t+\frac{\tau_c}{2}} w(\tau) d\tau \end{aligned} \tag{6.67}$$



(a) Rectangle and Blackman filter coefficients



(b) Filter results

Figure 6.10: Filtered derivative of odometry signal

Then the new standard deviation is calculated by applying the variance operator (3.4).

$$\begin{aligned}\sigma^2_{O,\tau_c} = E[w_{\tau_c}(t)w_{\tau_c}(t)] &= E[\frac{1}{\tau_s^2}\int_{t-\frac{\tau_c}{2}}^{t+\frac{\tau_c}{2}}\int_{t-\frac{\tau_c}{2}}^{t+\frac{\tau_c}{2}} w(t')w(t'')dt'dt'']\\ &= \frac{1}{\tau_s^2}\int_{t-\frac{\tau_c}{2}}^{t+\frac{\tau_c}{2}}\int_{t-\frac{\tau_c}{2}}^{t+\frac{\tau_c}{2}} E[w(t')w(t'')]dt'dt''\\ &= \frac{1}{\tau_s^2}\int_{t-\frac{\tau_c}{2}}^{t+\frac{\tau_c}{2}}\int_{t-\frac{\tau_c}{2}}^{t+\frac{\tau_c}{2}} A_{S,O}\delta(t'-t'')dt'dt''\\ &= \frac{1}{\tau_s^2}\int_{t-\frac{\tau_c}{2}}^{t+\frac{\tau_c}{2}} A_{S,O}dt'' = \frac{A_{S,O}}{\tau_c}\end{aligned} \tag{6.68}$$

For a time interval $\tau_c = 0.2\ sec$ this leads to a standard deviation of $\sigma_{O,\tau_c} = 0.88\ \frac{m}{s}$.

The amount of noise for the velocity constraints is difficult to estimate. The suspension system allows different velocities along the vertical direction at the rear frame and the IMU. Sideslips will occur in curves exceeding a yaw rate threshold of $3\frac{^\circ}{s}$ [8, p.642]. The linearisation introduces errors and the feedback loop for the estimated states might get unstable with overly great restrictions. Therefore enough additional noise has to be modelled for all three measurement axes. The final amount of this noise is determined empirically to $\sigma_{O,e} = [1.12\ 0.45\ 1.12]\frac{m}{s}$ when $\tau_c = 0.5s$. The measurement update rate is therefore chosen as 2 Hz.

Barometer

Out of the measured pressure the calculated height serves as a measurement for the Kalman filter. Therefore the noise $w_B(t)$ is calculated in units of meter. Figure 6.11 shows the mean values for multiple datasets in stationary phases.

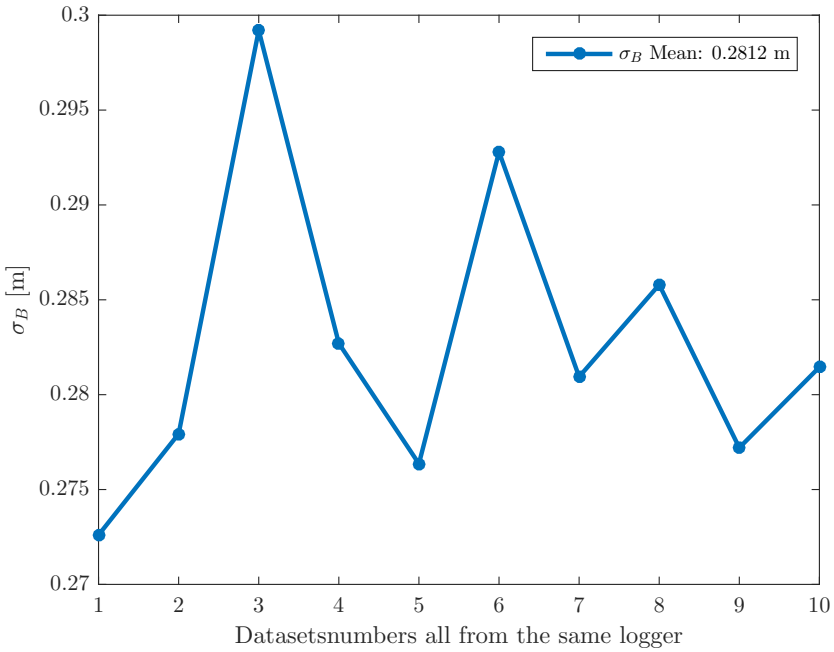


Figure 6.11: Barometer noise in standard deviations at a datarate of 10 Hz

6.3.7 Adaptive Innovation Filtering

From (3.30) the measurement innovation is given as the difference of the measurement $\boldsymbol{z}_k$ and the estimated measurement $\hat{\boldsymbol{z}}_k^-$.

$$\begin{aligned} \boldsymbol{dz}_k &= \boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_k^-) \\ &= \boldsymbol{z}_k - \hat{\boldsymbol{z}}_k^- \end{aligned} \tag{6.69}$$

Based on this information different faults in the measurements or the system state itself may be detected.

Innovation Filtering

The measurements are expected to have a bounded uncertainty defined by the matrix $\boldsymbol{R}$. When a measured data point exceeds this bound by far, it is most likely an outlier which

has to be rejected. This is achieved with a technique called innovation filtering. To get an boundary which is independent of the absolute measured difference a normalization of the innovation has been proposed in [9, p.707]. This is done with the covariance of innovation $\boldsymbol{C}_{\boldsymbol{dz},k}$ in (6.71).

$$\boldsymbol{C}^{-}_{\boldsymbol{dz},k} = \boldsymbol{H}_k \boldsymbol{P}_k \boldsymbol{H}_k + \boldsymbol{R}_k \tag{6.70}$$

$$y^{-}_{k,j} = \frac{dz^{-}_{k,j}}{\sqrt{C^{-}_{dz,k,j,j}}} \qquad j : \text{row of measurement in innovation-vector} \tag{6.71}$$

By choosing a threshold of $y^{max}_{k,j} = 3$ theoretically 99,73 % of valid measurements ($y^{-}_{k,j} \leq 3$) passes the filter. In practice the threshold has to be determined empirically as the measurements do not have an ideal Gaussian distribution [9, p.707].

In this thesis the innovation filter is implemented for the GPS-measurements. A threshold of 2D-position $y^{max}_{k,\lambda,L} = 3$ and for the height $y_{k,h} = 1$ are chosen to get rid of faulty GPS-data through multipath effects. As the height information is a lot more erroneous than the 2D-position data the value has been chosen even smaller. Test have shown that the algorithm tends be more stable when selecting a minimum time $\Delta T_{GPS} = 20\ sec$ between innovation rejections.

Innovation Sequence Monitoring

When the innovation rejection is repeatedly triggered, the fault is not likely to come from the measurement but rather in a biased state estimate. This can be detected with a technique called innovation sequence monitoring [9, p.709].

$$\mu_{kj} = \frac{1}{N} \sum_{i=k+1-N}^{k} y^{-}_{k,j} \tag{6.72}$$

The length of the sequence window N and the threshold for μ_{kj} have to be determined empirically in most cases. When the innovation is Gaussian distributed the standard deviation falls with $\frac{1}{\sqrt{N}}$. Therefore a threshold in terms of standard deviation $\sigma_{dz^{-}_{k,j}}$ would be [9, p.710]

$$T_{\mu_{kj}} = \frac{\sigma_{dz^{-}_{k,j}}}{\sqrt{N}} \tag{6.73}$$

This kind of of system fault detection is implemented for the GPS-measurement. A window-size of $N = 10$ is chosen as a compromise between reaction time of the sequence detection and reliability. The threshold is estimated with (6.73) and empirically optimized to $\sigma_{dz^{-}_{k,\lambda L}} =$

10. Aiming at eliminating the biased state the state uncertainty has to be increased. This is done by multiplying the $\boldsymbol{Q}$ by a factor during all iterations until the next update phase [9, p.711-712]. The factor is tuned empirically to $Q_{boost} = 100$.

6.4 Software Implementation

This section describes how the previously considered ideas are implement as an algorithm in software. Matlab is used as the software environment.

6.4.1 Top level

The Top-level script starts further functions and scripts which are illustrated as modules in figure 6.12. First of all a dataset has to be selected which can be processed for a restricted or entire time interval of the raw data. Subsequently a preparation function converts the dataset into a structure with common element names for later processing steps. The configuration of the Kalman filter is completely excluded in a Matlab script for the purpose of loading different settings. In the configuration script the temperature compensation (5.3.5) is done and the noise levels are evaluated based on all data. The 'interval processing'-box filters and smooths the data of the passed structure. It is described in more detail in following section. Subsequently the processed data can be saved in a new file. For debugging and visual results a Matlab-GUI has been developed illustrated in figure 6.13.

- It shows the coordinates in a plot with selectable Google-map types.
- Special points like bus stations or stops, curves, GPS updates and GPS rejections can be marked up.
- When selecting a coordinate in data cursor mode the box displays additional selectable information (inertial sensor data, all states, uncertainties etc.) for this point in time.
- The results of GPS, filter and smoother may turned on and off separately. If available a reference trajectory can be added.
- Arrows can indicate the direction of travel for every coordinate of the filter results.
- At the top an overview of the configuration is given.

The script 'Additional plots' displays several important navigation quantities over the time interval.

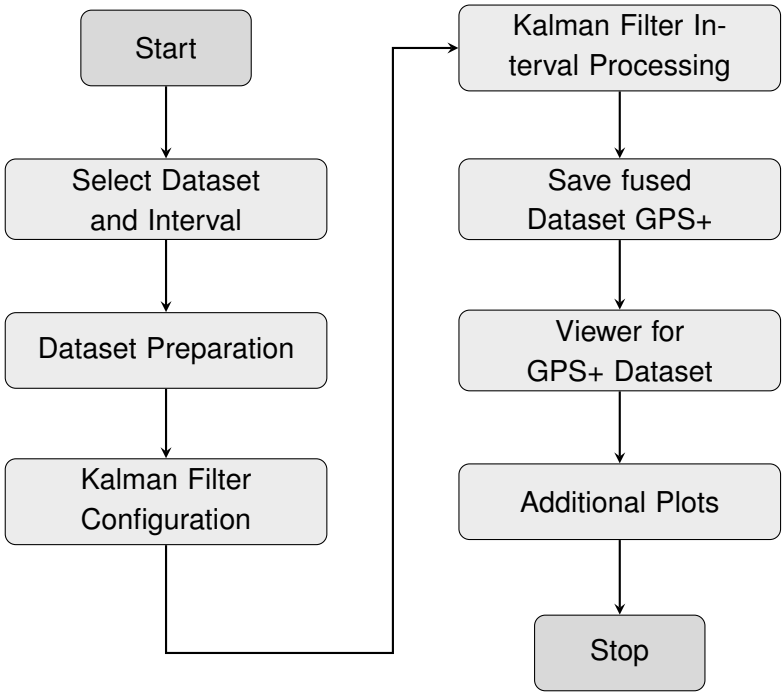


Figure 6.12: Flow chart diagram top level

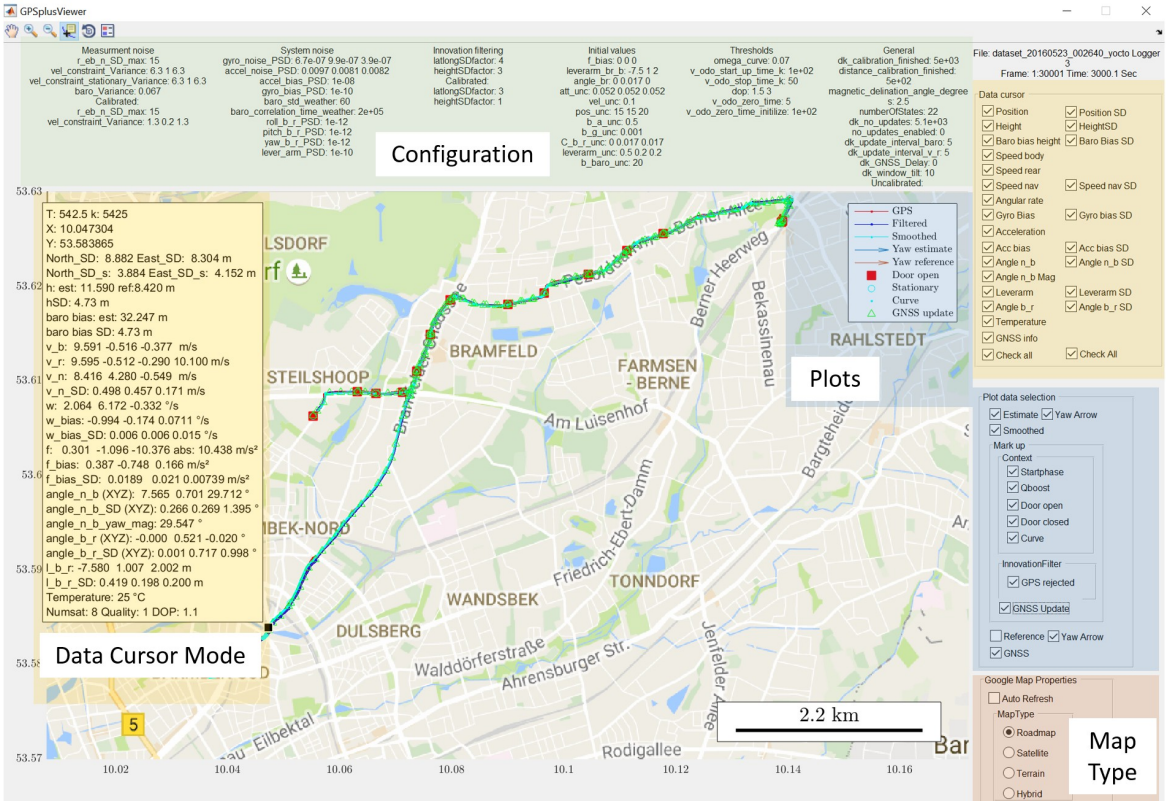


Figure 6.13: GUI in Matlab

6.4.2 Architecture: Interval Processing

Figure 6.14 gives an overview about the phases for fusing the data in the specified interval. Arrows between boxes with a dotted line indicate the way of handling erroneous situations e.g. faults in measurement data or instabilities of the algorithm.

Preparation The arguments of the function are structures for easier manageability. In a loop with thousands of iterations, arrays have a noticeable faster access time than structures (of factor 20), so all relevant information in the structures is allocated into arrays. These arrays have the length of the time vector $\boldsymbol{t}_k$ and their first elements corresponds to the time t_0. The position in later matrices of states is defined with variables, because a descriptive name is easier to comprehend than pure numbers. Also all arrays get preallocated by the length of the interval for processing speed.

Error loop The idea is to catch every error occurring inside the loop. For example in cases of erroneous data leading the filter to instability an error is thrown and the filter reinitialises in the next stationary phase. The old processed data gets smoothed until the last stationary point in time.

Initialisation The velocity derived from the odometry signal is used to search for the next stationary phase. All skipped points in time are set to the initial state determined in this stationary phase. The states and covariances get initialised as proposed in section 6.3.4. The configuration elements of the structure are allocated to variables for processing speed.

Filter loop The loop runs through all points in time starting with the first point after the initialisation phase k_{start}. If no errors occur, the loop finishes at the defined end of the time interval represented by k_{stop}. The next section 6.4.2 describes the filter loop in detail.

Smoother loop The smoother is realised the same way as in the one-dimensional example 3.12. All data points processed by the filter are filtered backwards to improve the estimate of every state.

Save results This module saves the arrays used in the loops into an easier manageable structure which is passed to the top-level script.

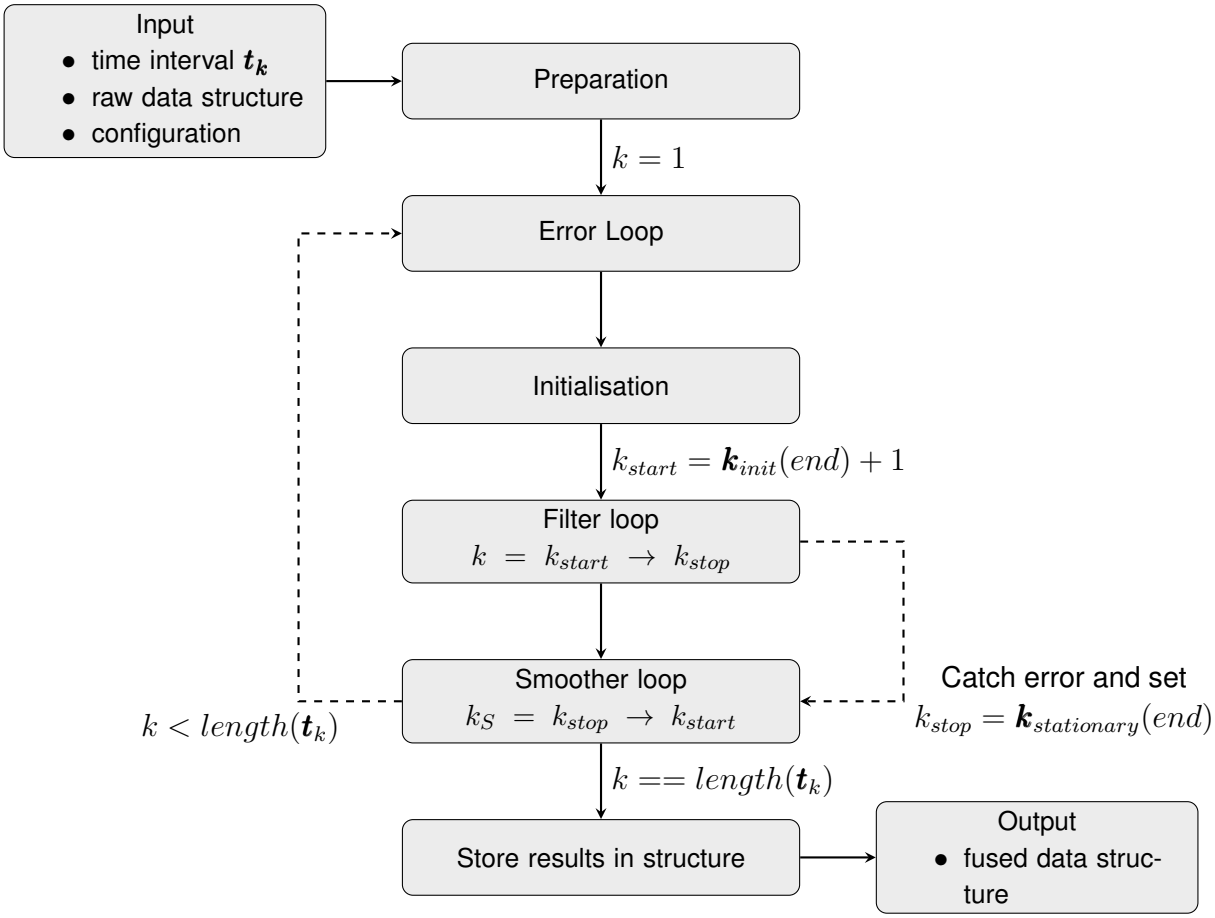


Figure 6.14: Architecture interval processing

Filter Loop

The modules of the filter loop in figure 6.15 are described separately.

Context detection Most of the further steps depend on vehicle's kind of motion. First the odometry information is used to differentiate between moving and stationary phase.

In the movement phase velocity and barometer updates are enabled.

Next it is divided into a curve phase, if a threshold of $4\frac{^\circ}{s}$ is exceeded. For an articulated bus the rigid body assumption is invalid and the odometry information is down weighted by increasing its variance in the measurement matrix. Also the GPS is explicitly disabled to capture high dynamic movements without discontinuities.

When moving straight forward the GPS signal is proved for validity by evaluating the quality reported from the GPS-NMEA string.

During a stationary phase the vehicle does not move and the calculated position must

stay at the same place for this time. Therefore the progress of Kalman filter loop is paused by skipping the illustrated modules in figure 6.15. This leads to constant states and covariances for the stationary phase which results in more realistic smoothed positions in the time just before and after stationary phase. This has been observed empirically. The door signal is evaluated to mark bus stations for later visualisation. In further development the results of the gyroscope or calibrated accelerometer may be analysed for bus tilts.

Update frequency The frequency of updates has been discussed in previous sections. A modulo operation ensures the correct rate.

Navigation solution The strapdown algorithm from 6.2 is processed leading to a independent reference system navigation solution in three-dimensional space. The GPS-signal gets corrected for the lever arm to the place of the INS. The pressure measured by the barometer leads to the height applying (5.6). At least the yaw angle resulting from the magnetometer is calculated for later plots.

Propagation phase In this module the state covariance matrix $\boldsymbol{P}_{k-1}^{-}$ gets propagated by the transition matrix $\boldsymbol{\phi}_k$ and the additional noise from the system noise covariance matrix $\boldsymbol{Q}_k$ calculated with the impulse approximation (3.45).

Measurement phase The measurement matrix $\boldsymbol{H}_k$ and the innovation vector $\boldsymbol{dz}_k^-$ gets extended in stages depending on the update frequency, the results of innovation- and innovation sequence filtering (6.3.7) and the context. All measurements from section 6.3.2 are evaluated and the R-Matrix is selected based on the considerations in section 6.3.6. When the innovation sequence detects a biased state in position the Q_{boost} factor becomes 100 and operates in the next iterations of the propagation phase until the innovation sequence stabilises. If the innovation sequence is triggered more than a defined number $N_{trigger,max}$ the Q-boost method is most likely not able to stabilise the system. Then an error is thrown which gets caught by the error loop and the algorithm reinitialises in the next stationary phase. The threshold is chosen as $N_{trigger,max} = 2 \cdot N$ where N is the window length (Section 6.3.7).

Update phase The Kalman gain is calculated by (3.39) and leads to the updates (3.2.3) for the error states $\boldsymbol{\delta x}_k^-$ and the covariance $\boldsymbol{P}_k^-$. In stationary cases this phase is skipped.

Correction phase The total states $\boldsymbol{x}_k^-$ gets updated by the error states $\boldsymbol{\delta x}_k^-$ in the same way as in the one dimensional example.

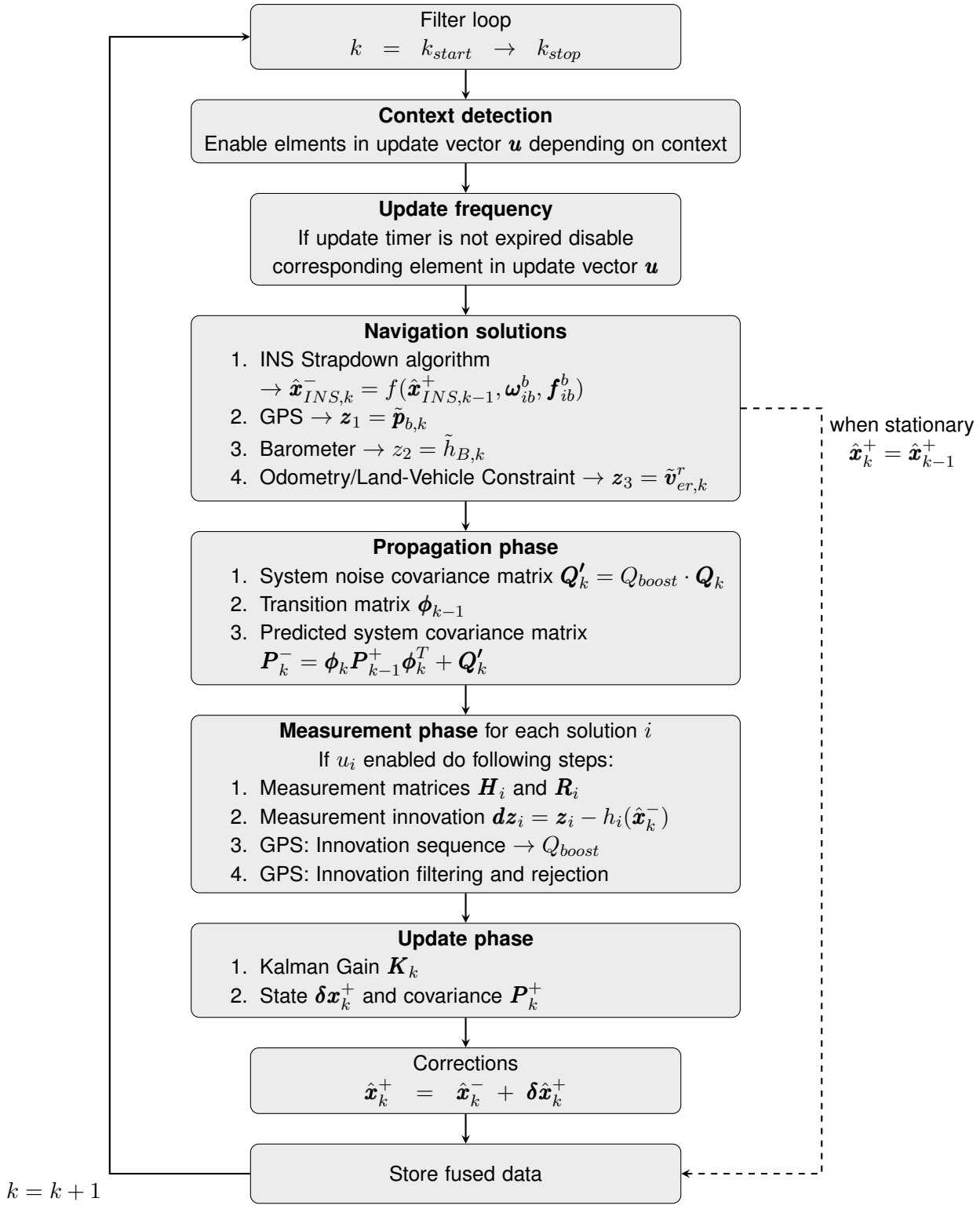


Figure 6.15: Architecture filter loop

6.4.3 Implementation Concept in C

For a real time solution the basic concept of the Matlab algorithm is portable to a C implementation on the Raspberry Pi. The following list sums up important differences in implementation:

1. A RTS-smoother is only available for post processing. In real time application a permanent fixed lag smoother (3.2.4) is an alternative.
2. The temperature compensation algorithm does not work in real time. Instead the zero-angular-rate update to fulfil this task.
3. The noise levels have to be selected based on data in the past.
4. An elliptic magnetometer calibration can be done with data of the past as well.
5. In C the application of constants can improve performance and readability of the code unlike Matlab which does not support a preprocessor functionality.
6. In real time processing the measurement data can directly be processed and does not have to be stored. Therefore the update rate of the inertial sensors should be raised to improve accuracy and reduce the noise of the sensors [14]. The results may be stored to the same frequency of 10 Hz.
7. The Matlab build in functions have to be replaced with corresponding libraries in C.

6.5 Summary

The first section (6.2) provides the equations for using the gyroscope to change the resolving axes of the accelerometer from the body frame to the navigation frame. The INS provides an independent solution for velocity and position and is used as the reference system. Aiding sensors like the GPS, Barometer and odometry in combination with a land-vehicle constraint correct the navigation solution of the INS in an extended Kalman filter algorithm (6.3.2). The noise configuration of the system (6.3.5) and the measurements (6.3.6) are discussed in an analytical way which enables funded empirical optimisation of the individual parameters. For the initialisation of the Kalman filter states several sensors provide solutions with restricted accuracy which lead to the initial covariances(6.3.4). An adaptive innovation filter detects erroneous data in GPS-measurements and the analysis of the innovation sequence enables detection of biased states within the system (6.3.7). Finally an overview for the software implementation in Matlab is given, containing flow charts. As well some ideas for further realisation in C are presented (6.4). This includes a concept for a robust mechanism with error handling (6.4.2).

7 Results

This Chapter shows the advantages of applying the developed Kalman filter algorithm on real data demonstrated in different situations compared to the results from the GPS receiver. Further the states which provide new information are analysed.

7.1 Reference

To give a statement about the accuracy a reference solution for every data point in time is required. A navigation grade system is not available, thus other methods have to be considered.

7.1.1 Height: Database Hamburg

A public database provides height information for the area of Hamburg [31]. The accuracy lies within ± 7 cm and is sufficient for evaluating the estimated height from the algorithm. The database is working with UTM coordinates (Universal Transverse Mercator coordinate system). To get the reference height out of the estimated curvilinear coordinates (latitude,longitude) an efficient and fast algorithm has been developed in Matlab.

7.1.2 Google Maps

For evaluating the accuracy of estimated latitude and longitude coordinates the google map function for Matlab[2] serves as a benchmark. The basic idea to determine the true position is that the bus is on the street. Another constraint can be achieved with the knowledge of the direction of travel and that Germany has right-hand traffic. If there is only one lane left, the position of the INS in the vehicle gives an additional advice where the bus has been. This results in an accuracy up to approximately 1 m to the vertical of the street.

Drawbacks are that the position along the street to a given point in time is difficult to determine. Combining the results of GPS, filter and smoother may lead to an accurate intuitive guess. Therefore some procedural errors can occur. A way to avoid any assumptions about the position along the street is to evaluate only the orthogonal distance from each point of interest to the street. This strategy has not been done here but may be useful for further investigations.

A Matlab function is created where the estimated positions of GPS, filter and smoother are selected with all points spaced in a defined distance in time (22,5 sec, v >5m/s). Each interval is displayed like in figure 7.1 with some coordinates before and behind the position of interest (round dot). Using the data cursor mode of Matlab a reference point is clicked on the map where the INS most likely has been. This procedure is repeated 50 times for a dataset to get some statistical data for a rough comparison between the results of GPS, filter and smoother. Also the estimated accuracy of the covariance can be compared with the real error.

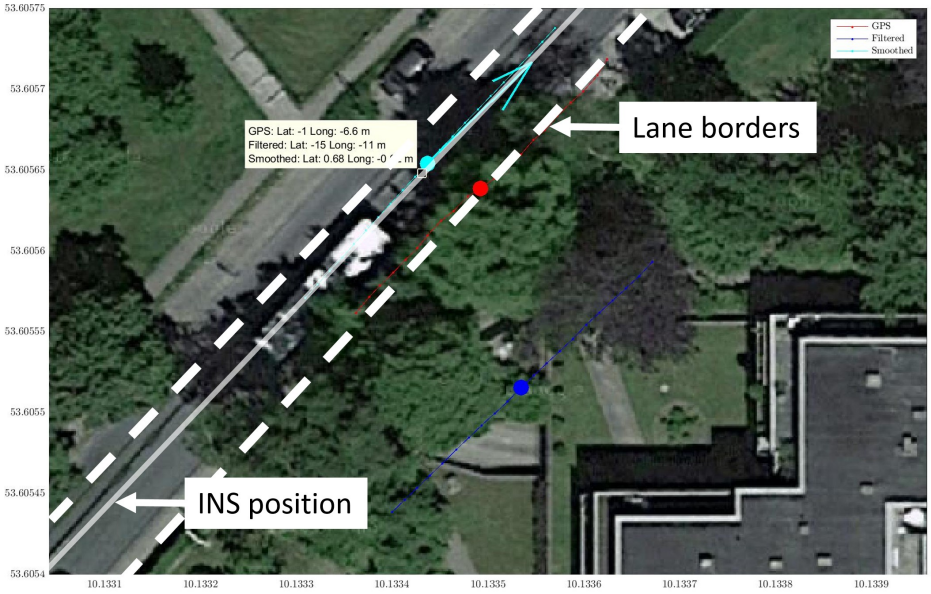


Figure 7.1: Reference creation with Google Maps

7.2 Test with different GPS Qualities

The environment where the logger works changes due to multiple reasons:

- The placement of the components is slightly different when comparing the installation in multiple buses. Especially the antenna might be placed in a way that the signal quality gets restricted. Also deviations in alignment of the IMU in the data logger may lead to various results with different quality.
- Refractions in ionosphere troposphere introduces an error in the pseudo range [9, p.309]
- Receiver measurement error arises from radio frequency interference [9, p.310]

- Small elevation angles of satellites leads to a larger impact of atmospheric delays. Therefore the satellite geometry affects the accuracy in positioning indicated by the DOP factor.
- The operational area crucial is for the influence of multipath effects resulting from buildings. In urban areas they occur much more frequently than in an environment with less high buildings.

The performance of the algorithm is firstly evaluated in an environment with very good GPS signal quality and no disturbances. This is a common approach to show that the developed system is working correctly and no other measurements beside the GPS lead to an unstable behaviour. Secondly an urban environment with medium signal quality and a lot of multipath effects is chosen as this has been the area where the algorithm should lead to the best improvements in position compared to the GPS.

7.2.1 Accurate GPS

The dataset from logger 3 with the highest average number of satellites (8.6) and the lowest average dilution of precision (HDOP=1.1) value is chosen. The vehicle model is a not articulated bus. 18 hours of recorded data has been processed within 1000 sec in Matlab. The results are presented in figure 7.2, which proves that the algorithm works stable over long periods of time. The next sections analyse the accuracy of the filter and smoother.

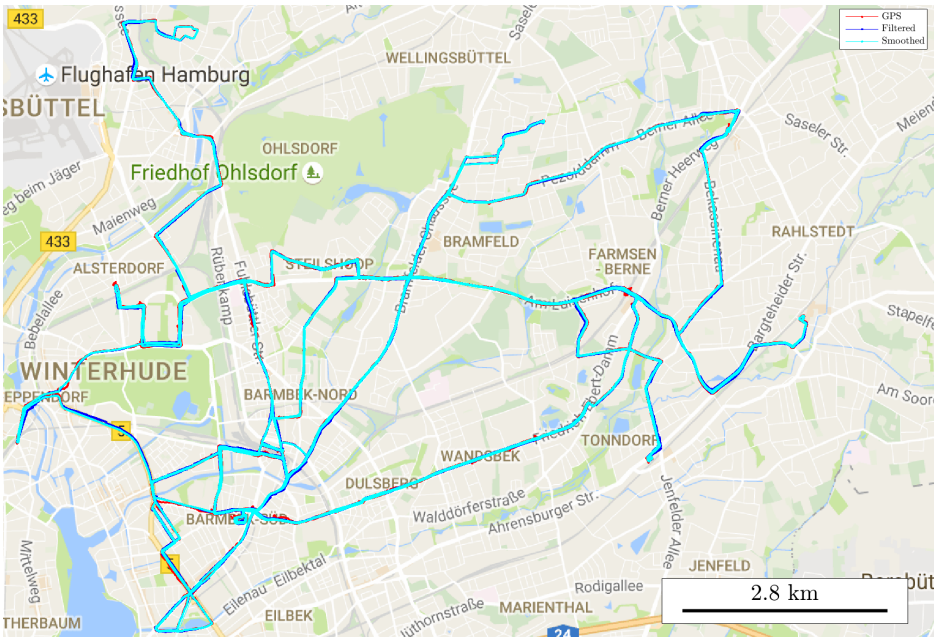


Figure 7.2: Results of processing 18 hours of data

Longitude and Latitude

The technique proposed in 7.1.2 is applied here. From the whole dataset an interval of 5000 sec is divided into multiple data points in time. Overall 50 coordinates are placed on the map which leads to a dataset of true positions. The reference is subtracted from the results of the GPS, filter and smoother. The statistical plots representing the true errors and estimated standard deviations are described in further analysis. The test route in semi-urban area of Hamburg is illustrated in figure 7.3.

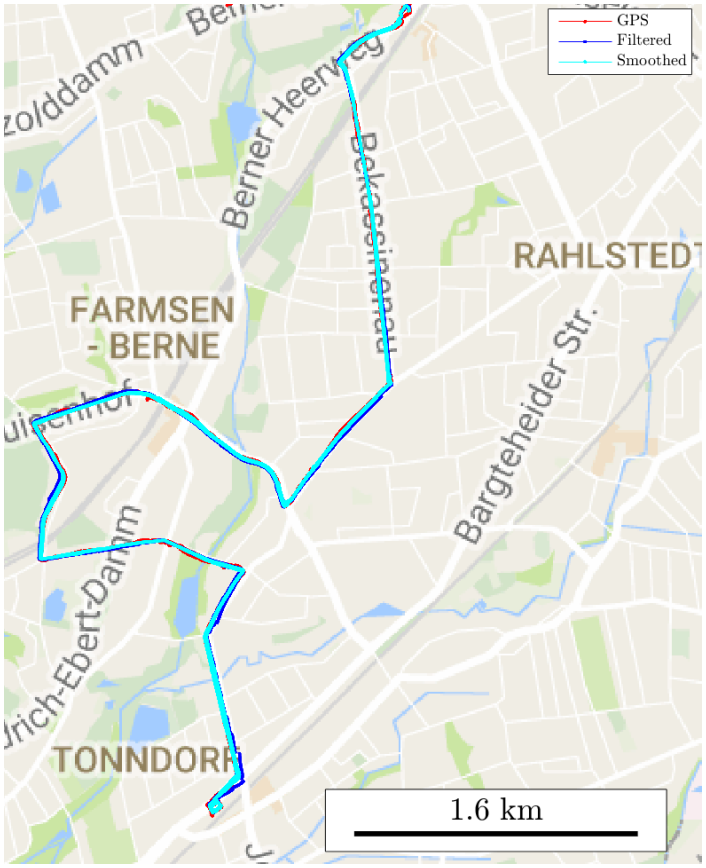


Figure 7.3: Test route for true error and standard deviation calculation

GPS In 6.64 an equation has been developed which gives a rough estimate of the error in the actual GPS-signal by evaluating the HDOP and number of satellites. Figure 7.4 confirms these considerations as the profile fairly matches the true errors. The equation (6.64) may be tuned to get rid of the offset between true error standard deviation and estimated standard deviation. Combining the true standard deviation $\sigma_{e,L,GPS}$ and $\sigma_{e,\lambda,GPS}$ the GPS has an overall accuracy of about five meters. The magnitude of mean errors $\bar{e}_{L,GPS}$ and $\bar{e}_{\lambda,GPS}$ is below 0.4m which indicates that the assumptions of zero mean noise in the measurements for the Kalman filter is right.

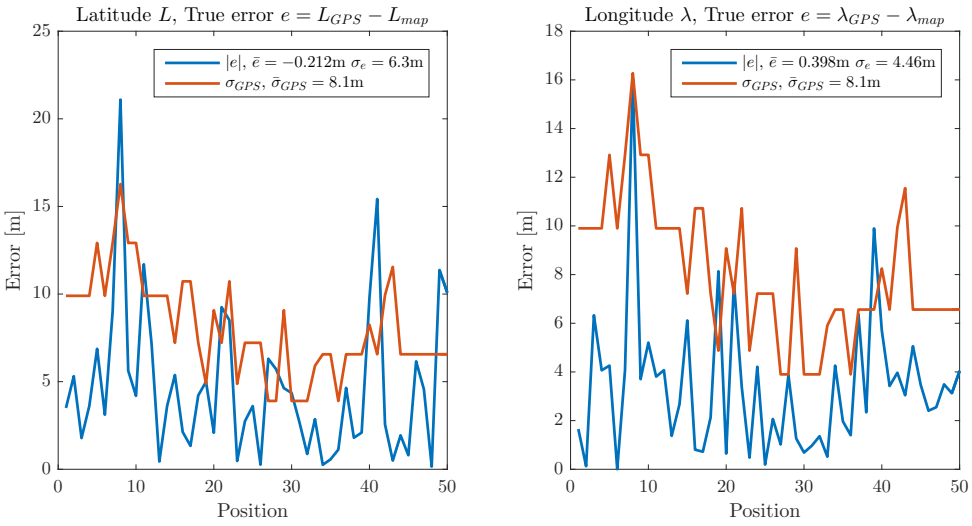


Figure 7.4: GPS true error and standard deviation

Filter The calculated standard deviations σ_{filter} in figure 7.5 seem to be overoptimistic as the averaged true standard deviation is larger. This indicates that the configured parameters for measurement noise and system noise may be too small. Especially the velocity measurement restricts the growth of estimated covariances between the large update interval of the GPS. Thus, this noise variance $\sigma^2_{v_{meas}}$ should be empirically adjusted. From the data in the plots the filter does not improve the accuracy in position by looking at the standard deviations. In return subsequent sections show that especially the dynamics of the vehicle are captured more realistically.

Smoother By processing all filtered data backwards the smoother is able to improve the accuracy in terms of standard deviation by factor 2 compared to the GPS. Also the estimated standard deviations fits to the true errors. An accuracy of 2.5 meters can be achieved with good GPS signal quality.

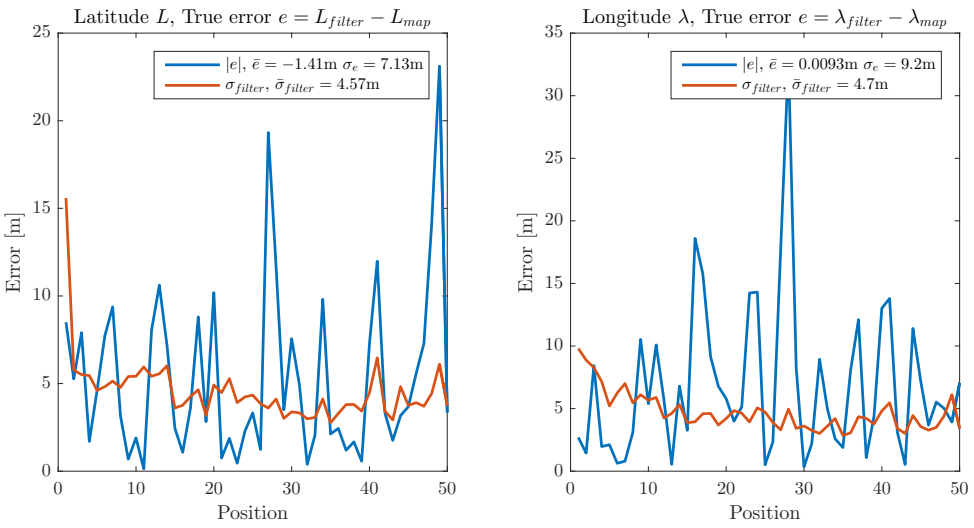


Figure 7.5: Filter true error and standard deviation

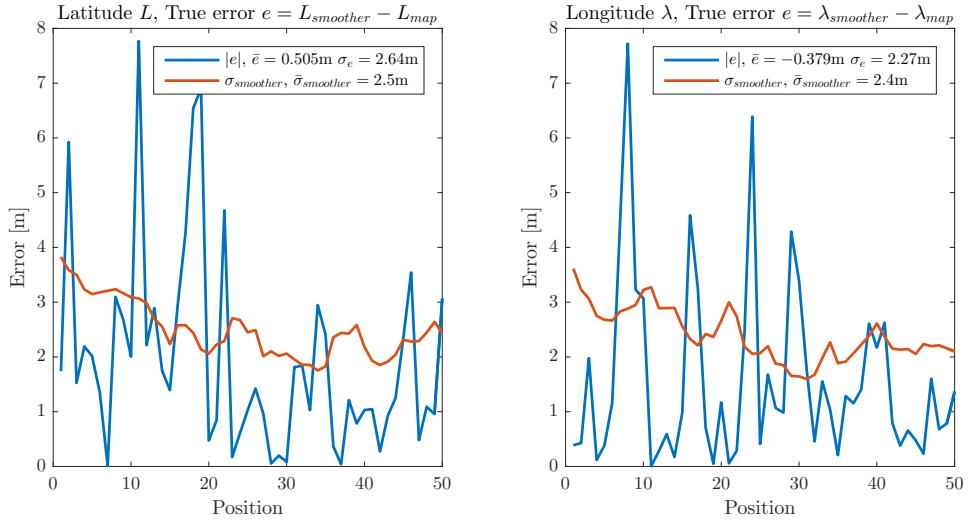
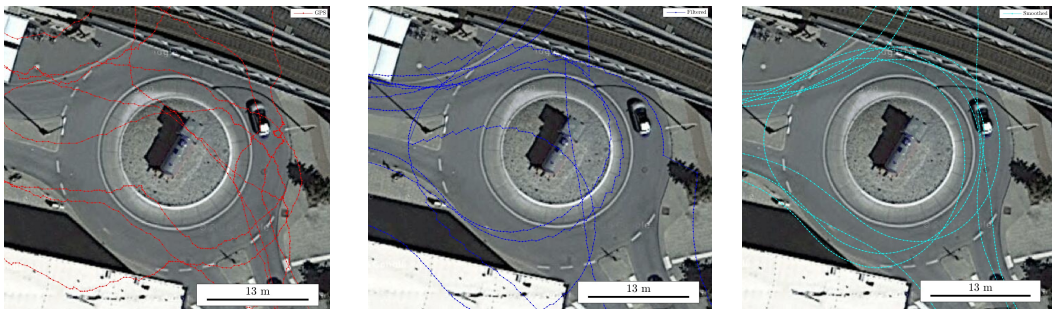


Figure 7.6: Smoother true error and standard deviation

A good choice to compare the results from GPS, filter and smoother is a roundabout. When plotting the entire dataset of 118 hours, the bus has been driven multiple times on this specific intersection of roads. Figure 7.7 illustrates the performance of each method. The GPS is hardly able to show the correct position of the bus around the roundabout. From the statistics in figure 7.5 the filter is not expected to give a better position. But the dynamics are represented more realistically. Finally the smoother outperforms the two other solutions. The dynamics and the accuracy in position has clearly improved.



(a) GPS (b) Filter (c) Smoother

Figure 7.7: Roundabout

Height

GPS In figure 7.8 the height standalone solutions of the GPS and the barometer with height formula (5.6) are illustrated compared to the true height of the database at the calculated smoother-position. Obviously both sensors produce worse results: The GPS has a high standard deviation while the mean of the barometer shows a constant offset. It is also worth noting that the mean error $\bar{e}_{GPS}$ is nearly zero which fulfils one of the Kalman filter noise requirements. The estimated standard deviation $\sigma_{GPS} = 16.5m$ with the formula (6.64) is selected unrealistically high compared to the true deviation $\bar{\sigma}_{GPS} = 9.9m$. Thus, the factor 2 in (6.64) has to be reduced to 1.5 for the height.

Filter and Smoother In figure 7.9 the results of combining this measurements in a Kalman filter are illustrated. Obviously the fusion of the barometer and the absolute height of the GPS works well.The performance improvement with the smoother is not as big as for the longitude and latitude performance. In figure 7.10 both errors of the filter and smoother are compared to the true errors. Overall the estimated standard deviation fits to the true components. The filter yields an true accuracy of 2.2m while the smoother does improve this up to 1.7m.

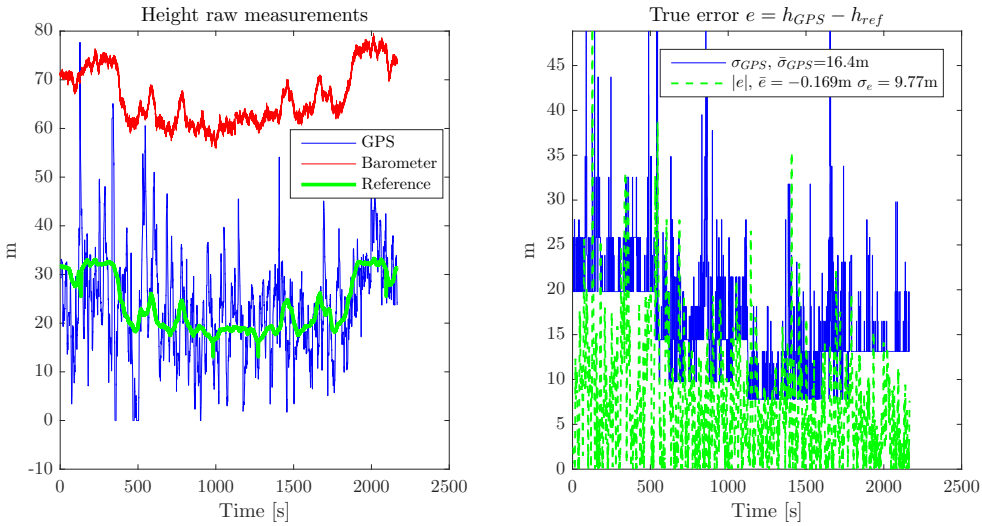


Figure 7.8: Raw measurement data for height

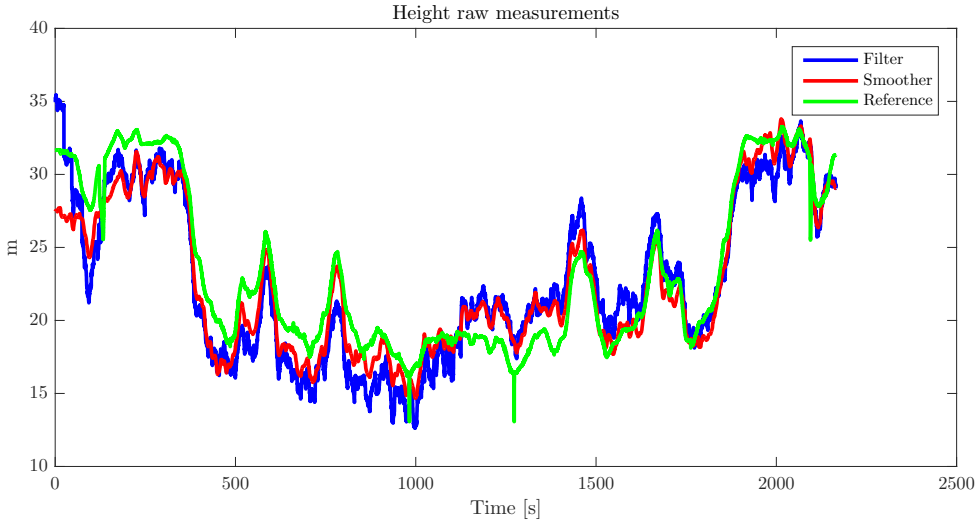


Figure 7.9: Results for height of filter and smoother

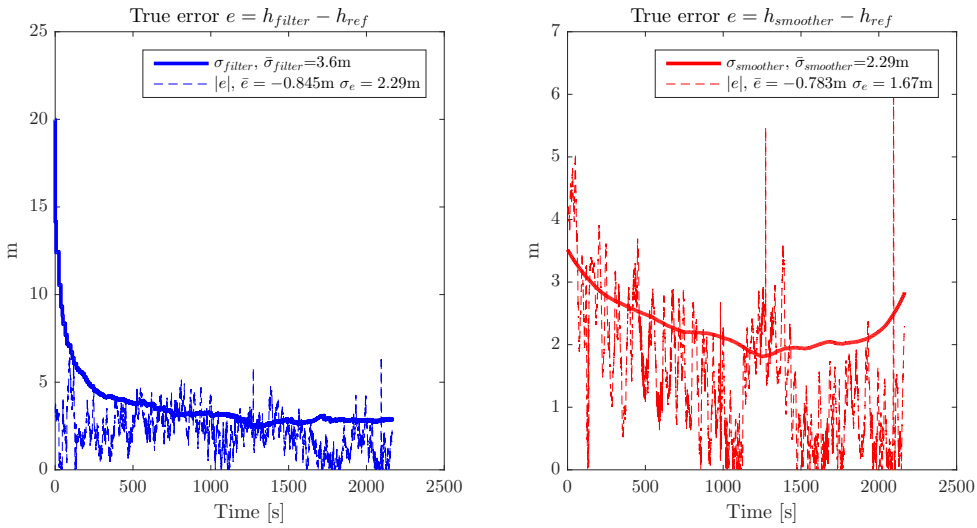


Figure 7.10: Error and standard deviation for filter and smoother

7.2.2 Noisy GPS

A dataset from logger 9 installed in an articulated bus with medium number of satellites (6.95) and low average HDOP (1.9) should challenge the abilities of the Kalman filter algorithm. Figure 7.11 shows the raw GPS signal and the output of the algorithm of processing 9 hours of data. Once more this proves the stability of the algorithm which has triggered its error loop once for this dataset due to erroneous measurements at the start.

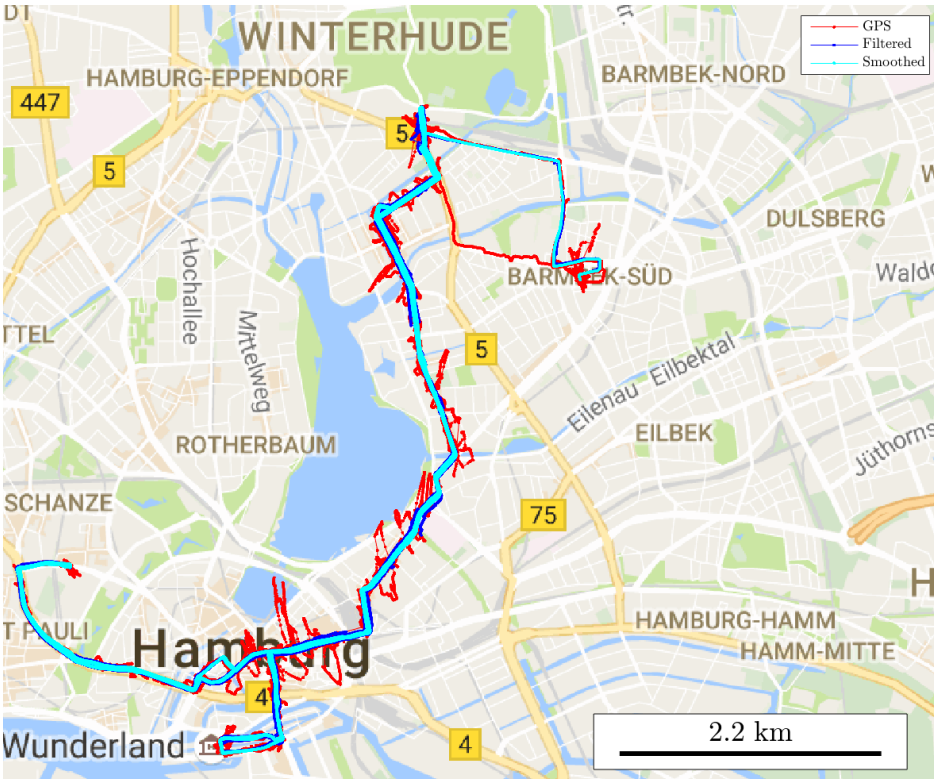


Figure 7.11: Results of processing data recorded in urban areas

Longitude and Latitude

Again the reference technique from 7.1.2 is applied on a dataset from an articulated bus for an interval of 6000 seconds. The true positions are generated for 50 coordinates and which lead to the true errors. The urban test route through the city of Hamburg is illustrated in figure 7.12.

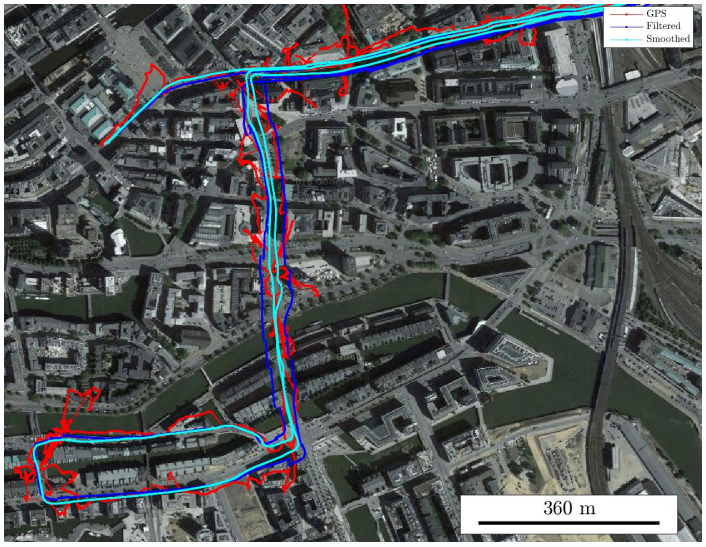


Figure 7.12: Test route for true error and standard deviation calculation

GPS The signal is disturbed by many multipath and bad geometry effects (Average satellites 5.46, average HDOP 2.5). Therefore equation 6.64 calculates the overall high standard deviations σ_{GPS} as illustrated in figure 7.13. Except from some big outliers in the beginning the profile of the calculated deviation matches the true error. By looking at the standard deviation of the true error $\sigma_{e,L,GPS}$ and $\sigma_{e,\lambda,GPS}$ a total accuracy of approximately 20m is achieved.

Filter In contrast to the undisturbed environment the filter improves the standard deviation to $\sigma_e = 11m$ in figure 7.14. This is nearly a factor 2 compared to the GPS. The true error is also more similar to the estimated standard deviation of $\sigma_{filter} = 8 - 9m$, which indicates that the configured parameters are working more realistic in urban area. The mean of the error tends to have a larger magnitude which is the result of weak corrections from the disturbed and down weighted GPS measurements. Thus the filter needs several updates to correct a false course. A smoother which doubles the update could have a lot of potential for improvements.

Smoother As implied the smoother fixes those periods where the filter is corrected several times. This is indicated by the mean error value $\bar{e}$ that is close to zero in figure 7.15. A true accuracy of $\sigma_e = 4.1m$ is achieved and matches perfectly with the predicted standard deviation of about $\bar{\sigma}_{smoother} = 4.4m$. Compared to the GPS this is an improvement of factor 5 in accuracy!

The previous statistics have described the accuracy with numbers. In order to provide a visual impression about the reliability some exemplary areas are selected out of processing the whole dataset. Figure 7.16 illustrates the advantages of combining the information of all

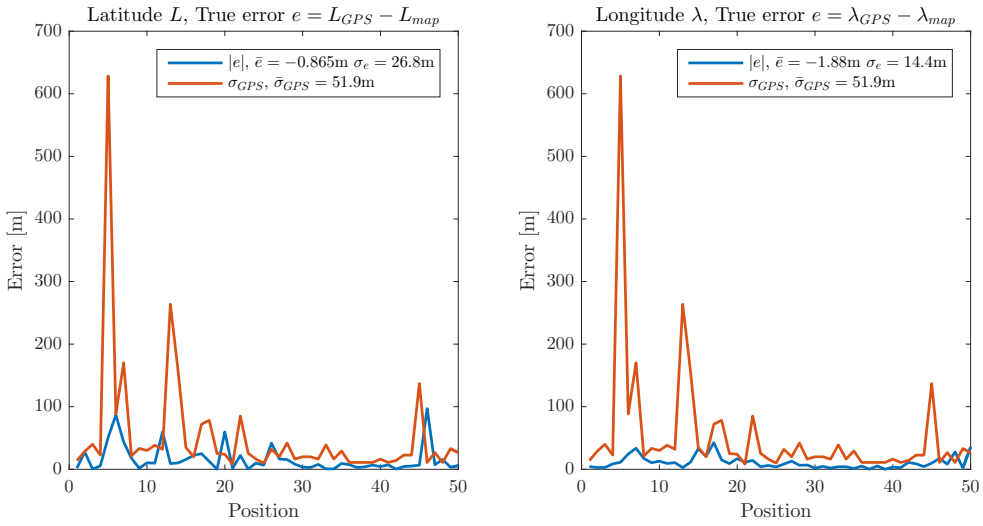


Figure 7.13: GPS true error and standard deviation

sensors. Nearly all estimated and smoothed positions are correctly on the road, although the GPS only provides an unreliable solution. The next graphic 7.17 shows that the dynamics in curves are also captured in urban canyons where the GPS is totally disturbed by multipath effects. The initialisation phase in figure 7.18 originating from the small dataset is remarkable as well. The biased filter has some serious deviations from the true trajectory. The smoother is able to correct those errors nearly completely.

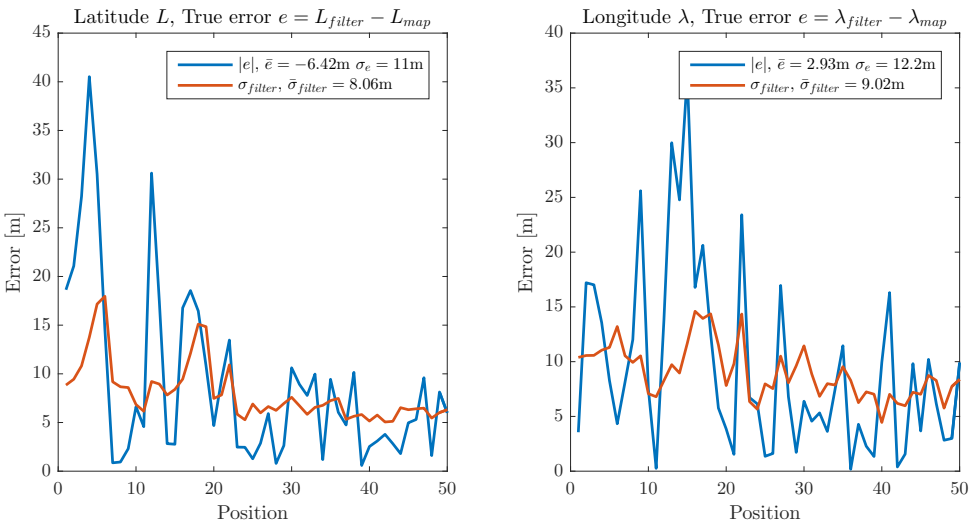


Figure 7.14: Filter true error and standard deviation

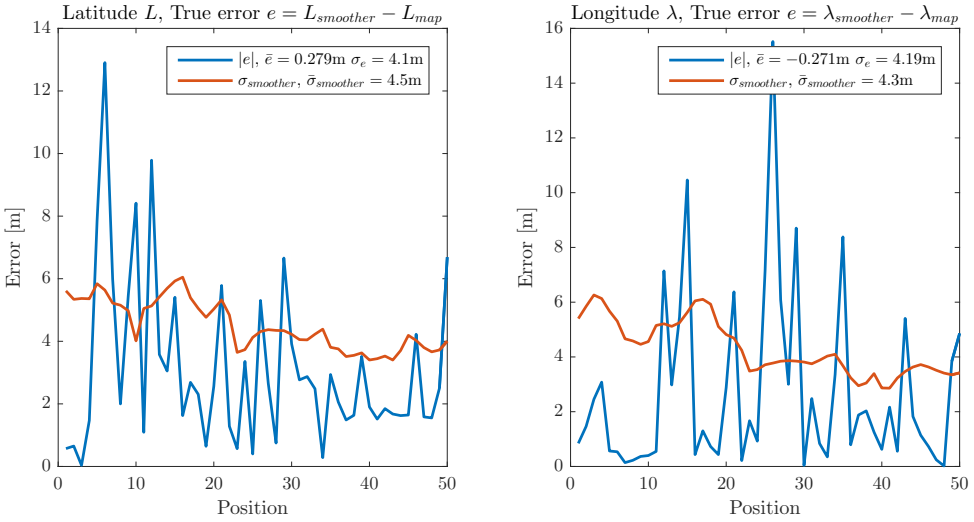


Figure 7.15: Smoother true error and standard deviation

Figure 7.16: High stability on straight roads

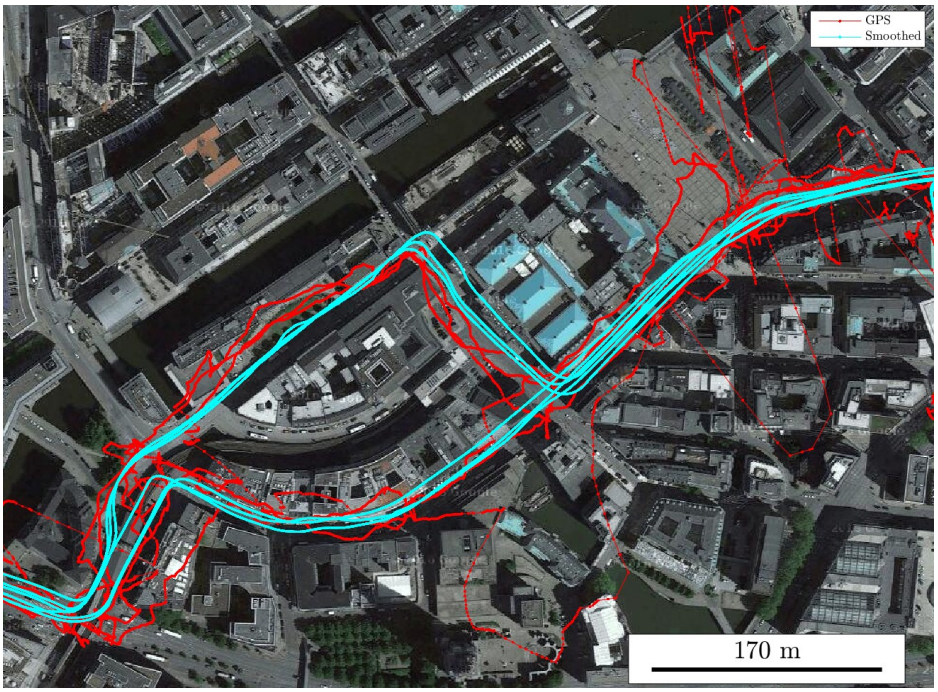


Figure 7.17: Dynamics captured in curves

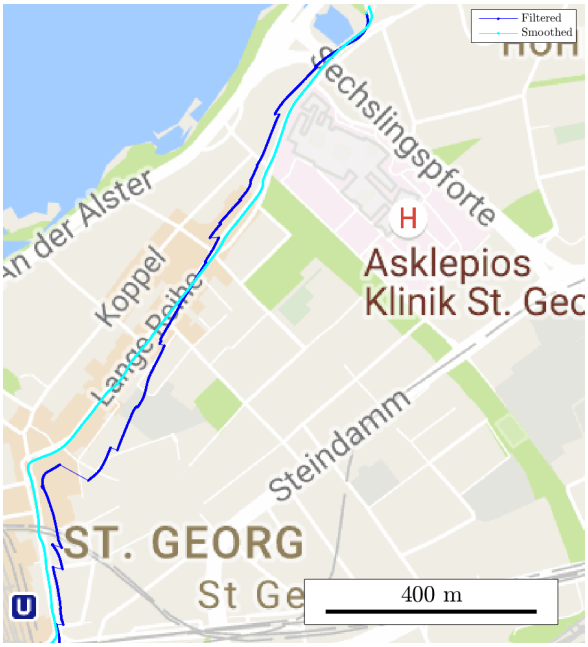


Figure 7.18: Initialisation phase

Height

GPS In figure 7.19 the raw measurements are visualised. The calculated standard deviation $\sigma_{GPS} = 87.5m$ of the GPS is much higher than the true standard deviation of $\sigma_e = 23.7m$. This is justified by with the high offset $\bar{e} = 17m$. Obviously the GPS receiver does not provide negative height values which leads to a single sided noise near sea level. Of course this might be critical for the Kalman filter which expects zero-mean noise. The results of the filter and smoother show the consequences. One way of a correction might be to add artificial negative noise for those time at zero height.

Filter and Smoother The offset of the GPS-measurements is directly transported into the results of the filter and the smoother in figure 7.20. The plots in 7.21 have an offset of $\bar{e}_{filter} = 13m$ and $\bar{e}_{smoother} = 10.5m$ which correlates with the 16m of the GPS. The estimated standard deviations $\bar{\sigma}_{filter} = 4m$ and $\bar{\sigma}_{smoother} = 2m$ correspond with the true deviation of $\sigma_{e,filter} = 3.9m$ and $\sigma_{e,smoother} = 1.8m$. That confirms that the Kalman filter predicts correctly and only the artificial restriction of the GPS receiver leads to the systematic error.

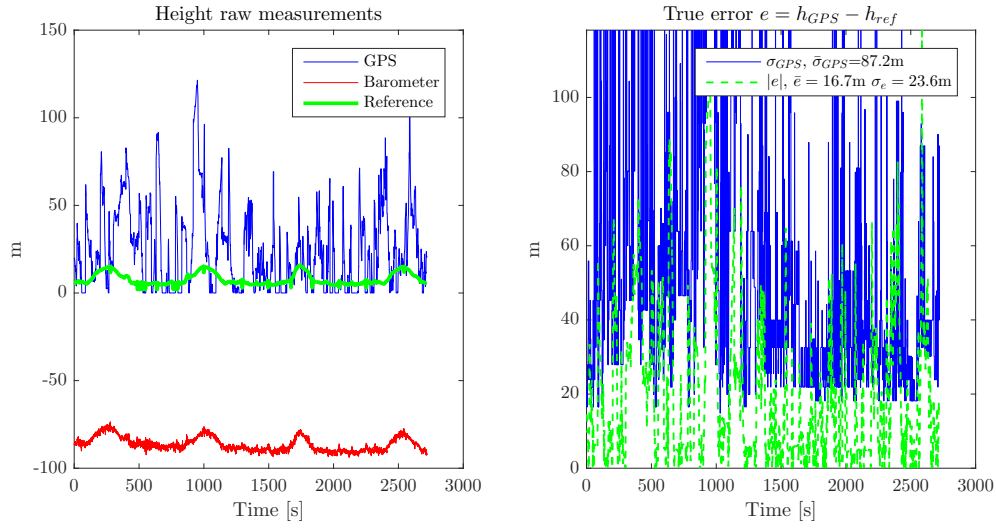


Figure 7.19: Raw measurement data for height

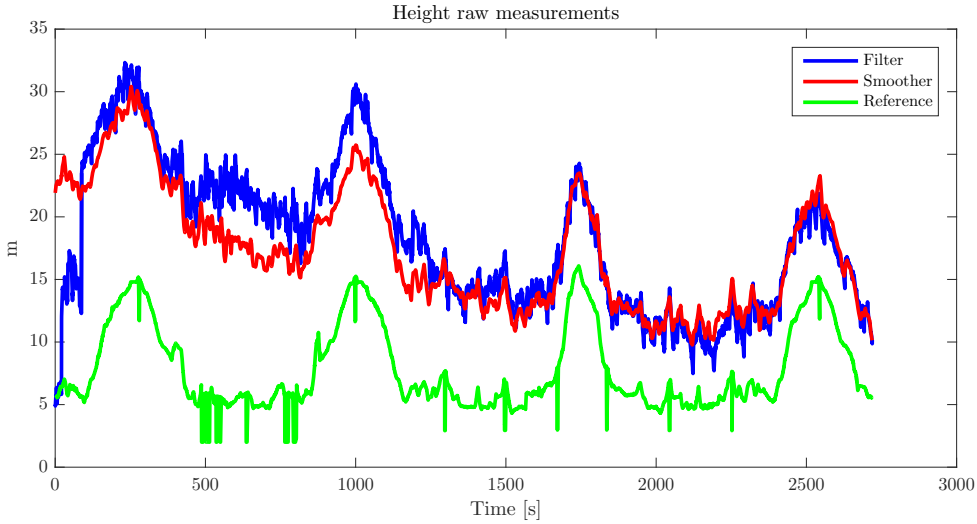


Figure 7.20: Error and standard deviation for filter and smoother

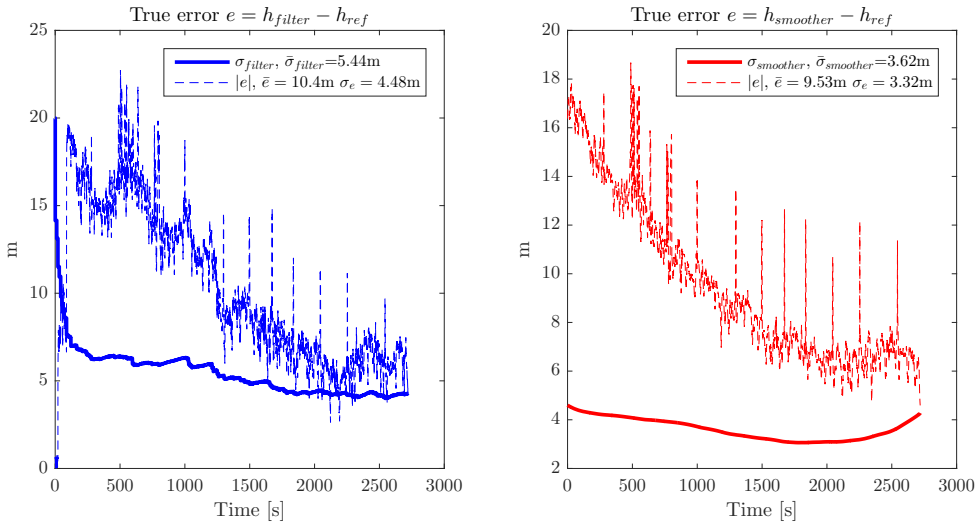
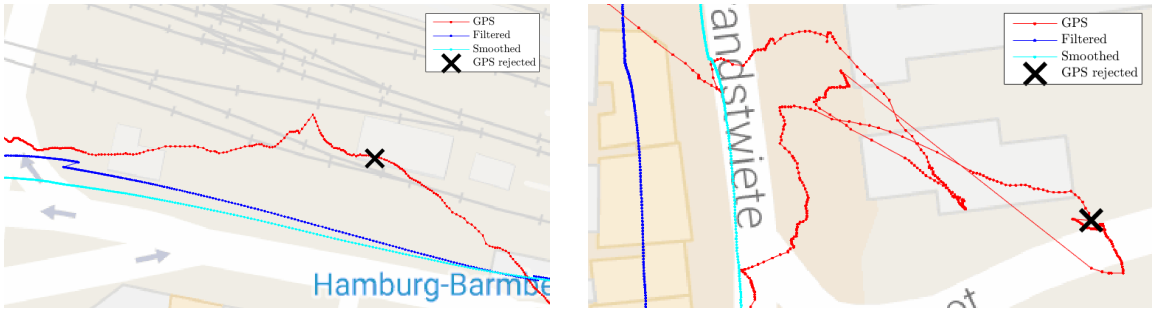


Figure 7.21: Results for height of filter and smoother

7.3 Robust Error Detection and Handling

The innovation filtering technique is working correctly most of the time. Figure 7.22 illustrates those situations where the error in the GPS signal is large compared to the accuracy given by the covariance of the kalman filter. This enables an adaptive threshold configuration for areas with good and areas with noisy GPS quality.



(a) Area with accurate GPS quality (b) Area with noisy GPS quality

Figure 7.22: Successful innovating filtering

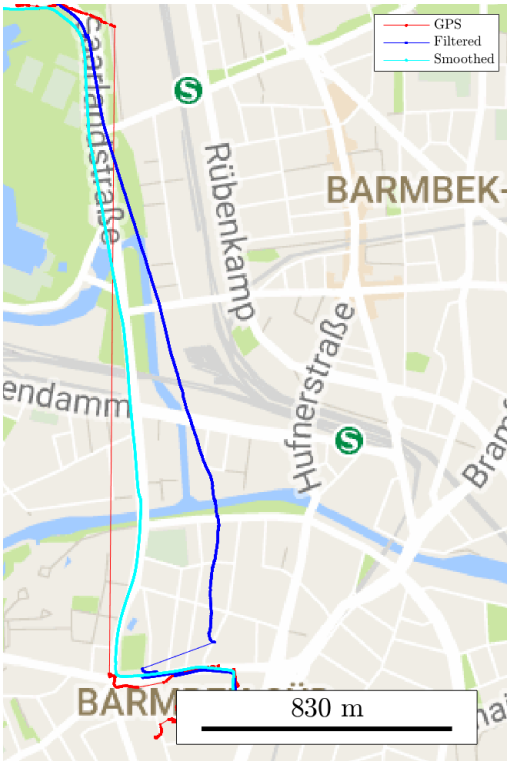
However, this technique has also some disadvantages. Figure 7.23 highlights a situation where the rejection is successfully in the first place (cros at the top),but short time later the GPS is being rejected incorrectly. This is the result of an optimistic tuning which constraints the growth of the error covariance matrix so that the position measurement is not accepted. The tuning of noise parameters and thresholds for innovation filtering is therefore a sensitive balancing of false alarm rate and correct rejections [9]. Smaller noise parameters or thresholds gives more weight to the reference system and measurements are rejected more likely. When increasing the noise or thresholds, erroneous measurements passes more frequently.

Figure 7.23: Correct and false innovation filtering

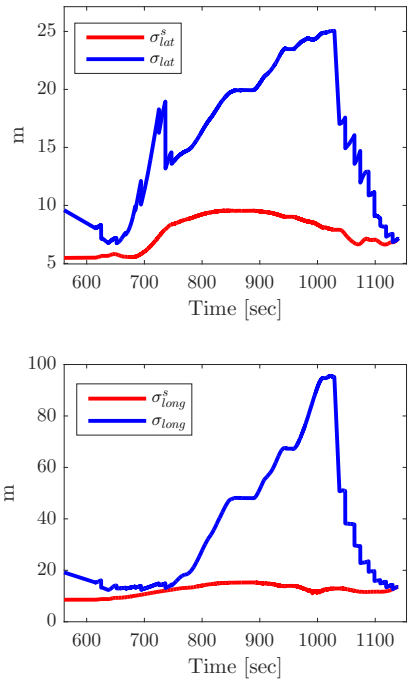
7.4 Smoothing GPS Outages

Sometimes the GPS is not able to provide any solution when moving through tunnels or due to hardware issues. This results in outages of the GPS position although the vehicle changes its position. Then the relative change in position is detected by the INS and the velocity signal. The barometer still provides an absolute solution for the height.

Figure 7.24a highlights the case of a hardware issue. As soon as the outage begins, the solution of the filter drifts away until the GPS gets its next position fix and the measurement update corrects the error. This is a remarkable example for the ability of the smoother to improve the position solution of the past. Regarding the outage-period the estimated accuracy in figure 7.24b grows constantly for the filter and shrinks at the GPS-update. The smoother is able to use this information to correct those filter estimates correspondingly to their covariances. Figure 7.25 illustrates the drive through a tunnel where the GPS does not provide a



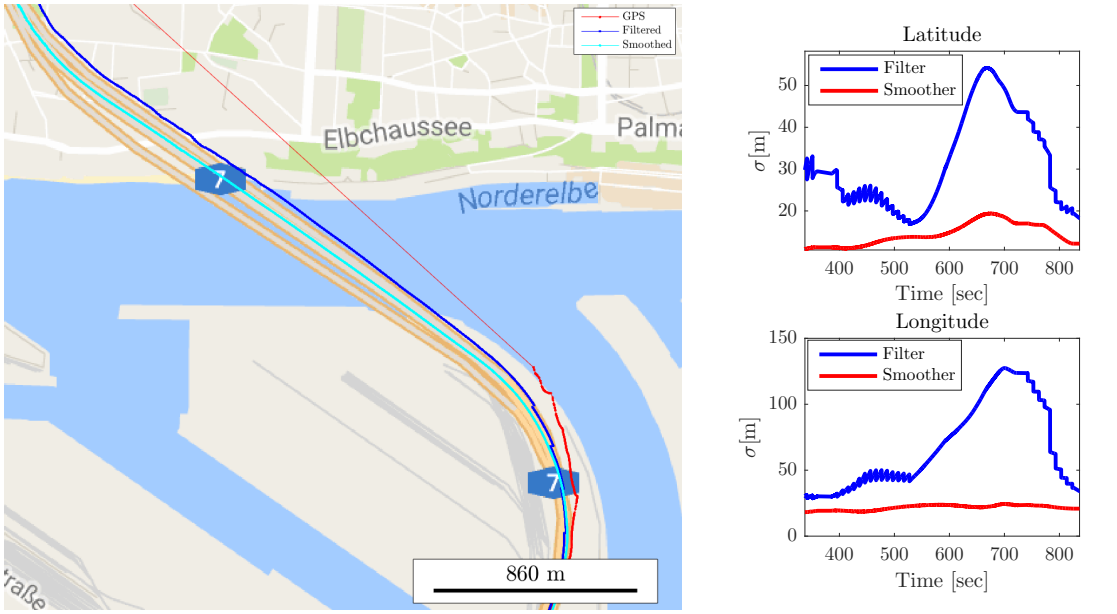
(a) GPS,filter and smoother results



(b) Standard deviation latitude and longitude

Figure 7.24: Outage due to hardware issue

solution.



(a) GPS, filter and smoother results

(b) Estimated standard deviation

Figure 7.25: Outage in a tunnel

7.5 Further Analysis of States

7.5.1 Biases

The gyroscope static bias is observed in the stationary initialisation phase. The static accelerometer bias is estimated with help of the aiding measurements which are related through the transition matrix and finally the Kalman gain. Figure 7.26 illustrates the change of the biases with respect to time for the whole dataset of 7.2. The offset of the accelerometer drifts slowly with growing temperature which indicates that the temperature compensation algorithm (5.7) was not very effective. On the other hand the gyroscope offset is totally stable thanks to the compensation.

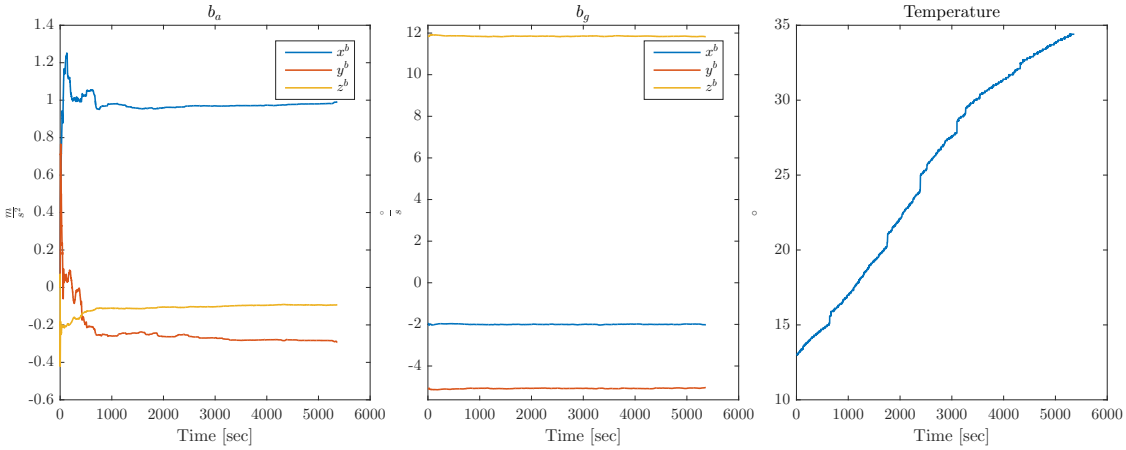


Figure 7.26: Temperature corrected bias progression

7.5.2 Alignment

The accuracy of alignment between the Earth and the vehicle is crucial for the performance in positioning as the angle $\boldsymbol{\psi}^n_{nb}$ determines the direction of acceleration resolved in navigation frame in the strap down algorithm (Section 6.2). In further analysis the results for alignment during the test route of section 7.2.1 are used.

For land-vehicles the yaw angle ψ_{nb} has the highest rate of change and is sufficient for 2D navigation. The accuracy given by the filter converges at $\sigma_{\psi^f_{nb}} = 1°$ and the smoother doubles the performance to $\sigma_{\psi^s_{nb}} = 0.5°$ illustrated in figure 7.27. The smoothed results can be used as the reference for evaluating the accuracy of the magnetometer in section 7.5.2.

The change in pitch-angle θ_{nb} is caused by the slope of street. The estimated body-rear pitch angle θ_{br} enables an absolute calculation of the street's slope. From figure 7.28 the standard deviation of the filter lies within $\sigma^f_{\theta_{nb}} = 0.25°$ and the smoother improves this up to $\sigma^s_{\theta_{nb}} = 0.12°$.

The roll angle in figure 7.29 is about $\bar{\phi}_{nb} \approx 9°$ on average. For the purpose of drainage streets (frame indicated by subscript s) have a cross slope of $\phi_{ns} = 2.5°$ at least [22, p.226]. Thus, ϕ_{nb} represents a combination of this cross-slope and the misalignment of the IMU within the bus. Changes in the roll angle originate from tilts and various cross slopes of the roads. The accuracy yields the same as for the pitch angle estimation. The impact of unrecognised tilts during stationary phases is not further investigated and is part of further research. A combination of bias compensated accelerometer and gyroscope may be used to record alignment in those phases.

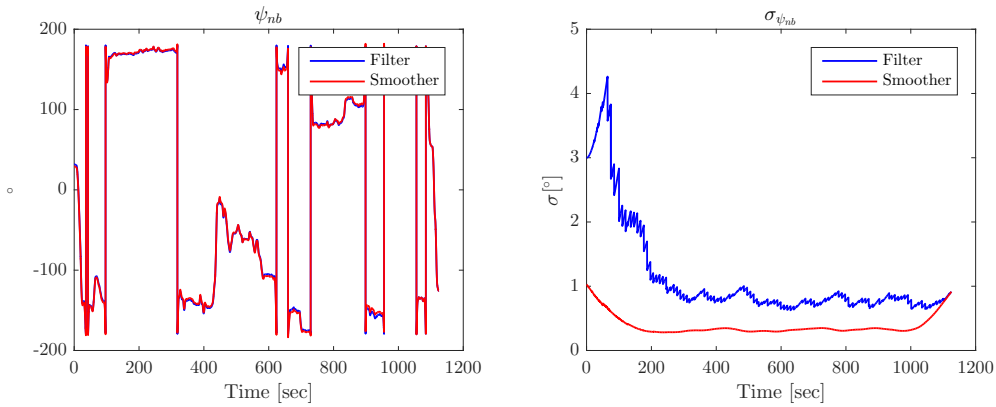


Figure 7.27: Yaw angle

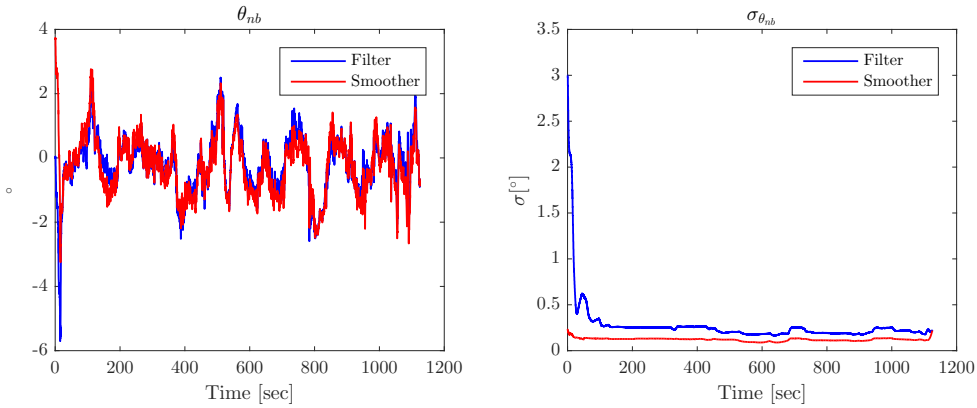


Figure 7.28: Pitch angle

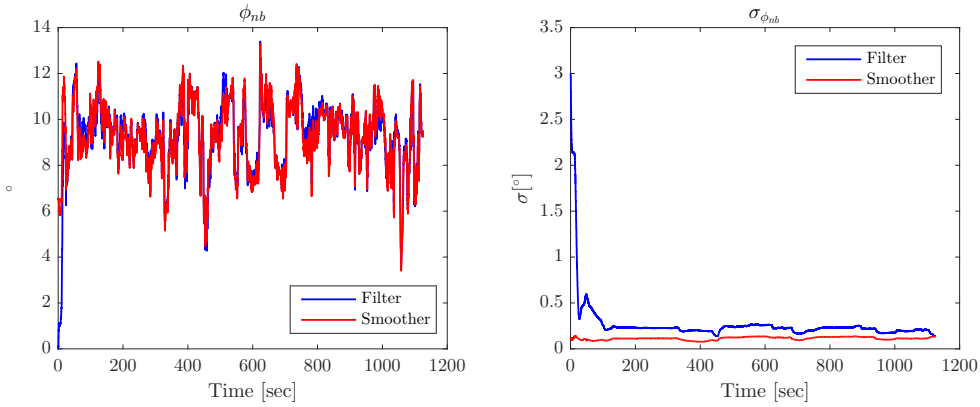


Figure 7.29: Roll angle

Magnetometer Performance

In section 6.3.4 a 2D calibration algorithm has been developed. Now its ability to eliminate the deterministic bias and scale factor errors of the magnetometer is tested. Figure 7.30 visualises the error statistics for the magnetometer yaw angle ψ_{nb}^{mag} calculated with (5.5) by using the accurate smoothed angle ψ_{nb}^{s} as reference. Outliers with an angle above $15°$ are rejected. The mean error $\bar{e}_{mag} = -0.198°$ lies within the uncertainty of the reference which confirms an excellent calibration of the bias. As the diagram is symmetric also the soft iron errors are eliminated successfully. Tests have shown that the noise of the magnetometer leads to a standard deviation of $\sigma_{\psi_{mag}}^{noise} \approx 1°$. Therefore the local anomalies produce a standard deviation of

$$\sigma_{\psi_{mag}}^{anomalies} = \sigma_{e_{mag}} - \sigma_{\psi_{mag}}^{noise} \approx 4° \tag{7.1}$$

The initial uncertainty for the Kalman filter has been set to $3°$ in 6.3.4 which should be increased to $4°$ to account for these anomalies while the noise gets averaged out. The rejected outliers can lead to divergence in initialisation which is caught by the error loop of the algorithm. The integration of the magnetometer as a measurement has been tested, but the result were insufficient due to the local disturbances.

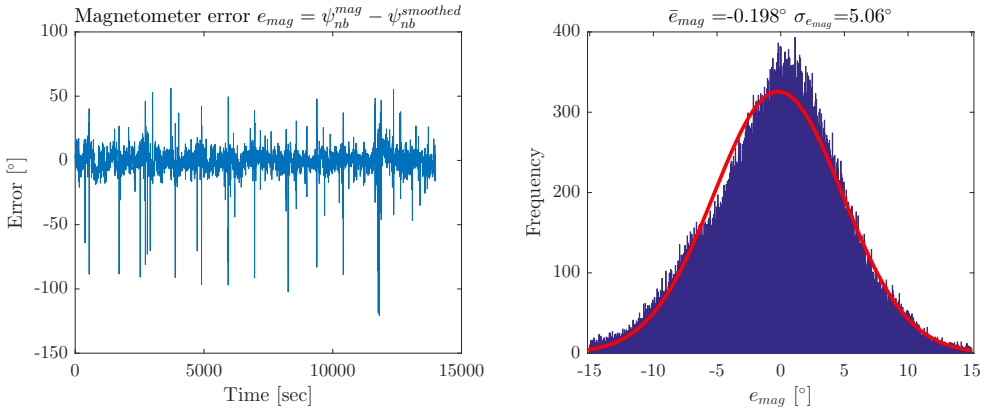


Figure 7.30: Statistics calibrated magnetometer

7.5.3 Lever arm and Body-Rear-Frame Alignment

For testing the ability of the Kalman filter to observe the states of lever arm the initial values are set to $l_{br}^{b} = [0\ 0\ 0]m$. The bus is not articulated and from 5.2 the distance from the place of the IMU to the rear axis is approximately $l_{br}^{b} \approx [8\ 1\ 2]m$. Therefore the initial uncertainty is increased to $\sigma_{l_{br}^{b}} = [2\ 0.5\ 0.5]m$. Figure 7.31 shows the largest part of the lever arm along the bus which converges around $\hat{l}_{br,x}^{b} = 7.9m$ which is in the expected range. The other

parts of lever arms estimate to $\hat{y}^b = 1.15m$ and $\hat{z}^b = 2.1m$ which is also in the expected range visualised in figure 7.32 and 7.33. As the constants do not change much the initial value can be set to the converged results dependent of the vehicle model.

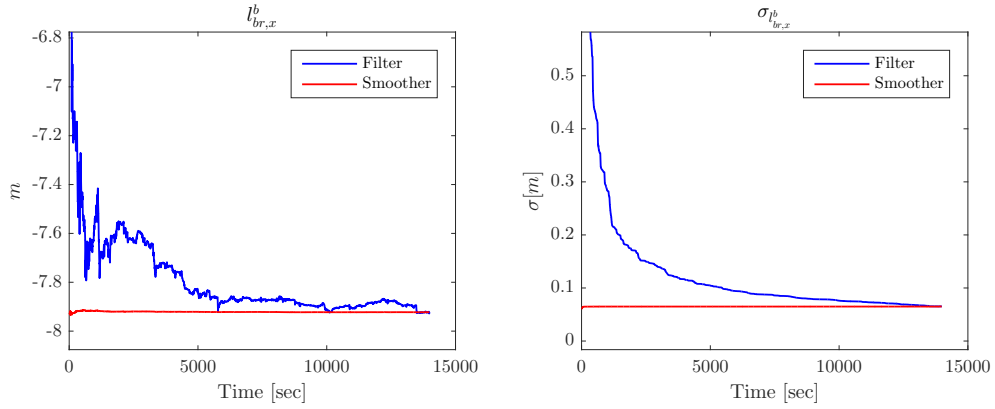


Figure 7.31: Lever arm x^b-axis

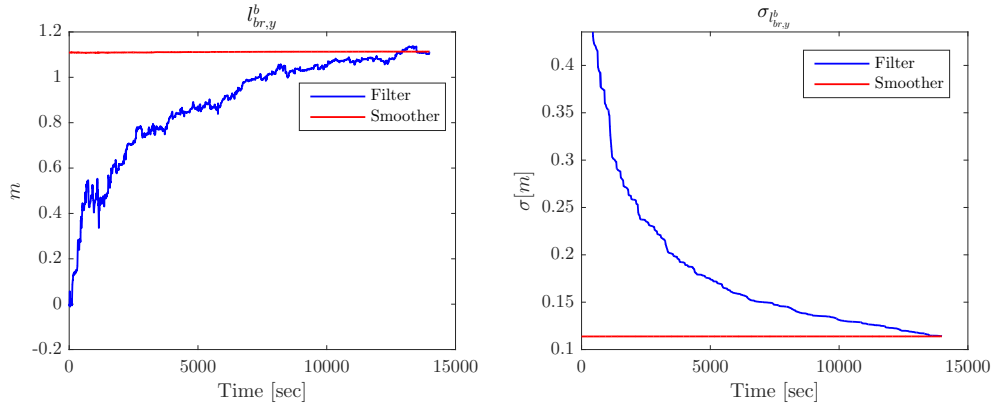


Figure 7.32: Lever arm y^b-axis

Similar the states describing the alignment in terms of Euler angles between the body and rear frame are in the state vector. It has been noted in section 6.3.2 that the roll angle ϕ_{br} is unobservable. Thus the initial uncertainties for this angle is set to zero while the pitch θ_{br} and yaw ψ^b_{br} are estimated with an initial uncertainty of $\sigma_{\theta_{br}} = \sigma_{\psi_{br}} = 1°$. All angles are initialised to $\boldsymbol{\psi}^b_{br} = [0\ 0\ 0]°$ as this is the ideally expected alignment in the bus. In figure 7.34 the angle θ_{br} converges to $-0.1°$ which confirms the expectations. The yaw angle converges very slowly in figure 7.35 to $\hat{\psi}_{br} \approx 0.4°$. This weak observability may be justified with slow vertical velocities between the rear and body frame as the slope of street does not change quickly (dynamic dependent observability [9, p.105]).

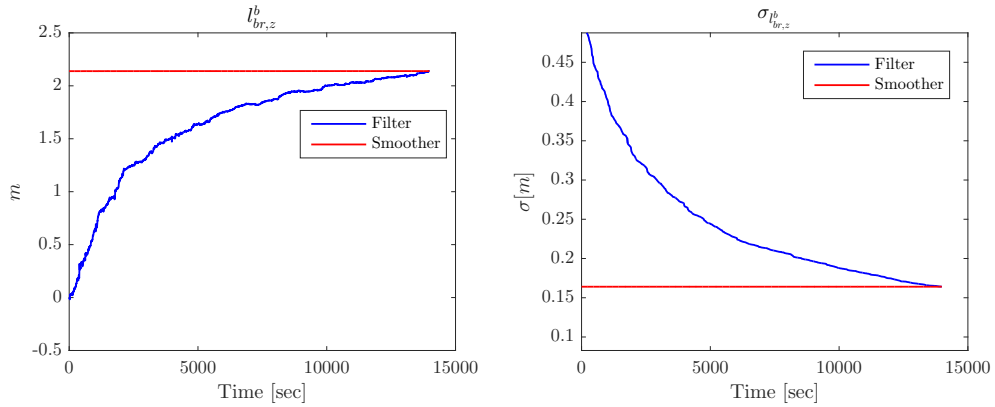


Figure 7.33: Lever arm z^b-axis

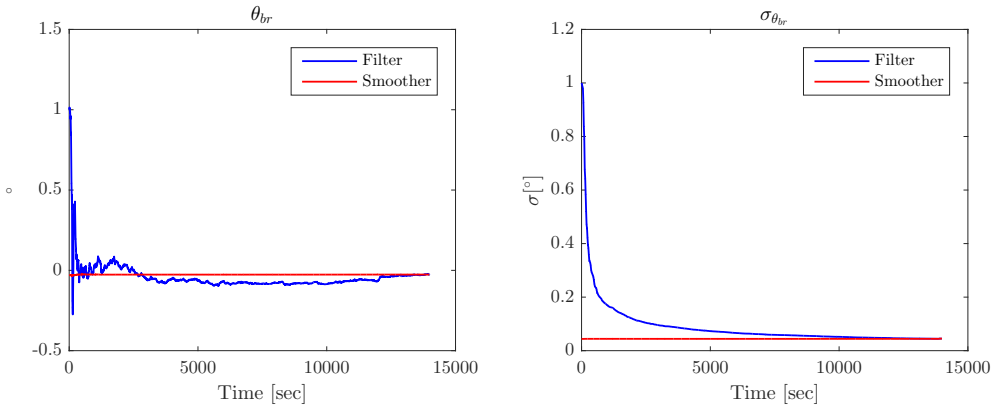


Figure 7.34: Pitch-alignment body and rear frame

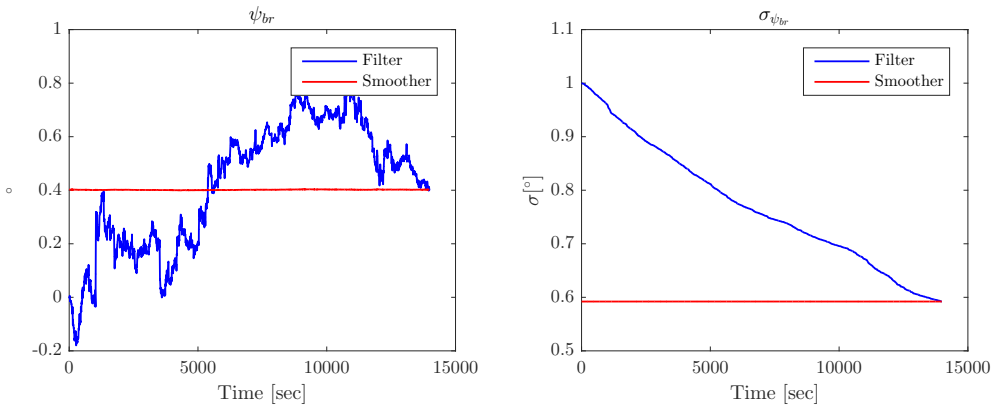


Figure 7.35: Yaw-alignment body and rear frame

8 Conclusion

8.1 Summary

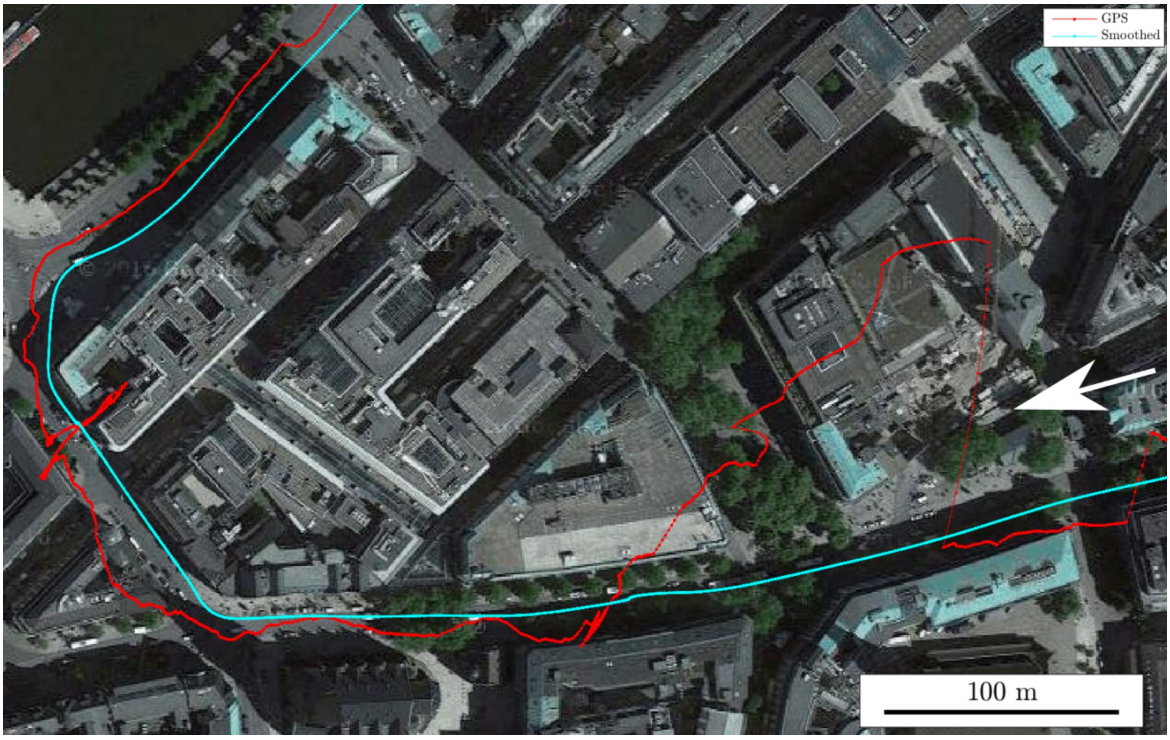


Figure 8.1: Improvements for example travel route of introduction

The introduction of the thesis described the purpose of improving the accuracy and reliability of positioning up to 2m. The disturbed GPS-signal in figure 1.1 illustrated especially the multi-path effects in urban canyons.

As a result of all previous considerations and calculations the derived algorithm is able to eliminate those GPS-errors in figure 8.1. The white arrow indicates the direction of travel and it is visible that the required lane accuracy is given. In order to achieve a statistical statement for the precision of sensor data fusion, usually the results of a navigation grade system serves as the reference. Since this kind of system was not available, a visual map matching technique has been developed and applied for two datasets in good and poor

GPS signal environment. Especially the smoother yields a remarkable accuracy under good GPS conditions of $\sigma_{smoother}$=2.5m (σ_{filter}=7m, σ_{GPS}=5m) and in poor GPS environment of $\sigma_{smoother}$=4m (σ_{filter}=11m, σ_{GPS}=20m). The forward filter does improve the accuracy only for the urban area. Due to the correlation of the GPS module's output and to fulfil the requirement for uncorrelated measurements in the Kalman filter, the update rate is set to an interval of 10 seconds. The limited performance of the filter is caused by the sensor noise which leads to an unavoidable drift in accuracy when no position update is made. The smoother is able to correct those drifts very effectively. Even large GPS outages of several minutes were successfully compensated (Section 7.4).

The height estimation is assisted by a barometer which is integrated using a Gauss-Markov model to account for pressure changes caused by the weather. Statistical data of the pressure measured over a period of three month enabled the calculation of realistic parameters. As a result a height accuracy of 1.7m with good GPS quality is achieved. The GPS module does not output negative values, thus at sea level the height noise is not zero mean. This leads to an offset in the measurements which is transported directly into the estimates.

An odometry information together with a vehicle-constraint has been integrated as an assisting velocity measurement. These information are transformed from the rear axis frame of the vehicle to the body frame of the IMU through a large lever arm (about 8m) and potential misalignment. The size of the lever arm is effectively estimated. The misalignment could only be observed for the pitch angle.

Beside the position, the Kalman filter provides also accurate estimates for the velocity and alignment in three dimensions.

For all states information about accuracy is provided. The variance of the smoother resembles the errors of the map's matching technique and the height database. This proves that the variances are in a realistic range and can be taken into account in further evaluation.

The dynamic bias of the accelerometer and especially of the gyroscope is removed by a scaling technique with temperature curves. Therefore only the static biases have to be observed by the Kalman filter which yields a higher stability for those states.

For the positioning performance of land-vehicles a good initial estimate of the yaw angle is significant to guarantee a short duration of the transient phase and system stability. A calibrated magnetometer is able to achieve this. Hence a two-dimensional elliptic calibration procedure eliminates the magnetometer's bias and scale factor effectively.

8.2 Further Research

The algorithm is implemented for post processing with Matlab. The basic structure of the algorithm can be used to develop a real time solution on the data logger. Section 6.4.3 highlights the conceptual differences in implementation with C-Code compared to Matlab.

The actual data logger configuration leads to a higher noise than would be expected from the data sheet noise density. A high data rate combined with a suitable low pass filter realised in C code could reduce the noise drastically for the accelerometer. The causes for outliers have have to be further investigated as they can lead to system instabilities.

Many parameters are determined empirically. These may be optimised in further tests. Different settings for the update rate of GPS measurements and the measurement noise parameters for the velocity constraint have the greatest influence on performance. The usage of the NMEA-String 'GCA' of the Venus module contains a vertical dilution of precision factor. This information would enable a more realistic setting for the GPS-height measurement noise.

The innovation filter technique can be implemented for all other measurements to avoid erroneous data.

An automatic tuning approach would be to create a significant amount of reference points through the map matching technique. Then the parameters could be tuned with brute force methods to minimise the error between the reference coordinates and estimates corresponding to the reference time of those points.

A new GPS receiver with shorter correlation times (single epoch receiver) would enable a smaller update interval which reduces the drift time of the INS and therefore could improve the real time filter accuracy. With a higher update rate also additional errors of the IMU are likely to become observable like scale factors which introduces an error of up to 10% for MEMS-sensors. Also GPS receiver providing access to the measured pseudo ranges would lead to a more accurate solution in tightly coupled integration strategy [9, p.567-573]. Furthermore with uncorrelated signals the innovation based adaptive estimation methods (IAE) can estimate the parameters for the system noise or measurement noise autonomously [9, p.124-125]. This approach was not successful in this thesis due to the low update rate and high correlated GPS measurements.

List of Tables

List of Figures

Bibliography

[1] AL., Yuanxin W. et: *Autonomous Land Navigation Using Inertial Sensors and an Uncalibrated Odometer: Self-calibration*. National University of Defense Technology, Changsha, 2009. – URL https://www.researchgate.net/publication/268557383

[2] BAR-YEHUDA, Zohar: *plot_google_map*. MathWorks, 2016. – URL https://www.mathworks.com/matlabcentral/fileexchange/27627-zoharby-plot-google-map

[3] BROWN&HWANG: *Random Signals and Applied Kalman Filtering*. Wiley, 2012. – ISBN 978-0-470-60969-9

[4] CAVCAR, Mustafa: *The International Standard Atmosphere*. Anadolu University, Turkey, 2005. – URL http://home.anadolu.edu.tr/~mcavcar/common/ISAweb.pdf

[5] D. H. TITTERTON, J. L. W.: *Strapdown Inertial Navigation Technology*. American Institute of Aeronautics and Astronautics, 2007. – ISBN 978-1-56347-693-8

[6] ENVIRONMENTAL INFORMATION, National C. of: *World Magnetic Model*. NOAA, 2015. – URL http://www.ngdc.noaa.gov/geomag-web/

[7] GAL, Ohad: *fit_ellipse*. Mathworks, 2003. – URL https://www.mathworks.com/matlabcentral/fileexchange/3215-fit-ellipse

[8] GREWAL&ANDREWS: *Kalman Filtering: Theory and Practise using MATLAB*. Wiley, 2015. – ISBN 978-1-118-85121-0

[9] GROVES, Paul D.: *GNSS,Inertial,and Multisensor Integrated Navigation Systems*. Artech House, 2013. – ISBN 978-1-60807-005-3

[10] HAMBURG, Universität: *Luftdruckdaten*. Meteorologisches Institut, 2016. – URL http://wettermast-hamburg.zmaw.de/

[11] H.E. RAUCH, C.T.Striebel: *Maximum Likelihood Estimates of Linear Dynamic Systems (AIAA Journal 3: 1445)*. AIAA, 1965

[12] JEKELI, Christopher: *Inertial Navigation Systems with Geodetic Applications*. Berlin, Germany: de Gruyter, 2000. – ISBN 3-11-015903-1

[13] KASTNING, Jan: *Packaging eines GPS-Datenloggers zur Integration in einen Linienbus*. HAW Hamburg, 2015

[14] KROH, Martin: *Charakterisierung eines Inertialsensor Clusters für den Einsatz in einem Kalman-Filter basierten GPS+*. HAW Hamburg, 2016

[15] LOAN, Charles F. van: *Computing Integrals Involving the Matrix Exponential*. IEEE, 1978. – ISBN IEEE Trans. Automatic Control, AC23 (3): 395-404

[16] LOAN, Charles F. V.: *A First Look at Scientific Computing and Numerical Optimization*. Cornell University Department of Computer Science, 2006. – URL https://www.cs.cornell.edu/cv/OtherPdf/Ellipse.pdf

[17] MERCEDES-BENZ: *Citaro Stadtbusse Technische Information*. Mercedes-Benz, 2015. – URL http://www.mercedes-benz.de/content/media_library/hq/hq_mpc_reference_site/bus_ng/services_accessories/brochures/2014/technic/technic_de/2014_08_01_TechInfo_Citaro_Stadt_DE_pdf.object-Single-MEDIA.tmp/TechInfo_Citaro_Stadt_de.pdf

[18] OHM, Jens-Rainer ; LÜCKE, Hans D.: *Signalübertragung Grundlagen der digitalen und analogen Nachrichtenübertragungssysteme*. Springer, 2014. – ISBN 978-3-642-5301-5

[19] OZYAGCILAR, Talat: *Calibrating an eCompass in the Presence of Hard- and Soft-Iron Interference Application Note AN4246*. NXP Freescale Semiconductors, 2015. – URL http://www.nxp.com/files/sensors/doc/app_note/AN4246.pdf

[20] RADIM HALIR, Jan F.: *Numerically Stable Direct Least Squares Fitting of Ellipses*. Proceedings of the 6th International Conference in Central Europe on Computer Graphics and Visualization (WSCG, Plzen, Czech Republic), 1998. – URL http://autotrace.sourceforge.net/WSCG98.pdf

[21] RECKTENWALD, Gerald: *Least Squares Fitting of Data to a Curve*. Portland State University, 2007. – URL http://web.cecs.pdx.edu/~gerry/nmm/course/slides/ch09Slides.pdf

[22] RICHTER, Thomas: *Planung von Autobahnen und Landstraßen*. Springer, 2016. – ISBN 978-3-658-13008-4

[23] RILL, Georg: *Road Vehicle Dynamics: Fundamentals and Modelling*. Taylor & Francis Group, LLC, 2012. – ISBN 978-1-4398-9744-7

[24] ROWELL, Derek: *Analysis and Design of Feedback Control Systems: Time-Domain Solution of LTI State Equations*. MIT, 2002. – URL http://web.mit.edu/2.14/www/Handouts/StateSpaceResponse.pdf

[25] SCHÖNE, Christoph: *Entwicklung eines integrierten Messsystems für einen GPS-Datenlogger mit verbesserter Höheninformation durch Sensorfusion*. HAW Hamburg, 2015

[26] STMICROELECTRONICS: *LSM303DLHC: Ultra compact high performance e-compass 3D accelerometer and 3D magnetometer module (Doc ID 018771 Rev 1)*. STMicroelectronics, 2011. – URL http://www.st.com/content/st_com/en/products/mems-and-sensors/e-compasses/lsm303dlhc.html

[27] STMICROELECTRONICS: *L3GD20H: MEMS motion sensor: three-axis digital output gyroscope (DocID023469 Rev 2)*. STMicroelectronics, 2013. – URL http://www.st.com/content/st_com/en/products/mems-and-sensors/gyroscopes/l3gd20h.html

[28] TECHNOLOGY, SkyTraq: *Binary Messages Of SkyTraq Venus 6 GPS Receiver (Ver 1.4.19)*. SkyTraq Technology, 2011. – URL https://elmicro.com/files/sparkfun/venus638flpx.pdf

[29] TECHNOLOGY, SkyTraq: *Venus638FLPx GPS Receiver*. SkyTraq Technology, 2011. – URL https://elmicro.com/files/sparkfun/venus638flpx.pdf

[30] TOWNSEND, Kevin: *Adafruit 10-DOF IMU Breakout*. Adafruit, 2016

[31] VERMESSUNG (LGV), Landesbetrieb G. und: *Digitales Höhenmodell Hamburg DGM10*. URL http://metaver.de/trefferanzeige?docuuid=D42A83D3-8D1A-45F9-9FD3-BEB8C51E091D, 2016

[32] VETTE, Stephan: *Studienarbeit: Entwicklung eines eingebetteten Multisensorsystem*. HAW Hamburg, 2015

[33] WEGNER, Mario: *Entwicklung eines autonom arbeitenden GPS-Datenloggers mit hoher Updatefrequenz für die Anwendung in Nahverkehrsbussen*. HAW Hamburg, 2015

Declaration

I declare within the meaning of section 25(4) of the Examination and Study Regulations of the International Degree Course Information Engineering that: this Bachelor report has been completed by myself independently without outside help and only the defined sources and study aids were used. Sections that reflect the thoughts or works of others are made known through the definition of sources.

Hamburg, August 24, 2016

City, Date

Sign