TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

D I P L O M A R B E I T

Preprocessing of IMU Observables and Multi-Sensor Kalman Filtering

ausgeführt am

Department für Geodäsie und Geoinformation
Forschungsgruppe Höhere Geodäsie

der Technischen Universität Wien

unter der Anleitung von

Univ.-Prof.Dipl.-Ing.Dr.techn. Robert Weber
Dipl.-Ing. Fabian Hinterberger

durch

Stefan Schaufler, BSc
MatrNr.: 0926563
Alxingergasse 27–29/19
1100 Wien, Österreich

Wien, Mai 2016

Stefan Schaufler

Abstract

The aim of navigation is to determine the position and the spatial orientation of a moving vehicle. This so-called trajectory can be achieved by using the combination of multiple sensors. The advantages of such a setup is that the benefits of the different systems are used to obtain improved navigation information. A possible *multi-sensor system* is the integration of GNSS[1] and IMU[2], where GNSS provides accurate position information and in contrast, the inertial sensor captures high-dynamic movements and provides a direct orientation determination. At GNSS outages (loss of signal), the navigation information is based on the IMU until the GNSS solution is available again. One of the limiting factors are the temporal increasing sensor errors of the inertial navigation. To achieve a navigation solution in the cm-range, these sensor errors must be modelled. Furthermore, the IMU measurements contain vehicle vibrations and measurement noise beside the actual vehicle motion wanted. Therefore, the resulting position error is proportional to these disorders. The entire multi-sensor navigation solution will most likely be improved when these disorders get eliminated.

In this thesis, the *wavelet analysis* is used to minimize the undesirable effects due to measurement noise and other disturbances. *Kalman filtering* is used for the sensor integration of GNSS and IMU and in addition, the sensor errors can be estimated within this technique. Finally, the effect of preprocessing the IMU observables is analysed within the multi-sensor navigation system. Thereby improvements of the initial orientation have been achieved, which leads to a smaller position and orientation error at the beginning of the trajectory. Likewise, it was shown that undesirably recorded vibrations can be eliminated, resulting subsequently in a smaller position error at GNSS outages.

[1] Global Navigation Satellite Systems, e.g. GPS
[2] Inertial Measurement Unit

Kurzfassung

Ziel der Navigation ist die zeitliche Bestimmung der Position und der räumlichen Orientierung eines bewegten Fahrzeuges. Diese sogenannte Trajektorie kann mit Hilfe der Kombination von mehreren Sensoren ermittelt werden. Dabei werden die Vorteile der unterschiedlichen Systeme genutzt, um eine verbesserte Navigationsinformation zu erzielen. Ein mögliches *Multi-Sensor-System* wäre somit die Integration von GNSS[3] und IMU[4]. Bei dieser Sensorenintegration liefert GNSS die präzise Positionsinformation und im Gegensatz dazu zeichnet der Trägheitssensor hoch-dynamische Bewegungen auf und liefert eine direkte Orientierungsbestimmung. Bei GNSS-Ausfällen (Signalverlust) übernimmt der Trägheitssensor die Positionierung, bis die GNSS-Lösung wieder möglich ist. Einer der limitierenden Faktoren ist, dass das inertiale Navigationssystem von Sensorenfehlern stark beeinflusst wird. Um trotzdem eine Navigationslösung im cm-Bereich zu erreichen, müssen diese Sensorenfehler modelliert werden. Des Weiteren enthält die inertiale Sensormessung außer den tatsächlichen Fahrzeugbewegungen auch Vibrationen und Messrauschen. Daher sind die resultierenden Positionsfehler proportional zu diesen Störungen. Es ist anzunehmen, dass die gesamte Multi-Sensor-Navigationslösung verbessert wird, wenn diese Störungen bestmöglich eliminiert werden.
In dieser Arbeit werden mit Hilfe der *Wavelet-Analyse* die unerwünschten Effekte auf Grund des Messrauschens minimiert. Die *Kalmanfilterung* wird für die Sensorintegration von GNSS und IMU verwendet. Zusätzlich können mit dieser Technik die Sensorenfehler mitgeschätzt werden. Am Schluss wird analysiert, wie sich diese Arbeitsschritte am Multi-Sensor-Navigationssystem auswirken. Dabei wurden signifikante Verbesserungen der Anfangsorientierung erreicht, was zu einem geringeren Positions- und Orientierungsfehler zu Beginn der Fahrzeugbewegung führt. Ebenso wurde gezeigt, dass unerwünscht aufgezeichnete Vibrationen eliminiert werden können, was in weiterer Reihenfolge zu einem geringeren resultierenden Positionsfehler bei GNSS-Ausfällen führt.

[3]Globales Navigationssatellitensystem, z.B. GPS
[4]Inertiale Navigation bzw. Trägheitsnavigation

Vielen Dank an Prof. Robert Weber, Fabian Hinterberger,
meinen Freunden, meiner Familie und besonders Isabella.

Contents

1 Introduction

The field of navigation deals with the determination of the position and the spatial orientation of a moving object. With the help of this information, several applications such as routing or guidance can be performed. A technology used most often in practice to achieve the requested navigation information uses satellite systems. It is possible to estimate the position information directly with *Global Navigation Satellite Systems* (*GNSS*) like the well-known *GPS*. To achieve information about the spatial orientation of the vehicle, it is possible to work with multiple GNSS antennas (see page 288 in Hofmann-Wellenhof et al. [2003]). The navigation information can be provided completely by GNSS without any additional sensors. The problem with such a setup is that the navigation solution is just based on one single positioning system and the weaknesses of GNSS directly influences the navigation performance. A disadvantage of GNSS may be the relatively low data rate (<10 Hz), which causes problems with high kinematic applications. Additionally, in the case of GNSS outages, which is common in dense urban areas or when driving through a tunnel, the navigation information cannot be provided because of a loss of satellite tracking. Therefore, GNSS is not suitable enough for many applications. To avoid such cases, GNSS is combined with another navigation sensor which is based on a totally different measurement method that performs a relative positioning (dead reckoning). A common assistant technique to support GNSS are *Inertial Navigation Systems* (*INS*). An inertial navigation system uses an *IMU* (*Inertial Measurement Unit*) to measure forces and rotation rates. Position, velocity and attitude can be computed from this information at a very fast rate. The big drawback of this navigation system are the uncertainties due to sensor errors, which increase with time. Therefore, the big advantage of an integration of GNSS and INS is that the weaknesses of both systems are quite complementary and so the systems support each other perfectly. GNSS works as a position-fixing system and delivers a position and velocity estimation of high quality, which is used to correct

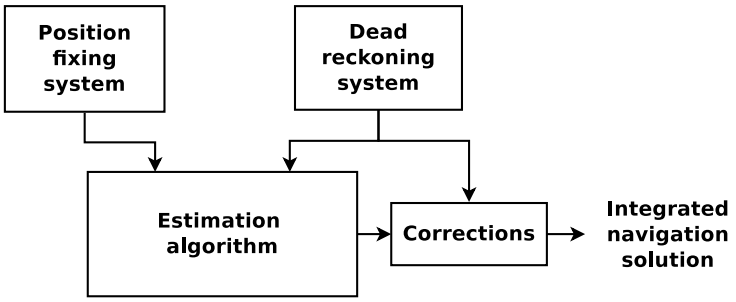


fig. 1.1: A typical integrated navigation architecture [Groves, 2008]

the navigation solution from the INS (see figure 1.1). In contrast, the INS provides the position, velocity and attitude estimation at a high data rate and is able to bridge when the GNSS solution gets lost. In practice it is common to use a Kalman filter for the sensor integration of GNSS and IMU.
But there are also some limiting factors: The inertial sensor outputs also contain sensor errors and measurement noise beside the actual vehicle motion. The schematic frequency spectrum of IMU measurements in figure 1.2a shows that the actual motion of interest is superimposed by noise and errors due to IMU sensor instability. As the inertial navigation system is a relative positioning technique, the error accumulates, which subsequently leads to poor long-term stability. This means that the IMU computed solution contains an error accumulation depending on time. To achieve the best result, also the sensor errors and measurement noise must be considered. One established strategy is to estimate the IMU sensor errors within the sensor integration algorithm. The multi-sensor setup is used to calibrate the IMU sensor drift in real-time. Additionally, it is a well-known fact that the IMU suffers from very high measurement noise. This noise affects the IMU measurement accuracy, which leads to inaccurate navigation information. If such a noise component can be sufficiently removed, the overall navigation solution is expected to improve. Figure 1.2 represents the schematic spectrum of IMU measurements after preprocessing and sensor error estimation within the IMU/GNSS integration.

In this thesis, the basics of the inertial navigation are described first (chapter 2.1), followed by a short introduction into the fundamental techniques which were important for this thesis: Wavelets analysis (3.1), Kalman filter (3.3) and sensor integration strategies (3.2). The practical part starts with chapter 4 and includes the implementation of the sensor fusion of GNSS and IMU. Sensor integration was performed by using a Kalman filter. Furthermore, the algorithm was extended to estimate the inertial sensor errors in real-time. To gain information about the noise characteristics of the recorded IMU data, signal

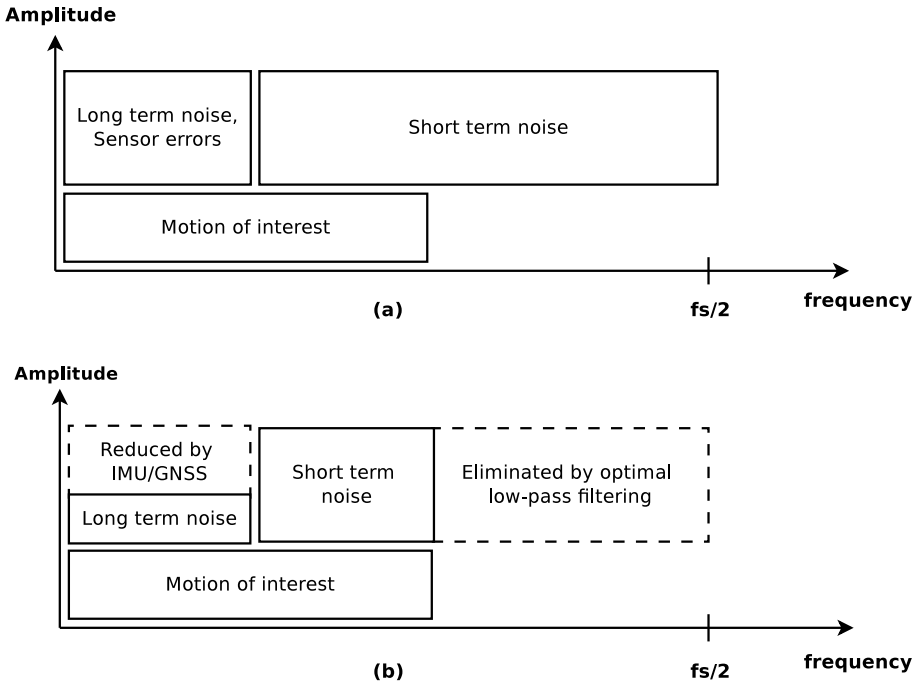


fig. 1.2: Schematic frequency spectrum of inertial measurements: (a) before filtering, (b) after IMU/GNSS integration and optimal low-pass filtering [Skaloud et al., 1999]

analysing was carried out. The *preprocessing of the IMU data* part (chapter 5) deals with analysing the spectrum of the IMU measurements to figure out its noise characteristics. Additionally, synthetic noise is added to the inertial measurements (section 5.1.3) to stress the navigation system and to analyse its behaviour in such situations. Another step is to improve the IMU measurement by de-noising the signal with the wavelet thresholding technique (section 5.2). Chapter 6 deals with the analysis of the alignment by using noised and de-noised IMU data and GNSS outages are simulated to research the behaviour of the multi-sensor system in such situations. Finally, chapter 7 summarizes the results achieved and lists impulses for future work in this field of research.

2 Navigation Systems

2.1 Inertial Navigation Systems

This chapter presents a short overview about *inertial navigation*. The overview is based on the books [Hofmann-Wellenhof et al., 2003] and [Jekeli, 2001]. Inertial navigation is a dead-reckoning technique to determine position, velocity and attitude of a vehicle. Observed accelerations and rotation rates in a well-defined reference frame are used to obtain the navigation information. The accelerations are sensed by a force measurements system called *accelerometer*. *Gyroscopes* are used to stabilize the reference frame (usually denoted as navigation frame) by monitoring their angular motions.

Starting from a known velocity $\dot{\boldsymbol{x}}(t_0)$ at the time t_0, the current velocity is computed by integration of the vehicle accelerations $\ddot{\boldsymbol{x}}(t)$:

$$\dot{\boldsymbol{x}}(t) = \dot{\boldsymbol{x}}(t_0) + \int_{t_0}^{t} \ddot{\boldsymbol{x}}(\tau) d\tau \tag{2.1}$$

With a further integration, the current position $\boldsymbol{x}(t)$ relative to the start position $\boldsymbol{x}(t_0)$ can be obtained:

$$\boldsymbol{x}(t) = \boldsymbol{x}(t_0) + \int_{t_0}^{t} \dot{\boldsymbol{x}}(\tau) d\tau \tag{2.2}$$

In formula 2.1 and 2.2 some pre-information is needed: The initial velocity vector $\dot{\boldsymbol{x}}(t_0)$ and the position vector $\boldsymbol{x}(t_0)$ of a certain time t_0 must be known. In addition, the attitude of the vehicle, i.e. the rotation of the body frame with respect to the navigation frame, must also be determined. Measurements take place in the body frame (the coordinate

frame of the vehicle), but they have to be transformed to the navigation frame. The process of obtaining the pre-information is called *initialization* or *alignment* and will be explained in section 2.1.4. But first of all, a short overview of coordinate frames which are important for inertial navigation is introduced.
The term *inertial navigation system* (INS) is used to describe the combination of an IMU (*inertial measurement unit*), the *navigation equation* and a processor.

2.1.1 Coordinate Frames

In inertial navigation, some coordinate frames have to be considered. A brief description of these frames follows.

The inertial frame ($\boldsymbol{r}^i$) is a reference frame that does not accelerate or rotate. The *Newton's laws of motion* apply in such a system. In navigation applications we use a coordinate frame which has its origin at the earth's center of mass but does not rotate with earth. Due to the earth orbiting the sun, accelerations occur and therefore this frame is called *quasi-* or *pseudo-inertial* frame. Nevertheless, the classical Newtonian dynamics apply, because the gravitational field of the solar system is relatively weak and the curvature of space-time is very small [Jekeli, 2001].

The earth-fixed frame ($\boldsymbol{r}^e$) has its origin at the earth's center of mass, but it follows the rotation of the earth. The x-axis points toward the *Greenwich* meridian, the z-axis is a mean direction of the rotation axis of the earth and the y-axis completes the system to become a three-dimensional, orthogonal right-handed system [Hofmann-Wellenhof et al., 2003]. The implementation for such systems are *WGS84* (reference frame for GPS) or *ITRF* (*Inertial Terrestrial Reference Frame*).

The local-level frame or navigation frame ($\boldsymbol{x}^l$) is a locally defined cartesian coordinate system. Its origin is a point on a reference ellipsoid and its axes point toward the *north* (forming a tangent to the meridian), *east* (forming a tangent to the circle of latitude) and *down* (ellipsoidal normal). The navigation information such as the velocity of a vehicle is often described in this frame.

The body frame ($\boldsymbol{x}^b$) is defined by the axes of the IMU, which is mounted onto the vehicle. Normally they are supposed to point in the vehicle directions front, right and

down (see figure 2.1). The angles which describe the rotation from the body to the local-level frame are usually called *roll* η, *pitch* χ and *yaw* α (*Euler angles*).

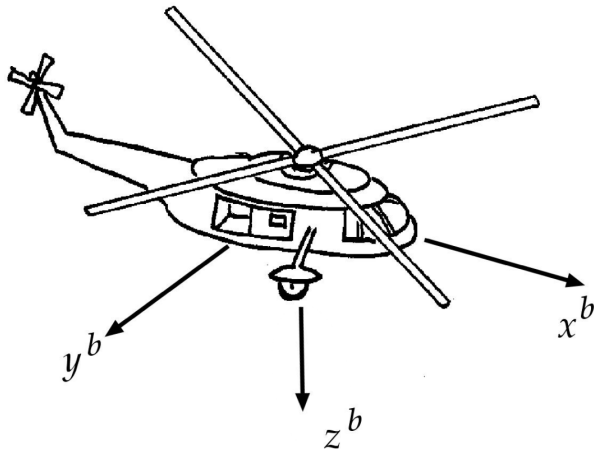


fig. 2.1: Body system [Blauensteiner, 2008]

The measurements and the calculation of the position, velocity and attitude of the trajectory refer to different coordinate systems. It is therefore essential to transform between the individual coordinate systems. In general, a coordinate transformation between systems of the same origin is written as:

$$\boldsymbol{x}^q = \boldsymbol{R}_p^q \boldsymbol{x}^p \tag{2.3}$$

In this formula, the vector $\boldsymbol{x}^p$ is defined in the p-frame and $\boldsymbol{x}^q$ is the same vector defined in the q-frame. The transformation matrix $\boldsymbol{R}_p^q$ is an orthogonal matrix and describes the transformation from p-frame into q-frame. Several possibilities exist how to define the transformation matrix. It can be described, for example, by *Euler angles* or *Quaternions*. For more information the reader is referred to Jekeli [2001].

2.1.2 Inertial Measurement Unit

Inertial navigation requires orthogonal triads of *accelerometers* and *gyroscopes* (abbr. gyros) to measure the full acceleration and angular velocity vectors, which affects an object [Gebre-Egziabher et al., 2009]. The accelerometers are used to measure the specific force vector $\boldsymbol{f}^b$ along the axis of a coordinate frame and the triad of gyros are used to

sense the spatial orientation $\boldsymbol{\omega}_{ib}^{b}$. The *inertial measurement unit* (IMU) combines those two triads on a platform (see figure 2.2). According to Hofmann-Wellenhof et al. [2003], there are two types of platforms: *gimbaled* and *strapdown*.

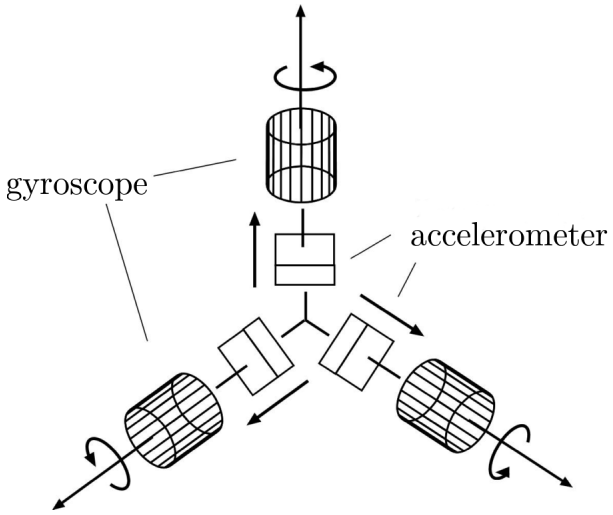


fig. 2.2: Schematic representation of an IMU [Blauensteiner, 2008]

Gimbaled or stabilized system: The orientation of this type of platform is kept stable with respect to the chosen navigation frame. This means that the platform is isolated from the rotational dynamics of the vehicle. This is done by three orthogonal gimbals.

Strapdown system: The platform is mounted onto the vehicle and tracks all motions performed by it. All measurements are performed in the body frame and the gyros are used to determine the rotation of the body frame in relation to the navigation frame.

Due to the technical development of the sensors and the high technical effort at the gimbaled systems, most navigation systems are nowadays realized as strapdown systems. Typically output rates vary between 100 and 1000 Hz [Groves, 2008]. Strapdown system technique is also used in this thesis. Depending on the chosen navigation frame, two considerations have to be made:

- The force $\boldsymbol{f}$ measured from the IMU includes the acceleration information $\boldsymbol{a}$ wanted and the local gravity vector $\boldsymbol{g}$.

$$\boldsymbol{f}^{b} = \boldsymbol{a}^{b} + \boldsymbol{g}^{l} \tag{2.4}$$

- So-called apparent forces occur due to the rotation of the body frame in relation to the inertial frame (*centrifugal* and *coriolis* force).

2.1.3 Navigation Equation

The aim of navigation systems is to deliver the position $\boldsymbol{r}^e$, velocity $\boldsymbol{v}^l$ and attitude $\boldsymbol{R}_b^l$ of a moving object at any point in time. This information can be derived from IMU sensors measurements. The relationship between quantities measured by the IMU (accelerations $\boldsymbol{f}^b$ and angular velocity $\boldsymbol{\omega}_{ib}^b$) (2.5) and the information wanted (2.6) is given by a system of differential equations. These differential equations are referred to as *navigation equations*.

$$\boldsymbol{f}^b = \begin{pmatrix} f_x^b \\ f_y^b \\ f_z^b \end{pmatrix} \qquad \boldsymbol{\omega}_{ib}^b = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{2.5}$$

$$\boldsymbol{r}^e = \begin{pmatrix} \phi^e \\ \lambda^e \\ h^e \end{pmatrix} \qquad \boldsymbol{v}^l = \begin{pmatrix} v_N \\ v_E \\ v_D \end{pmatrix} \qquad \boldsymbol{R}_b^l = R(\eta, \chi, \alpha) \tag{2.6}$$

The navigation equations depend on the chosen frame. In fact, it is commonly a combination of the local-level frame and earth-fixed frame. The position coordinates are described in an earth-fixed frame and on the other hand, the velocity vector is required to be known in a local-level frame. A third piece of information wanted is the rotation of the vehicle with respect to the navigation frame. In formula 2.7, the final navigation equation can be seen. The base of the derivation of the navigation equations (2.7) is Newton's law, which enables the description of a moving object in inertial space and can be found in Jekeli [2001].

$$\dot{\boldsymbol{r}}^e = \boldsymbol{D}^{-1}\boldsymbol{v}^l \tag{2.7a}$$

$$\dot{\boldsymbol{v}}^l = \boldsymbol{R}_b^l \boldsymbol{f}^b - (\boldsymbol{\Omega}_{il}^l + \boldsymbol{\Omega}_{ie}^l)\boldsymbol{v}^l + \boldsymbol{g}^l \tag{2.7b}$$

$$\dot{\boldsymbol{R}}_b^l = \boldsymbol{R}_b^l(\boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{il}^b) \tag{2.7c}$$

In this thesis, the position is described in ellipsoid coordinates (ϕ^e, λ^e, h^e) and therefore the matrix $\boldsymbol{D}^{-1}$ in 2.7a performs the transformation between the local-level frame and the earth-fixed frame.

$$\boldsymbol{D}^{-1} = \begin{pmatrix} \frac{1}{M+h} & 0 & 0 \\ 0 & \frac{1}{(N+h)cos\phi} & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{2.8a}$$

$$\text{with:} \qquad M = \frac{a(1-e^2)}{(1-e^2 sin^2\phi)^{\frac{3}{2}}} \qquad N = \frac{a^2}{\sqrt{a^2 cos^2\phi + b^2 sin^2\phi}} \tag{2.8b}$$

In the formula above, M and N are the meridian and prime vertical radius of curvature of the ellipsoid. a denotes the major radii and e the eccentricity of the reference ellipsoid.

It was mentioned in the last section that the quantities measured are superposed by earth gravity and must therefore be corrected by the gravity vector $\boldsymbol{g}^l$. The effective gravity depends on the spatial position and includes also the centrifugal force. Information on how this effective gravity vector can be evaluated and, in addition, information about some position depending gravity models can be found in Jekeli [2001]. The Coriolis acceleration, which occurs due to the motion of the vehicle relative to the rotating earth, is symbolized in the second term of 2.7b. The third and final navigation equation describes the relationship between the gyro measurements $\boldsymbol{\Omega}_{ib}^l$ and the rotation matrix $\boldsymbol{R}_b^l$. The rotation matrix between the body and the local system $\boldsymbol{R}_b^l$ is known as *attitude matrix*. As only the angular velocities between the local and the body frame are required, the

gyro measurements are compensated by the rotation rate of the local-level frame relative to the inertial frame, parametrized in the body frame $\boldsymbol{\Omega}_{il}^{b}$ [Hinterberger et al., 2011]. The position, velocity and attitude information can now be obtained by integration.

2.1.4 Initialization and Alignment

As already mentioned, various conditions must be determined before an inertial navigation system is ready to perform. *Initialization* is the process of obtaining the

- initial position (ϕ_0, λ_0, h_0),
- initial velocity $(v_{N,0}, v_{E,0}, v_{D,0})$,
- initial orientation of the IMU with respect to the navigation frame $(\eta_0, \chi_0, \alpha_0)$.

Information about position and velocity must be initialized using external information. This may be achieved by importing it manually by the user or by getting it delivered from another navigation system like GNSS. A procedure to determine the initial orientation is called *course alignment*. In this approach, the vehicle must stay in idle state ($\boldsymbol{v}^l = 0$); the IMU just senses *static data*. The IMU measurements (the specific force vector $\boldsymbol{f}^b$ and rotation rate vector $\boldsymbol{\omega}_{ib}^{b}$) are provided in the body frame and a relationship with two known quantities can be established: the rotation rate of the earth $\boldsymbol{\omega}_{ib}^{l}$ and the local-level gravity vector $\boldsymbol{g}^l$. These quantities are normally known in a local frame and they are used to provide the initial alignment of the IMU. According to Jekeli [2001], a third vector $\boldsymbol{c}^b$ is defined by forming the cross product between the measured gravity vector and the earth rotation rate vector, which leads to following relationships:

$$\boldsymbol{f}^b = -\boldsymbol{R}_l^b \boldsymbol{g}^l \tag{2.9a}$$

$$\boldsymbol{\omega}_{ib}^{b} = \boldsymbol{R}_l^b \boldsymbol{\omega}_{ib}^{l} \tag{2.9b}$$

$$\boldsymbol{c}^b = \boldsymbol{f}^b \times \boldsymbol{\omega}_{ib}^{b} = \boldsymbol{R}_l^b (-\boldsymbol{g}^l \times \boldsymbol{\omega}_{ib}^{l}) \tag{2.9c}$$

It was mentioned above that the gravity vector and the earth rotation vector of the initialization position (ϕ_0, λ_0) must be known (2.10). The information about gravity can either be obtained from by gravity measurements or theoretically determined by gravity models.

$$\boldsymbol{g}^l = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix}, \qquad \boldsymbol{\omega}_{ib}^l = \begin{pmatrix} \omega_e cos\phi_0 \\ 0 \\ -\omega_e sin\phi_0 \end{pmatrix} \tag{2.10}$$

With this information, the following relationship can be established (see 2.11):

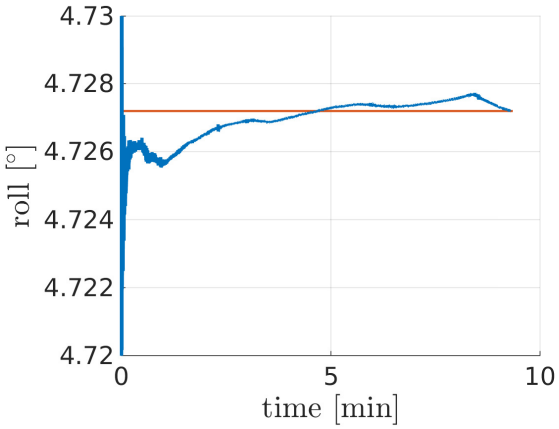
$$\begin{pmatrix} \boldsymbol{f}^b & \boldsymbol{\omega}_{ib}^b & \boldsymbol{c}^b \end{pmatrix} = \boldsymbol{R}_l^b \begin{pmatrix} 0 & \omega_e cos\phi_0 & 0 \\ 0 & 0 & -g\omega_e cos\phi_0 \\ -g & -\omega_e sin\phi_0 & 0 \end{pmatrix} \tag{2.11}$$

The rotation matrix $\boldsymbol{R}_l^b$ can now be calculated at each epoch by solving the equation 2.11:

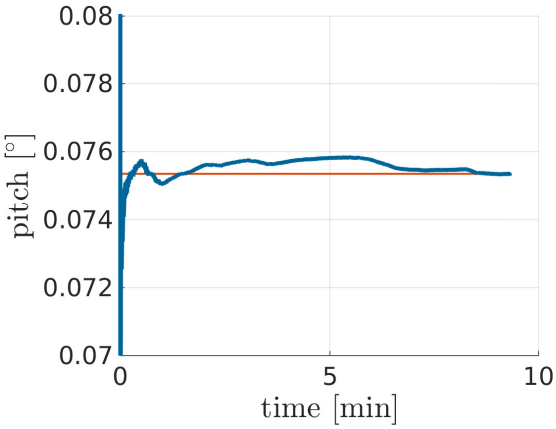
$$\boldsymbol{R}_l^b = \begin{pmatrix} \boldsymbol{f}^b & \boldsymbol{\omega}_{ib}^b & \boldsymbol{c}^b \end{pmatrix} \begin{pmatrix} 0 & \omega_e cos\phi_0 & 0 \\ 0 & 0 & -g\omega_e cos\phi_0 \\ -g & -\omega_e sin\phi_0 & 0 \end{pmatrix}^{-1} \tag{2.12}$$

Because of the high sensor noise of accelerometers and gyros, the components of the rotation matrix have to be averaged over some epochs. The averaging process usually requires some minutes of static data. The time length depends on the sensor noise and can be shortened by using noise filter methods (see section 6.1). After that, the Euler angles (*roll* η, *pitch* χ, *yaw* α) can be extracted from the rotation matrix. These angles are used to indicate the rotation between body and local coordinate frame.

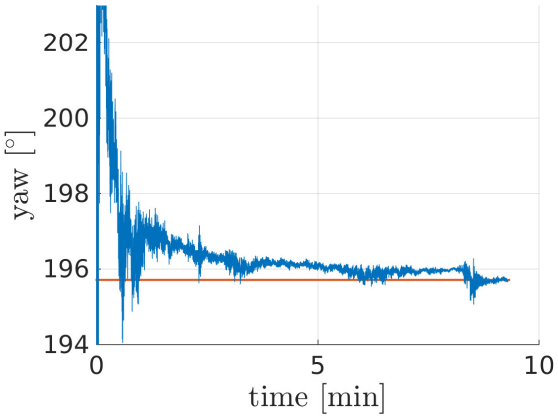
$$\boldsymbol{R}_l^b = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{2.13a}$$



(a) *Roll* alignment



(b) *Pitch* alignment



(c) *Yaw* alignment

fig. 2.3: Determination of the Euler angles

$$\eta = arctan\left(\frac{r_{21}}{r_{11}}\right), \qquad \chi = arcsin(-r_{31}), \qquad \alpha = arctan\left(\frac{r_{32}}{r_{33}}\right) \tag{2.13b}$$

Figure 2.3 shows how the choice of the initialization time affects the alignment. Therefore, the Euler angles at a certain time are calculated and displayed. It can be seen that *roll* η and *pitch* χ can be obtained with a higher accuracy than the *yaw* α angle. The reason for this is that the determination of roll and pitch are based on the local gravity vector, which is a much stronger signal than the earth rotation vector. The yaw angle is just based on the earth rotation rate data sensed. That leads to a worse possibility of determining the yaw angle.

2.1.5 System Performance

The operation of an inertial navigation system is limited as a result of data errors. According to Titterton and Weston [2004], the error sources which determine the navigation performance can be categorized as:

- Initial Alignment Errors
- Sensor Errors
- Computational Errors

Inaccuracies arise in such a system because of initial alignment errors and imperfections in the performance of the IMU. Also the speed of the computation must be high enough. Otherwise, significant errors can arise because it is not possible to sense and record the actual motion. The initial alignment and sensor error sources are introduced in this thesis. The following sections deal with these errors and how they affect the navigation system performance.

Sensor Errors

It was mentioned before that the quality of the inertial navigation information depends on the inertial sensor errors, which cause a rapid degradation in the integrated navigation solution, when not treated properly. The growth of these errors depends on the type of the inertial sensor. Normally they are categorized in high-, medium- or low-graded IMUs, based on the increase rate of the horizontal position error in nautical miles per hour [Jekeli, 2001]. The major error sources in gyroscopes and accelerometers can be divided as follows:

- **Bias or Drift** is a result of manufacturing imperfections of the sensors. It is the constant offset of the sensor measurement from its true value (zero point shift) and does not depend on the measurements.
- **Scale factor** is the ratio of the output and the true physical quantity being measured. In contrast to the bias offset, the scale factor is proportional to the measurements and increases at highly dynamic movements.
- **Axis misalignment** arises from the misalignment of the sensitive axes with respect to the orthogonal axes of the body frame due to manufacturing limitations.

- **Random noise** are random errors in the measurements and has white noise characteristics. It can be reduced with *wavelet thresholding*. (see chapter 3.1.3)

Using the notation from Groves [2008], the following equations show the main error sources of accelerometer $\delta \boldsymbol{f}^b$ and gyroscope $\delta \boldsymbol{\omega}_{ib}^b$:

$$\delta \boldsymbol{f}^b = \boldsymbol{b}_a + \boldsymbol{M}_a \boldsymbol{f}^b + \boldsymbol{w}_a \tag{2.14}$$

$$\delta \boldsymbol{\omega}_{ib}^b = \boldsymbol{b}_g + \boldsymbol{M}_g \boldsymbol{\omega}_{ib}^b + \boldsymbol{w}_g \tag{2.15}$$

Where $\boldsymbol{f}^b, \boldsymbol{\omega}_{ib}^b$ denote the true force and angular rate vector, $\boldsymbol{b}_a, \boldsymbol{b}_g$ denote the corresponding biases and $\boldsymbol{w}_a, \boldsymbol{w}_g$ random noise. The matrices $\boldsymbol{M}_a, \boldsymbol{M}_g$ include the scale s and misalignment m factors.

$$\boldsymbol{M}_a = \begin{pmatrix} s_{a,x} & m_{a,xy} & m_{a,xz} \\ m_{a,yx} & s_{a,y} & m_{a,yz} \\ m_{a,zx} & m_{a,zy} & s_{a,z} \end{pmatrix} \qquad \boldsymbol{M}_g = \begin{pmatrix} s_{g,x} & m_{g,xy} & m_{g,xz} \\ m_{g,yx} & s_{g,y} & m_{g,yz} \\ m_{g,zx} & m_{g,zy} & s_{g,z} \end{pmatrix} \tag{2.16}$$

The accelerometer and gyro errors of an inertial navigation system consist of two parts: a deterministic part and a random part [Nassar and El-Sheimy, 2005]. The deterministic or systematic part can be corrected by the IMU processor using laboratory calibration data ([Groves, 2008], p.112). Random errors can be studied and modelled as stochastic processes. These stochastic models can be considered by a *Kalman filter* to provide optimized estimation of the navigation parameters. The stochastic model in the navigation filters is normally a random constant (RC), a random walk (RW), a Gauss-Markov model (GM), or autoregressive (AR) processes. For more information about this stochastic models and how to model IMU sensor errors the reader is referred to Nassar [2003].

When not using high-grade IMUs, it is necessary to estimate the random sensor errors mentioned above. Only then it is possible to achieve the best inertial navigation performance. The random part consists of a high-frequency and a low-frequency component and cannot be corrected by calibration. De-noising techniques like *wavelet thresholding*

can filter the high-frequency term. According to El-Diasty and Pagiatakis [2008], the low-frequency random errors (bias and scale) can usually be modelled by stochastic models inside a Kalman filter at each epoch and then simultaneously removed during the navigation equation implementation. This can be achieved through integration with another navigation sensor (e.g. GNSS).

Initial Alignment Errors

Coarse alignment is a straightforward method that requires high-quality gyros and accelerometers with small biases and high signal-to-noise ratio to get useful results. A disadvantage of this method is that no information on the accuracy of the orientation is achieved during this process. The alignment solution is largely determined by the precision of the measurements from the instrument outputs. According to Rogers [2003], only a rough estimation can be carried out. In this approach the navigation frame misalignments in north ϵ_N, east ϵ_E and down ϵ_D depend on the sensor errors $(\delta \boldsymbol{f}, \delta \boldsymbol{w})$.

$$\epsilon_N = -\frac{\delta f_y}{g} \tag{2.17a}$$

$$\epsilon_E = -\frac{\delta f_x}{g} \tag{2.17b}$$

$$\epsilon_D = \frac{\delta f_y}{g} tan\phi_0 + \frac{\delta \omega_y}{\omega_e cos\phi_0} \tag{2.17c}$$

It was shown in figure 2.3 that the roll and pitch angle are faster adjusted than the yaw angle. The reason for this is that roll and pitch is based on gravity vector measurements. On the other hand, the reason for the difficult determination of true yaw angle stems from the fact that it can just be determined by earth rotation rate measurements. This can be seen in formula 2.17c. The earth rotation rate signal is weak and therefore often hidden behind noise. The overall coarse alignment procedure suffers extremely from the sensor errors and additionally from the sensor noise. It is difficult to give an exact statement about the accuracy of the alignment because it depends on the IMU sensor quality. In order to improve the precision of the alignment, a data de-noising procedure, such as

the wavelet thresholding introduced in chapter 3.1.3, can be applied [El-Sheimy et al., 2004]. The improvements of de-noising IMU data for initialization is shown in chapter 6.1.2. Another improvement of the initialization can be achieved by adding external information like azimuth measurements or by estimating the sensor errors with the help of a Kalman filter (zero velocity update). Furthermore, a so-called *kinematic alignment* is possible if external information like GNSS observations are available [Jekeli, 2001].

2.2 GNSS as an Aiding System

Global Navigation Satellite Systems (*GNSS*), which were originally designed for military purposes, enable a global and accurate determination of the position and the velocity of a moving object. The systems are designed that the navigation information can be obtained anywhere in the world and in addition, GNSS perform at any time and in any weather condition. These are useful characteristics for the purpose of navigation. Worldwide various satellite positioning systems are in operation. The most famous one is the American *GPS*, but also the Russian *GLONASS*, European *GALILEO* or the Chinese *BeiDou* operate worldwide. All different satellite positioning systems are combined in the term Global Navigation Satellite Systems *GNSS*. For more information about the GNSS architecture and observables the reader is referred to e.g. Hofmann-Wellenhof et al. [2012].

Navigation information delivered by GNSS can be derived in different quality. In the navigation approach of this thesis, GNSS measurements are used to update the navigation information from the inertial sensors (see chapter 4). The position and velocity information delivered is subsequently used to reset the time-depending IMU sensor drift and in addition, to estimate the IMU sensor errors. Therefore, a position accuracy of few centimeters is needed. This can be reached by observing the carrier-phases. To solve the ambiguities *kinematic relative positioning* is performed. In this approach, the GNSS measurements, collected by the vehicle's receiver (*rover*), are compared with measurements from one or more reference stations. A reference station (*base*) is a receiver on a known point. This base station data stems either from a real located receiver or from a so-called virtually located reference station (*VRS*). Assuming that the GNSS receivers observe the same satellites simultaneously, double-differences can be built to resolve the ambiguities and to eliminate error sources. Therefore, relative positioning determines the vector between the two receivers (*rover* - *base*), which is called baseline [Hofmann-Wellenhof et al., 2012]. The resulting rover's position solution refers to the coordinate frame of the base. An important fact is that the reliable ambiguity solution depends on the distance between the receivers. It is supposed that the signal paths to the two receivers are almost identical, because there are also some limitations due to distance-depending errors mainly caused

by signal delays in the atmosphere.

The resulting position accuracy depends on the distance between the GNSS receivers. To achieve a reliable result, the maximum baseline length should not exceed 20 km.
Ambiguity resolution in relative positioning can be performed either in post-processing or on-the-fly in real-time called *Real-Time Kinematik* (*RTK*). In this thesis, GNSS data were collected and stored to estimate the position in post-processing mode. Thereby, after accomplishing to solve the ambiguities, cm-level position accuracy can be achieved. Additionally to the position information, GNSS delivers also velocity information. This is performed through *Doppler-shift* measurements.

In contrast to inertial navigation systems, GNSS is a position fixing system. Thus, no information on previous positions of the receiver is necessary. This characteristics of GNSS is used to achieve high-precision position measurements, which subsequently update the multi-sensor system. In addition to the GNSS position delivered at a certain time, its stochastic information is gained. This information is directly adopted in the stochastic model of the Kalman filter (see section 4.1.3) and is needed in the Kalman gain computation. The Kalman gain is used to specify the weighting of the measurements and is introduced in chapter 3.3.

3 Fundamental Techniques

3.1 Wavelets

In this thesis, *wavelets* are used to analyse and de-noise the IMU sensor outputs. Wavelet technology enables possibilities that other signal analysis techniques miss. Detecting trends, discontinuities or sharp spikes in the signal is made possible. Wavelets are also capable of compressing or de-noising a signal without degradation of the original signal. [Grap, 1995]
In inertial navigation the gyroscope and accelerometer output suffers from relatively high measurement noise. The noise affecting the inertial sensor contains two parts:

- Low frequency component: Has random process characteristics which can be modelled with sufficient accuracy.
- High frequency component: Has white noise characteristics and is not possible to model.

The performance of inertial measurements is expected to improve if the white noise component can be removed from the original signal. This can be achieved by using a de-noising technique like *wavelet decomposition*. [Nassar and El-Sheimy, 2005]
In this chapter, the basics of *wavelets* and *wavelet transform* are described first, following by *wavelet multi-resolution analysis* (3.1.3) which can be used to de-noise signals. Keller [2004] gives a good introduction into the wavelet theory and therefore, the following overview about wavelets is based on this book.

3.1.1 Definition

The technique of using wavelets is based on analyzing a signal through signal windowing with variable window sizes. Wavelets $\psi(t)$ are small wave-packages where following conditions are fulfilled (cf. [Keller, 2004], p. 31):

- $\psi(t)$ must decay quickly at both ends:

$$0 < c_\psi = 2\pi \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty \tag{3.1}$$

  Where $\Psi(\omega)$ is the Fourier transform of the wavelet $\psi(t)$.

- From equation 3.1 it follows that $\psi(t)$ must be an oscillatory function with zero mean:

$$0 = \Psi(0) = (2\pi)^{-\frac{1}{2}} \int_{-\infty}^{\infty} \psi(t) dt \tag{3.2}$$

For the approximation of a signal a set of wavelets is used. Each wavelet is derived from the so-called *mother wavelet* ψ by scaling a and shifting b .

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right), \qquad a \in \mathbb{R}\backslash\{0\}, b \in \mathbb{R} \tag{3.3}$$

There are many kinds of mother wavelets that have been documented in the literature. For example figure 3.1 shows the Morlet and Mexican hat wavelet.

3.1.2 Wavelet Transform

Conventional Fourier transform of a signal has one big drawback: When looking at the Fourier transform of a time domain signal, it is not possible to identify when a particular event took place (e.g. change of frequency). To overcome this drawback, the wavelet transform is based on a windowing technique with variable window size to analyse a signal (see example in Keller [2004], p. 32-34). Therefore wavelets are localised in both time (via translation) and frequency (via scaling) domains.

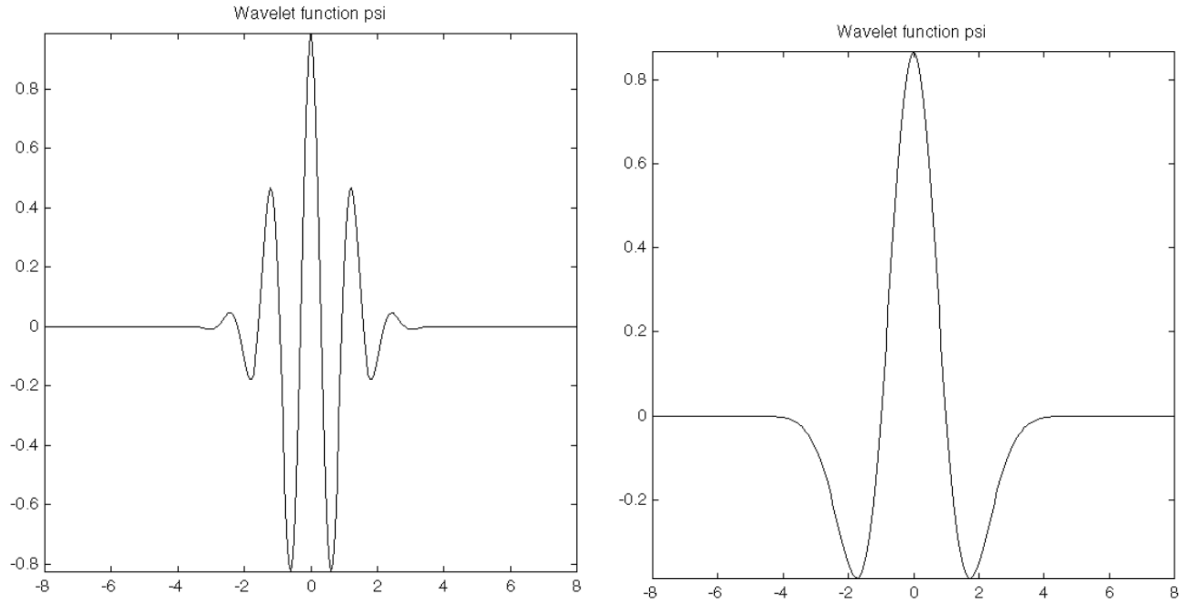


fig. 3.1: Morlet wavelet and Mexican hat wavelet [Misiti, 2006]

The wavelet transform is defined in a continuous (CWT) and discrete (DWT) way. The CWT of a function $f(t)$ is expressed in formula 3.4.

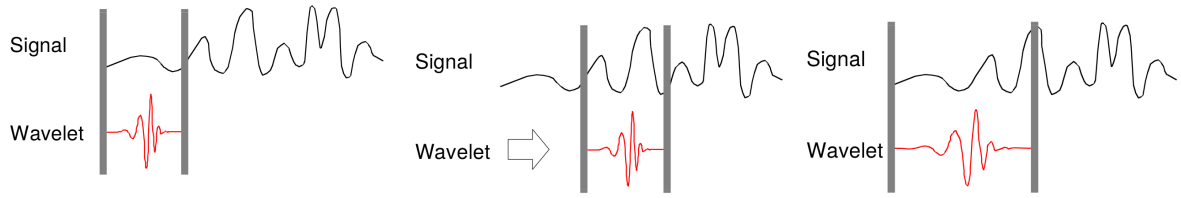
$$W_\psi(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t)\psi^*\left(\frac{t-b}{a}\right)dt, \qquad a \in \mathbb{R}\backslash\{0\}, b \in \mathbb{R} \tag{3.4}$$

In this formula, $W_\psi(a,b)$ is called *wavelet coefficient* of the function $f(t)$ with respect to the mother wavelet $\psi(t)$. The index $*$ indicates that the complex conjugate must be used in case of a complex wavelet. Figure 3.2 illustrates the steps of CWT schematically.

In practice, it is time-consuming to calculate the wavelet coefficients at every possible translation b and scale a and in addition, inertial sensor data are supplied discretely. Therefore, the CWT is performed at discrete grid points which are selected from a subset of all possible scales and translations. [Keller, 2004]

In the *discretized continuous wavelet transform* the continuous parameters a and b are sampled with the constant number α to the power of n:

$$a = \alpha^n \quad \text{and} \quad b = m\alpha^n \tag{3.5}$$



(a) Selecting mother wavelet and calculating correlation

(b) Shifting the wavelet

(c) Scaling the wavelet

fig. 3.2: Continuous wavelet transform [Misiti, 2006]

Here n and m are integer numbers presenting the discrete scaling and translation. By substituting formula 3.5 into 3.4 the DWT of a discrete signal $f(k)$ can be described:

$$W_\psi(m,n) = \frac{1}{\sqrt{\alpha^n}} \sum_k f(k)\psi^*(k\alpha^{-n} - m) \tag{3.6}$$

According to studies, if the base of the exponentiation α is two, the analysis is much more efficient and just as accurate and because of that it is used for the DWT algorithm. [Misiti, 2006].

For many signals (especially IMU sensor data) the high frequency component indicates the noise and the low frequency component is the one of interest, because it gives the signal its identity. [Nassar and El-Sheimy, 2005]. The next chapter shows how to separate the wanted components from the raw signal.

3.1.3 Wavelet Multi-Resolution Analysis

The continuous wavelet transform of chapter 3.1.2 performs a multi-resolution analysis of a signal to obtain its different frequencies with different resolutions. Information about low frequencies is achieved with high-scaled wavelets and high frequencies with low-scaled wavelets. In *discrete wavelet transform* (DWT) the multi-resolution is obtained by splitting the signal into two parts: an *approximation* part (low frequency) and a *detail* part (high frequency). This is done by passing the signal through two complementary filters: a low-pass and high-pass filter (see figure 3.3).

The result is two sets of coefficients: approximation coefficients cA_1 and detail coefficients cD_1. For the mathematical implementation of an efficient algorithm please refer to the corresponding chapter on *Mallat algorithm* in Keller [2004].

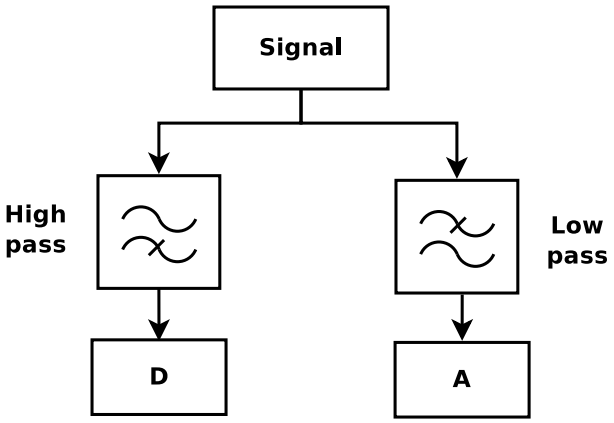


fig. 3.3: Decomposition of a signal

If f_s is the sampling rate of the data (e.g. data from the inertial sensor), the maximum detectable frequency is $f_s/2$, according to the *Nyquist theorem*. When using DWT to analyse the data, the cutoff frequency of the filters is exactly one half of the maximum detectable frequency. The approximation part will have frequencies less than $f_s/4$ and the detail part will consist of frequencies from $f_s/4$ to $f_s/2$. In a next step the approximation part will be further divided into two parts, the decomposition will be done iteratively. This technique is called *Wavelet Multi-Level of Decomposition* (LOD) (figure 3.4). As a consequence, the signal is broken down into many signal components with lower resolution. The original signal can be represented by a finite sum of components at different resolutions. The decomposition with A representing the approximation, D the detail parts and n the wavelet LOD is depicted in 3.7. This is one of the powerful properties of using wavelets for signal processing or analysing [Goswami and Chan, 2011].

$$\begin{aligned} signal &= A_1 + D_1 \\ &= A_2 + D_2 + D_1 \\ &= A_n + D_n + D_{n-1} + ... + D_1 \end{aligned} \tag{3.7}$$

At each step of decomposition, the highest observed frequency f_a in the approximation wavelet coefficients can be calculated from the sample frequency f_s as:

$$f_a = \frac{f_s}{2^{n+1}} \tag{3.8}$$

As an example, figure 3.5 shows the frequency spectrum from a gyroscope measurement

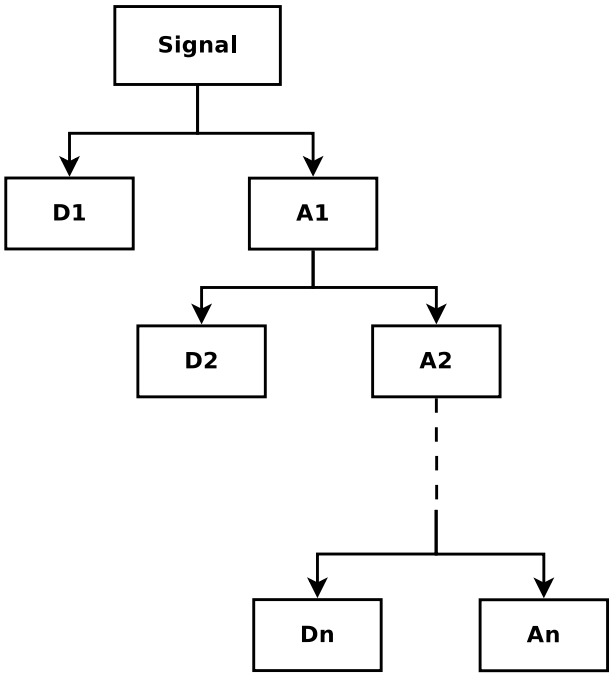


fig. 3.4: Multi-level decomposition

with different levels of decomposition. In this example, the sample frequency of the raw data is 100 Hz. Therefore, the raw data can represent frequencies up to 50 Hz (Nyquist theorem). According to formula 3.8, the signal after the second level of decomposition includes frequencies up to 12.5 Hz and the signal after the fourth decomposition represents only frequencies below 3.125 Hz.
It can be seen that this decomposition process can be done many times and after each step the inertial sensor measurement is broken down into many lower-resolution components. Theoretically, since the decomposition process is iterative, it can be done infinitely. Practically, the wavelet decomposition can only continue until the part consists of only a single frequency.

3.1.4 Signal Denoising

In the previous section, a technique was explained how to analyse and decompose a signal. The second part deals with the process of how the components can be composed again without any information loss and with additionally removing the unwanted noise. The mathematical implementation is called the *inverse discrete wavelet transform* (IDTW).

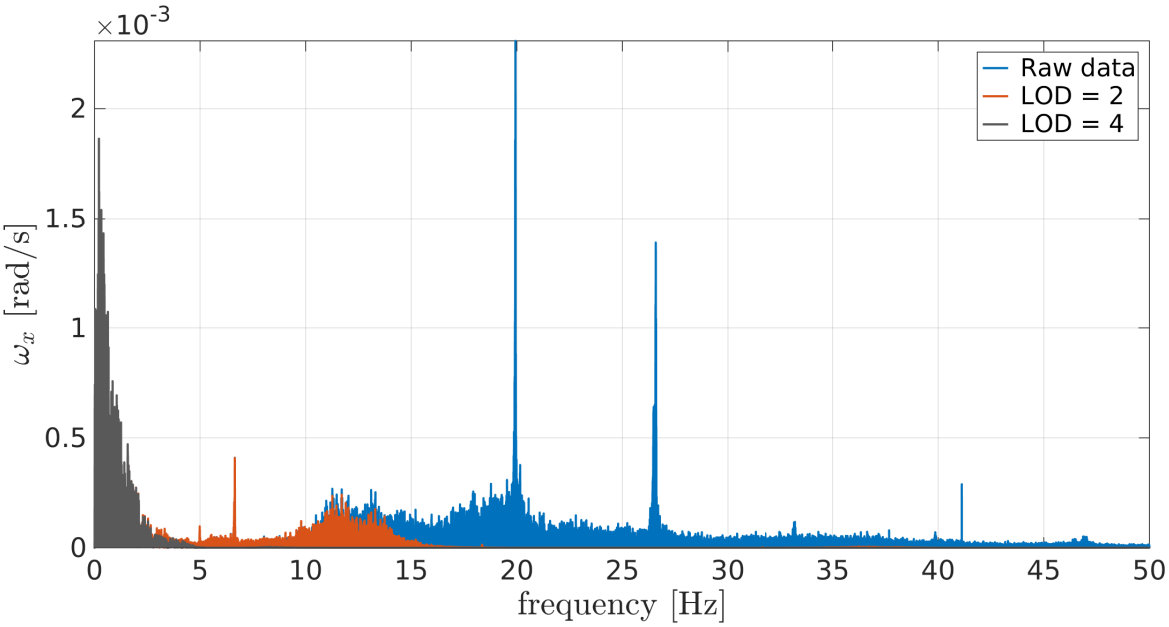


fig. 3.5: Frequency spectrum of gyroscope measurements with different level of decomposition

$$f(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} W_\psi(m,n)\psi_{m,n}(t), \qquad \text{with: } \psi_{m,n}(t) = \frac{1}{2^n}\psi(k2^{-n} - m) \tag{3.9}$$

If the aim is to de-noise the signal, not every wavelet coefficient is used for the reconstruction. A technique called *wavelet thresholding* works in the following way: The idea of thresholding is to set all coefficients which are below a defined threshold to zero. The reason for this is that some of the resulting wavelet coefficients related to details in the signal are small. Those might be excluded without affecting the main features of the data set. It is assumed that the magnitude of the wanted signal is greater than the noise. Thresholding generally gives a kind of low-pass filtering of the original signal [Grap, 1995]. Merry [2005] mentioned that the technique is a significant step forward in handling noisy data, because no sharp structures from the original signal are smoothed out. The result is a cleaned-up set of data that still shows important details.

In practice, the thresholding technique is classified into two most used operators: *soft* and *hard* thresholding. In the case of hard thresholding, all wavelet coefficients below a defined threshold are replaced by zero (3.10a). The remaining coefficients are kept without any modification. With soft thresholding, all coefficients below the threshold are replaced by zero too, but all coefficients above the selected threshold are reduced by the threshold

value (3.10b).

$$\eta^{hard} = \begin{cases} w, & |w| > t \\ 0, & |w| \leq t \end{cases} \tag{3.10a}$$

$$\eta^{soft} = \begin{cases} sign(w)(|w| - t), & |w| > t \\ 0, & |w| \leq t \end{cases} \tag{3.10b}$$

Here w represents the wavelet coefficient and t the chosen threshold. Comparing the two methods, the soft thresholding technique is known to have a better performance in the case of de-noising data than the hard thresholding technique [Kang et al., 2011]. This is due to the fact that the noise power is smaller than the power of the signal. Therefore the soft thresholding technique is used in this thesis.

There are different approaches how to define the threshold. A general threshold, which just takes the level of decomposition n into account, is determined by the formula 3.11:

$$t = \sqrt{2log(n)} \tag{3.11}$$

According to Misiti [2006], the de-noising procedure can now be carried out in three steps:

- **Decompose:** Choose a wavelet, choose a level of decomposition and compute the wavelet coefficients of the signal.
- **Apply thresholding:** For each level, select a threshold and apply soft thresholding to the coefficients.
- **Reconstruct:** Compute inverse wavelet transform to reconstruct the de-noised version of the signal.

In general, the terms *wavelet multi-resolution analysis*, *wavelet decomposition* or *wavelet thresholding* are used for the wavelet de-noising technique. For more information about wavelet thresholding and how to define thresholds refer to Donoho and Johnstone [1994].

3.2 Integration of IMU and GNSS

3.2.1 Complementary Nature of IMU and GNSS

GNSS and *inertial navigation* techniques differ in many cases. First of all GNSS is an absolut positioning system where inertial navigation is a dead-reckoning technique which uses measured forces and rotation rates to obtain its position relative to its last position. The relatively low data output from GNSS my cause problems at kinematic operations where the inertial sensor can supply between GNSS epochs at high temporal resolution. The error dynamics of an IMU are very complicated (chapter 2.1.5). It has a high short-term stability, but because of its dead-reckoning principle, residual errors accumulate which leads to a bad long-term behaviour. In contrast GNSS positioning derives from a direct measurement of distances where errors do not accumulate with time. But in short-term, where it is not possible to average, the errors are larger. GNSS positions, when derived from phase-observation data, can be used to calibrate the long-time IMU sensor errors. The GNSS system, however, depends on the space segment and a direct line of sight to the satellites is necessary. This disadvantage is eliminated in inertial navigation, because this sensor works independently (except required initial position and velocity). Therefore IMU can bridge between GNSS epochs or even when the GNSS solution is lost [Jekeli, 2001]. Table 3.1 compares the main features of GNSS and inertial navigation.

3.2.2 IMU-GNSS Integration Types

There are different methods to link GNSS and IMU. The integration depends on how the different sensor outputs are combined. The GNSS receiver can either supply raw signals, raw measurements (phase- or code-pseudoranges) or already processed information (position, velocity and time). An IMU supplies the acceleration and angular velocity related information [Gebre-Egziabher et al., 2009]. There are different approaches how these data are brought together. But in all strategies the integration is performed by means of a Kalman filter. The following section gives a short overview of different integration types.

tab. 3.1: Characteristics of GNSS and inertial navigation (from [Jekeli, 2001] and [Hofmann-Wellenhof et al., 2003])

Characteristics	**GNSS**	**Inertial navigation**
Information	absolut	relative
Operation	nonautonomous	autonomous
Measurements	pseudocode, phase, doppler shift	acceleration, rotation rate
Navigation solution	position, velocity, time	position, velocity, attitude
Output rate	low ($<$ 20 Hz)	high ($<$ 1000 Hz)
Short-term accuracy	low	high
Long-term accuracy	high	low
Availability	limited	unlimited

Uncoupled Integration

The simplest integration type is known as *uncoupled*. It is a decentralized approach which is characterized by a sequential processing of information. Both systems provide their solution independently. In the case of GNSS position, velocity, time; and for INS position, velocity and attitude. A master processor combines to a final solution. Following (Jekeli [2001] p. 311) the combining processor may be accomplished with a simple selection algorithm. The output relies on the inertial navigation only if GNSS is not available.

$$\begin{pmatrix} \boldsymbol{r} \\ \boldsymbol{v} \end{pmatrix}_k = \begin{cases} \begin{pmatrix} \boldsymbol{r}_{GNSS} \\ \boldsymbol{v}_{GNSS} \end{pmatrix}_k & \text{, if GNSS solution is available} \\ \begin{pmatrix} \boldsymbol{r}_{INS} \\ \boldsymbol{v}_{INS} \end{pmatrix}_k + \begin{pmatrix} \boldsymbol{r}_{GNSS} - \boldsymbol{r}_{INS} \\ \boldsymbol{v}_{GNSS} - \boldsymbol{v}_{INS} \end{pmatrix}_0 & \text{, if GNSS solution is not available} \end{cases} \tag{3.12}$$

Formula 3.12 represents the decision algorithm where the index k is the current epoch and *0* indicates the last epoch where GNSS was available. The inertial solution is reset to zero at GNSS update because of the IMU sensor drifts and GNSS long-term stability.

Loose Integration

In *loose architecture* the inertial navigation and GNSS processor calculates their navigation information also independently. A third navigation solution is estimated by fusing them. In contrast to the uncoupled scheme the usefulness of this loose integration type is to extract the attributes wanted from IMU and GNSS while holding the unwanted attributes down. The merged navigation solution is normally estimated by a Kalman filter (see chapter 3.3). But each sensor works as a separate navigation system. The result is a high data rate position information from the IMU and bounded errors from the GNSS. The GNSS solution resets the IMU sensor error. The loosely coupled schema is shown in figure 3.6. The box with dashed lines symbolizes that it is common to model the systematic IMU sensor errors to improve the navigation solution. Such a construction is also called *feed-forward*. To get useful results this architecture just works for high-end inertial sensors because of smaller random error characteristics of the IMU and its stability. Like it was shown in chapter 2.1.5 the IMU errors can be modelled and used for the integrated navigation solution.

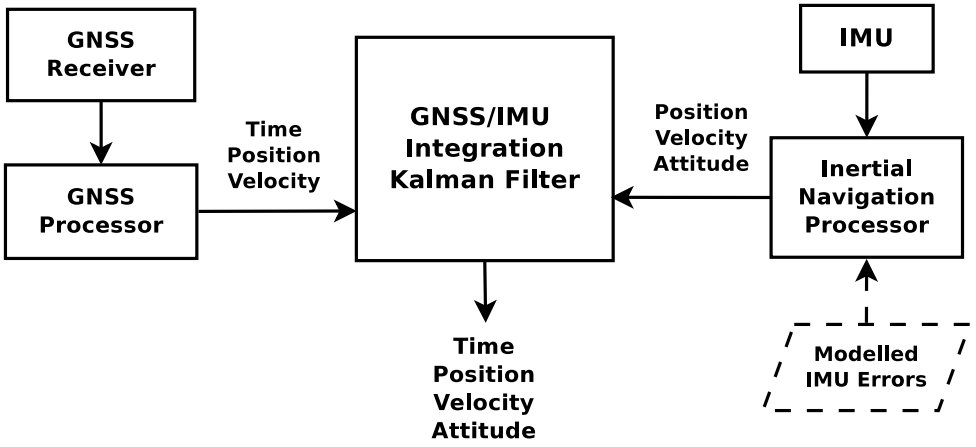


fig. 3.6: Feed-forward configuration in loose integration

Closed loop: Configuring a *feedback loop* for the intertial navigation processor is another method to reduce the inertial sensor errors. The IMU errors are estimated by the Kalman filter and the accelerometer and gyro outputs are corrected. The effect is that the sensors are calibrated in real-time. In figure 3.7 the dashed line symbolizes the feedback path. The feedback configuration is necessary for low-cost IMUs, because of their large output errors. For high-quality IMUs the feedback loop is not essential. Error characteristics are known better and they are more stable, but a closed-loop configuration can also

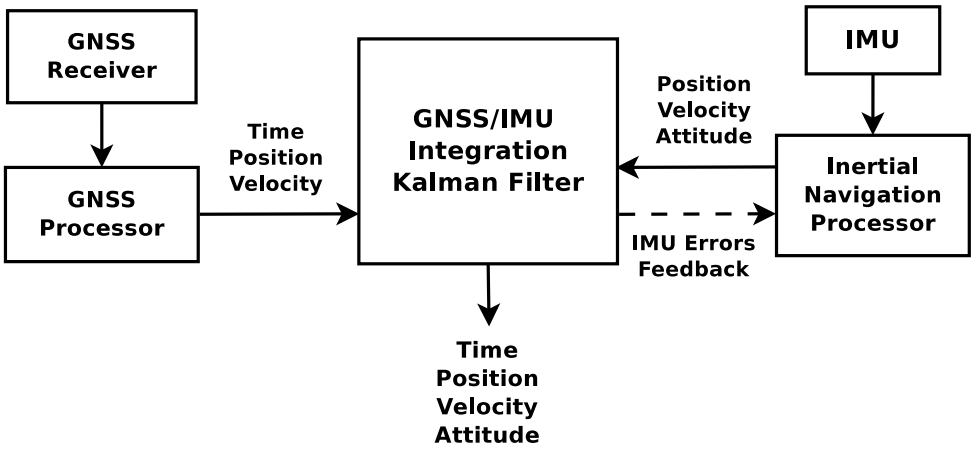


fig. 3.7: Feedback configuration in loose integration

improve the navigation solution.
Gebre-Egziabher et al. [2009] mentioned that the advantage of the loose integration approach is that it is relatively easy to implement because the integration occurs at the position, velocity and time level and not at the sensor level. This means that just little modification of the basic INS or GNSS algorithms is needed. A big advantage of loosely coupled systems is that the system still delivers a navigation solution if one of the two sensors fails, because they work independently.

Tight Integration

In this architecture type the sensors are reduced to their basic sensor outputs. In the case of IMU accelerations and gyro measurements and for GNSS pseudoranges and pseudorange rates. Pseudorange rates $\dot{P}$ can be calculated from the observed doppler-shift Δf_D using the carrier frequency f_s and speed of light c:

$$\dot{P} = -\frac{c}{f_s}\Delta f_D \tag{3.13}$$

These raw measurements are used in one central Kalman filter to generate one single navigation solution. Therefore it is also known as *centralized architecture*.

Closed loop: In tight architecture integration there are also subvariants. As shown in figure 3.8 the dashed line indicates a feedback loop. Information about position and/or velocity from Kalman filter is fed back to the receiver of the GNSS to improve its perfor-

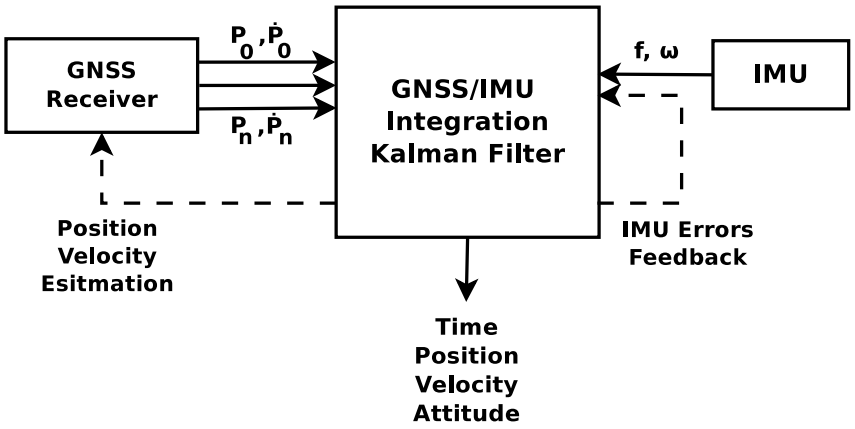


fig. 3.8: Tight coupled integration with feedback loop

mance and IMU sensor error estimations are fed back to the IMU. This information from the filter can shorten the acquisition or reacquisition time of the GNSS signal and, in addition, the velocity information can be used to aid the code and carrier tracking loops in the GNSS receiver [Gebre-Egziabher et al., 2009]. This allows the receiver to continue tracking signals even in high dynamic motions where loosely coupled systems might have problems.

Loose against Tight Integration

It should be noticed that in tight integration the GNSS raw measurements are not processed in a separate step any more but they are directly combined in one unique filter. That leads to a more accurate and more robust result than a loose integration. On the other hand, the separate filters in the loose approach have the advantage that their state vectors are smaller than of the corresponding Kalman filter in a tight configuration. That yields to faster processing times which can be useful in real time applications. The enhanced accuracy in tight strategy is, in part, due to the fact that the GNSS information used by the Kalman filter shows less temporal correlation than the processed navigation solution in loosely coupled systems [Gebre-Egziabher et al., 2009]. One main problem in loose coupled system that is mentioned in Petovello [2003] is that the estimated position supplied from the GNSS receiver is mostly Kalman filtered and the errors of Kalman filter outputs are time correlated. Whereas the Kalman filter, which finally estimates the merged navigation solution, assumes that the measurement errors are uncorrelated in time. In a tightly coupled architecture, the two Kalman filters of the loosely coupled architecture are combining into one and that eliminates the statistical problems [Groves,

2008]. The gain of robustness comes from the possibility to extract useful GNSS information even when less than four satellites are available. This information supports the IMU and the integrated system still delivers a navigation solution where a single GNSS solution would not be possible. The big drawback of tight integration is that no multiple independent navigation solution exists like in loose integration. That means that the GNSS receiver is no longer independent from the IMU any more. A badly working IMU sensor affects the solution which in turn will affect the performance of the GNSS receiver. Given the same inertial instruments and the same GNSS user equipment, a tightly coupled IMU and GNSS almost always perform better than its loosely coupled counterpart in terms of both accuracy and robustness [Groves, 2008]. But, on the other hand, loose architecture is a simple and flexible approach, which gives the possibility to fuse the existing GNSS-IMU system with other sensors. The multi-sensor system can be extended.

A summary of the individual benefits is provided in the following table 3.2.

tab. 3.2: Characteristics of loose and tight architecture [Abdel-Hamid, 2005]

Loose integration	**Tight integration**
Uses positions and velocities from GNSS	Uses range and range rate measurements from each satellite
No GNSS solution if less than 4 satellites	Use every single satellite
More vulnerable due to jamming	Better jamming resistance
Lower performance in case of high dynamics	Reliable tracking under high dynamics
Temporal correlation because of cascade architecture	Central architecture
Multiple navigation information	Single navigation information
Simple and flexible approach (possible to extend with other sensors)	Difficult to apply multi-sensor aid
Small filter size	Large filter size

Deep Integration

Another integration type, which just will be mentioned shortly, is called *deep* or *ultra-tight*. In this approach the GNSS receiver architecture is totally different from GNSS receivers which are used at the loose or tight method. The GNSS and INS devices no longer work as independet systems. GNSS measurements are used to estimate IMU errors

and IMU measurement to aid GNSS receiver tracking loops during interference or otherwise degraded signal conditions like jamming [Petovello, 2003]. This approach requires access to the GNSS receiver's firmware or the tracking loop information and is typically only implemented by equipment manufacturers. Therefore it is not treated in this thesis.

A final remark in this section is, that the loosely GNSS - IMU integration will be the one adopted herein, and will therefore be implemented for processing the inertial and GNSS measurements in this thesis. Chapter 4 will present the practical implementation of GNSS and IMU by using an extended Kalman filter which is introduced in chapter 3.3.

3.3 Kalman Filter

The basic technique of the *Kalman filter* algorithm was developed by Rudolf Kalman in 1960 and can be found in Kalman [1960]. It is an optimal least square estimator for dynamic systems. The Kalman filter is a recursive algorithm using a *system model* and *measurement model* to obtain an optimal state vector estimation. It is a favourite integration method for navigation sensor data to achieve an optimal overall system performance and has quickly become very popular since its invention [Hofmann-Wellenhof et al., 2003].

3.3.1 Structure

The key advantage of the Kalman filter over other stochastic estimation methods is its iterative structure which qualifies for use in real-time applications. As stated in Groves [2008], the algorithm consists of 5 core elements:

a) The state vector is a set of parameters which are used to describe the dynamic system. In the case of navigation, the state vector describes the trajectory and consists of components of position, velocity and attitude but also sensor errors. Related to the states are their corresponding covariances, which represent the uncertainty of the Kalman filter state estimation.

b) The system model (also called prediction model) is a function of time which describes the continuing behaviour of the state vector and its stochastic information. Normally, linear dynamic systems can be expressed with formula 3.14. Here it must be noted that manipulating variables like wind or temperature can also be modelled. But in order to simplify the following formulas, they are ignored.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{C}(t)\boldsymbol{w}(t) \tag{3.14}$$

In the formula above, $\boldsymbol{x}(t)$ represents the state vector and $\boldsymbol{F}(t)$ is called system matrix. $\boldsymbol{C}(t)$ represents the noise distribution matrix and system noise is denoted as $\boldsymbol{w}(t)$

and generally describes the uncertainties of modelling the dynamic system behaviour [Hofmann-Wellenhof et al., 2003]. As measurement updates take place at discrete time stamps, the system model must be expressed discretely. In the discrete Kalman filter the state vector estimation can be achieved as a linear or linearized function of its previous state:

$$\boldsymbol{x}_{k+1} = \boldsymbol{T}_{k,k+1}\boldsymbol{x}_k + \boldsymbol{S}_{k,k+1}\boldsymbol{w}_k \tag{3.15}$$

In 3.15, $\boldsymbol{x}_k$ describes the state vector at a certain time; $\boldsymbol{w}_k$ is system noise, which should be Gaussian distributed with zero mean ($\boldsymbol{w} \sim N(\boldsymbol{0}, \boldsymbol{Q}_k)$) and $\boldsymbol{S}_{k,k+1}$ represents the noise coefficient matrix. The system transition matrix $\boldsymbol{T}_{k,k+1}$ updates the state vector from epoch t_k to the epoch t_{k+1}. It is usually calculated by a power-series expansion of system matrix $\boldsymbol{F}$ as follows. (cf. [Jekeli, 2001], page 221):

$$\boldsymbol{T}_{k,k+1} = exp(\boldsymbol{F}(t - t_0)) = \sum_{i=0}^{N}(\frac{\boldsymbol{F}^i}{i!}(t - t_0)^i)) \tag{3.16}$$

c) The measurement vector is a set of measurements of properties of the system. For example it can be the position and velocity measurement from GNSS. In association with the measurement vector, its covariance matrix describing the uncertainty of the measurements must be provided.

d) The measurement model combines the measurement vector with the state vector. It is be used to update the state vector estimation with a set of measurements. In a standard Kalman filter the measurement vector $\boldsymbol{l}(t)$ is modelled as a linear function of the state vector $\boldsymbol{x}(t)$, where $\boldsymbol{A}(t)$ presents the measurement matrix (or design matrix) and $\boldsymbol{v}(t)$ the noise term.

$$\boldsymbol{l}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{v}(t) \tag{3.17}$$

If the measurement vector is taken at discrete time stamps, 3.17 becomes:

$$\boldsymbol{l}_k = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{v}_k \tag{3.18}$$

e) The Kalman filter algorithm is the centerpiece. It uses system model, measurement vector and measurement model to obtain optimal state vector estimation. It consists of three main steps:

- **Step 1: Time update (prediction)**
  For the time update step, no measurements are required. The prediction is based only on the system model and is done at a user-defined interval. Starting from the estimated state vector at epoch t_k, the system states are predicted forward in time using:

$$\bar{\boldsymbol{x}}_{k+1} = \boldsymbol{T}_{k,k+1}\hat{\boldsymbol{x}}_k \tag{3.19}$$

  while the propagation of the parameter covariances $\boldsymbol{\Sigma}_{\bar{x}\bar{x}}$ at epoch t_{k+1} is given by:

$$\boldsymbol{\Sigma}_{\bar{x}\bar{x},k+1} = \boldsymbol{T}_{k,k+1}\boldsymbol{\Sigma}_{\hat{x}\hat{x},k}\boldsymbol{T}_{k,k+1}^T + \boldsymbol{Q}_k \tag{3.20}$$

- **Step 2: Gain computation**
  On arrival of a new measurement $\boldsymbol{l}$ at epoch t_{k+1}, the measurements of the true system must be compared with the predicted system. Depending on the stochastic properties of the measurements and system states, a so-called *gain-matrix* is calculated to specify the weighting of the measurements or predictions. The Kalman gain $\boldsymbol{K}$ is computed using the covariance matrix of the innovation $\boldsymbol{D}$:

$$\boldsymbol{K}_{k+1} = \boldsymbol{\Sigma}_{\bar{x}\bar{x},k+1}\boldsymbol{A}_{k+1}\boldsymbol{D}_{k+1}^{-1} \tag{3.21}$$

$$with: \boldsymbol{D}_{k+1} = \boldsymbol{\Sigma}_{ll,k+1} + \boldsymbol{A}_{k+1}\boldsymbol{\Sigma}_{\bar{x}\bar{x},k+1}\boldsymbol{A}_{k+1}^T \tag{3.22}$$

- **Step 3: Measurement update (correction)**
  The measurement is used to update the prediction to achieve the best estimation. First, the measurement innovation $\boldsymbol{d}$ is computed by calculating the difference of the measurement vector $\boldsymbol{l}$ with the predicted measurements delivered by $\boldsymbol{A}_{k+1}\bar{\boldsymbol{x}}_{k+1}$.

$$\boldsymbol{d}_{k+1} = \boldsymbol{l}_{k+1} - \boldsymbol{A}_{k+1}\bar{\boldsymbol{x}}_{k+1} \tag{3.23}$$

The estimation of the state parameters $\hat{\boldsymbol{x}}_{k+1}$ is now done as a weighted merge of the predicted state vector with the observations using the innovation and Kalman gain.

$$\hat{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{x}}_{k+1} + \boldsymbol{K}_{k+1}\boldsymbol{d}_{k+1} \tag{3.24}$$

While the estimated covariance matrix of the states is given by:

$$\boldsymbol{\Sigma}_{\hat{x}\hat{x},k+1} = \boldsymbol{\Sigma}_{\bar{x}\bar{x},k+1} - \boldsymbol{K}_{k+1}\boldsymbol{D}_{k+1}\boldsymbol{K}_{k+1}^{T} \tag{3.25}$$

The schematic algorithm of the Kalman filter is presented in figure 3.9.

3.3.2 Extended Kalman Filter

The Kalman filter algorithm is designed for linear systems, but in real applications like inertial navigation systems this is not the case. The functional dependencies of the measurement and/or system model are non-linear. The system matrix $\boldsymbol{F}(t)$ and/or measurement matrix $\boldsymbol{A}(t)$ are replaced by non-linear functions $f(\boldsymbol{x}(t))$ and $a(\boldsymbol{x}(t))$. Linear approximation has to be applied. This can be achieved by linearization to a predetermined trajectory (*Linearized Kalman filter*) or to the predicted one (*Extended Kalman filter*). "The advantage of using a predicted trajectory for the linearization is that the discrepancies are usually smaller compared to using an nominal trajectory." ([Hofmann-Wellenhof et al., 2003], page 278)

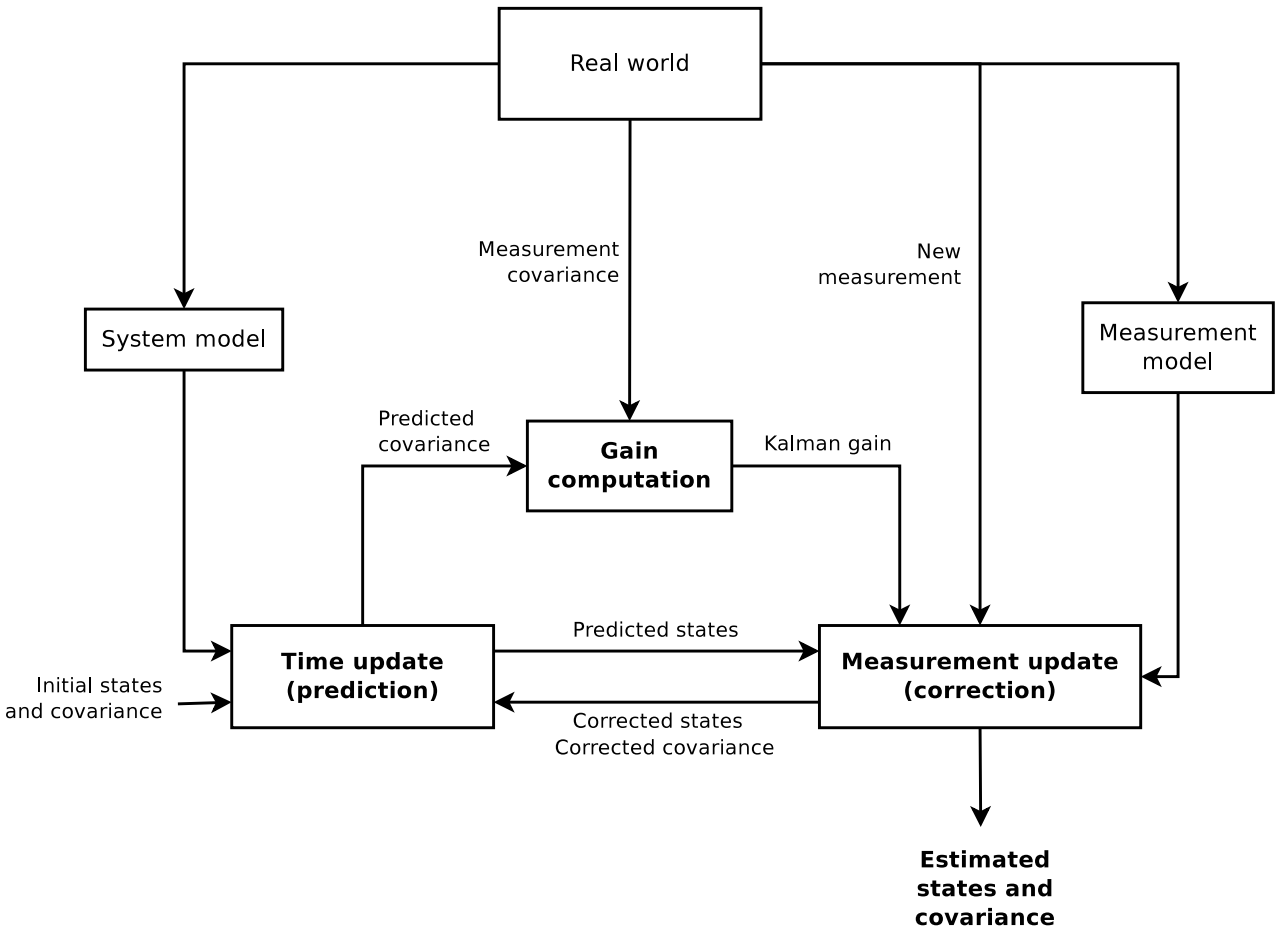


fig. 3.9: Kalman filter algorithm

The system model of 3.14 becomes:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}(t)) + \boldsymbol{w}(t) \tag{3.26}$$

Linearizing the system model with respect to the predicted trajectory $\hat{\boldsymbol{x}}$ for epoch k is done by evaluating the Jacobian:

$$\boldsymbol{F}_k = \left.\frac{\partial f_k(\boldsymbol{x}_k)}{\partial \boldsymbol{x}}\right|_{x=\hat{x}_k} \tag{3.27}$$

Now the system transition matrix $\boldsymbol{T}_{k,k+1}$ can be computed with formula 3.16 which is used for every time update step. If the measurement model $a(\boldsymbol{x}(t))$ is non-linear, linearizing of

the design matrix must also be carried out:

$$\boldsymbol{l}(t) = a(\boldsymbol{x}(t)) + \boldsymbol{v}(t) \qquad \text{and} \qquad \boldsymbol{A}_k = \left.\frac{\partial a_k(\boldsymbol{x}_k)}{\partial \boldsymbol{x}}\right|_{x=\hat{x}_k} \tag{3.28}$$

After these steps, the discrete Kalman filter algorithm can be used as given in the previous section. Only the following changes must be made:

- System states are now predicted with:

$$\bar{\boldsymbol{x}}_{k+1} = f_k(\hat{\boldsymbol{x}}_k) \tag{3.29}$$

- Innovation is calculated using:

$$\boldsymbol{d}_{k+1} = \boldsymbol{l}_{k+1} - \boldsymbol{a}_{k+1}(\bar{\boldsymbol{x}}_{k+1}) \tag{3.30}$$

It was shown that the Kalman filter algorithm has a recursive structure which qualifies for use in real-time applications. Therefore, it has become a favourite integration method for navigation sensors. In chapter 4 the practical sensor fusion of IMU and GNSS is performed by using an extended Kalman filter.

4 Practical Implementation of the Sensor Fusion

Different integration concepts to combine an IMU with GNSS were introduced in chapter 3.2.2. The integration type that was chosen for the sensor fusion in this thesis was the *loose architecture* with a *closed-loop Kalman filter*. This Kalman filter integration algorithm is an estimator where multiple sources can be combined; in addition to that, the closed-loop approach enables the Kalman filter to correct the system itself, because the error states are fed back at every iteration step. The big advantage of a loosely coupled system is that its approach is simple and flexible and can be extended with other navigation sensors in future. In the integration scheme used, the positions and velocities from GNSS are used to update the predictions which are derived from the IMU measurements.

The Kalman filter integration of IMU and GNSS was developed by the research group *Advanced Geodesy* at the *Department of Geodesy and Geoinformation* at *TU Wien*. The basic concept of the sensor fusion was implemented by *Franz Michael Blauensteiner* (see Blauensteiner [2008]) and extended by *Fabian Hinterberger* (see Hinterberger et al. [2011]). The most important steps to implement the integration of IMU and GNSS using a Kalman filter are explained in the following section.

4.1 Extended Kalman Filter Design

The Kalman filter technique is based on the fundamental assumption of linearity. The problem in real applications is that this assumption cannot always be applied, because the observation equations and/or the system dynamics are non-linear. In our case, the system equations (equations of motion) are clearly not linear. Therefore, an extended

Kalman filter (EKF) has to be used for the GNSS/INS integration. The EKF is designed for non-linear systems. This algorithm has two positive aspects (see section 3.3): Kalman filters generate optimal estimates of the navigation states by minimizing the covariance of the estimation error and their recursive form makes them well-suited to efficient implementation.

The first step to implement a Kalman filter centers on the question of state space. It is standard that the state vector is described in error-state instead of total state [Jekeli, 2001]. This means that the state parameters describe the differences between the estimated position, velocity, and attitude and the true position, velocity, and attitude (formula 4.1).

$$\delta \boldsymbol{x}_k = (\underbrace{\delta\phi \quad \delta\lambda \quad \delta h}_{\delta \boldsymbol{r}^e} \quad \underbrace{\delta v_N \quad \delta v_E \quad \delta v_D}_{\delta \boldsymbol{v}^l} \quad \underbrace{\epsilon_N \quad \epsilon_E \quad \epsilon_D}_{\boldsymbol{\epsilon}^l})_k^T \tag{4.1}$$

Furthermore, the error-state vector is extended with inertial sensor errors. Inertial sensor errors were introduced in section 2.1.5. The sensor bias $\boldsymbol{b}_a^b$, $\boldsymbol{b}_g^b$ and the scale factor $\boldsymbol{s}_a^b$, $\boldsymbol{s}_g^b$ for each gyroscope or accelerometer are estimated as well. In total, the state vector $\delta \boldsymbol{x}_k$ consists of 21 states.

$$\delta \boldsymbol{x}_k = (\delta \boldsymbol{r}^e \quad \delta \boldsymbol{v}^l \quad \delta \boldsymbol{\epsilon}^l \quad \boldsymbol{b}_a^b \quad \boldsymbol{s}_a^b \quad \boldsymbol{b}_g^b \quad \boldsymbol{s}_g^b)_k^T \tag{4.2}$$

The true value of the navigation information ($\boldsymbol{r}^e$, $\boldsymbol{v}^l$, $\boldsymbol{R}_b^l$) can be calculated with the help of the IMU derived navigation information ($\boldsymbol{r}_{IMU}^e$, $\boldsymbol{v}_{IMU}^l$, $\boldsymbol{R}_{bIMU}^l$) and the error-states from the estimated state vector and is expressed by the equations 4.3.

$$\boldsymbol{r}_{IMU}^e = \boldsymbol{r}^e + \delta \boldsymbol{r}^e \tag{4.3a}$$

$$\boldsymbol{v}_{IMU}^l = \boldsymbol{v}^l + \delta \boldsymbol{v}^l \tag{4.3b}$$

$$\boldsymbol{R}_{bIMU}^l = (\boldsymbol{I} - \boldsymbol{E})\boldsymbol{R}_b^l \tag{4.3c}$$

$$with: \boldsymbol{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{E} = \begin{pmatrix} 0 & -\epsilon_D & \epsilon_E \\ \epsilon_D & 1 & -\epsilon_N \\ -\epsilon_E & \epsilon_N & 0 \end{pmatrix}$$

4.1.1 System Model

It was introduced in chapter 3.3 that the *system model* (or *prediction model*) is one fundamental part of the Kalman filter. Its equation, which describes the continuing behaviour of the state vector, is presented here in formula 4.4 again.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{C}(t)\boldsymbol{w}(t) \tag{4.4}$$

In the formula above, $\boldsymbol{x}(t)$ represents the state vector; $\boldsymbol{F}(t)$ is called system matrix; $\boldsymbol{C}(t)$ represents the noise coefficient matrix and $\boldsymbol{w}(t)$ is system noise. The system equations of the IMU-GNSS Kalman filter are called *error dynamic equations* and are derived from the navigation equations (2.7). The navigation equations are non-linear and by linearization the final error dynamic equations are achieved (formula 4.5). According to Gebre-Egziabher et al. [2009], this can be performed by a first-order *Taylor series expansion* or *perturbation analysis*. Their derivation can be found e.g. in Jekeli [2001] or Reinstein [2010].

$$\delta\dot{\boldsymbol{r}}^e = \boldsymbol{F}_{\dot{r}r}\delta\boldsymbol{r}^e + \boldsymbol{F}_{\dot{r}v}\delta\boldsymbol{v}^l \tag{4.5a}$$

$$\delta\dot{\boldsymbol{v}}^l = \boldsymbol{F}_{\dot{v}r}\delta\boldsymbol{r}^e + \boldsymbol{F}_{\dot{v}v}\delta\boldsymbol{v}^l - \boldsymbol{\epsilon}^l \times \boldsymbol{f}^l + \boldsymbol{R}_b^l\delta\boldsymbol{f}^b + \delta\boldsymbol{g}^l \tag{4.5b}$$

$$\dot{\boldsymbol{\epsilon}}^l = \boldsymbol{F}_{\dot{\epsilon}r}\delta\boldsymbol{r}^e + \boldsymbol{F}_{\dot{\epsilon}v}\delta\boldsymbol{v}^l - \boldsymbol{\omega}_{il}^l \times \boldsymbol{\epsilon}^l - \boldsymbol{R}_b^l\delta\boldsymbol{\omega}_{ib}^b \tag{4.5c}$$

In addition, it was mentioned in chapter 2.1.5 that it is necessary to estimate the inertial sensor errors ($\delta\boldsymbol{f}^b$, $\delta\boldsymbol{\omega}_{ib}^b$) in order to achieve the best navigation performance. The equations for the modelled sensor errors were already shown in 2.1.5. In the implementation, which is presented in this section, bias $\boldsymbol{b}$ and scale factor $\boldsymbol{s}$ of the accelerometers and gyros

are taken into account (axis-misalignments are not modelled in this implementation). The simplified form of 2.14 and 2.15 are presented again in formulas 4.6a and 4.6b.

$$\delta \boldsymbol{f}^b = \boldsymbol{b}_a + \boldsymbol{M}_a \boldsymbol{f}^b \qquad with: \boldsymbol{M}_a = \begin{pmatrix} s_{a,x} & 0 & 0 \\ 0 & s_{a,y} & 0 \\ 0 & 0 & s_{a,z} \end{pmatrix} \tag{4.6a}$$

$$\delta \boldsymbol{\omega}_{ib}^b = \boldsymbol{b}_g + \boldsymbol{M}_g \boldsymbol{\omega}_{ib}^b \qquad with: \boldsymbol{M}_a = \begin{pmatrix} s_{a,x} & 0 & 0 \\ 0 & s_{a,y} & 0 \\ 0 & 0 & s_{a,z} \end{pmatrix} \tag{4.6b}$$

In this implementation, the biases for accelerometers and gyroscopes are modelled as *random walk* and the scale factors as *random constant*. The chosen type is of interest in the stochastic model of the Kalman filter implementation (section 4.1.3), because a random constant has zero variance.

The matrices $\boldsymbol{F}_{ij}$ from 4.5 are sub-matrices of the system matrix $\boldsymbol{F}$ and are listed below.

$$\boldsymbol{F}_{\dot{r}r} = \begin{pmatrix} 0 & 0 & -\frac{v_N}{(M+h)^2} \\ \frac{v_E * sin\phi}{(N+h)cos^2\phi} & 0 & -\frac{v_E}{(N+h)^2 cos\phi} \\ 0 & 0 & 0 \end{pmatrix} \tag{4.7}$$

$$\boldsymbol{F}_{\dot{r}v} = \begin{pmatrix} \frac{1}{M+h} & 0 & 0 \\ 0 & \frac{1}{(N+h)cos\phi} & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{4.8}$$

$$\boldsymbol{F}_{\dot{v}r} = \begin{pmatrix} -2\omega_e v_E cos\phi - \frac{v_E^2}{(N+h)cos\phi} & 0 & \frac{v_N v_D}{(M+h)^2} + \frac{v_E^2 tan\phi}{(N+h)^2} \\ -2\omega_e v_D sin\phi + 2\omega_e v_N cos\phi + \frac{v_E v_N}{(N+h)cos^2\phi} & 0 & -\frac{v_E v_D}{(N+h)^2} - \frac{v_N v_E tan\phi}{(N+h)^2} \\ 2\omega_e v_E sin\phi & 0 & \frac{v_E^2}{(N+h)^2} + \frac{v_N^2}{(M+h)^2} - \frac{2\gamma}{\sqrt{MN}+h} \end{pmatrix} \tag{4.9}$$

$$\boldsymbol{F}_{\dot{v}v} = \begin{pmatrix} \frac{v_D}{M+h} & -2\omega_e sin\phi - \frac{2v_E tan\phi}{N+h} & \frac{v_N}{M+h} \\ 2\omega_e sin\phi + \frac{v_E tan\phi}{N+h} & \frac{v_D}{N+h} + \frac{v_N tan\phi}{N+h} & 2\omega_e cos\phi + \frac{v_E}{N+h} \\ -\frac{2v_N}{M+h} & -2\omega_e cos\phi - \frac{2v_E}{N+h} & 0 \end{pmatrix} \tag{4.10}$$

$$\boldsymbol{F}_{\dot{e}r} = \begin{pmatrix} -\omega_e sin\phi & 0 & -\frac{v_N}{(N+h)^2} \\ 0 & 0 & \frac{v_N}{(M+h)^2} \\ -\omega_e cos\phi - \frac{v_E}{(N+h)cos^2\phi} & 0 & \frac{v_E tan\phi}{(N+h)^2} \end{pmatrix} \tag{4.11}$$

$$\boldsymbol{F}_{\dot{e}v} = \begin{pmatrix} 0 & \frac{1}{N+h} & 0 \\ -\frac{1}{M+h} & 0 & 0 \\ 0 & -\frac{tan\phi}{N+h} & 0 \end{pmatrix} \tag{4.12}$$

$$\boldsymbol{f}^l = \begin{pmatrix} 0 & -f_D & f_E \\ f_D & 0 & -f_N \\ -f_E & f_N & 0 \end{pmatrix} \tag{4.13}$$

$$\boldsymbol{\omega}_{il}^l = \begin{pmatrix} 0 & -\omega_e sin\phi - \frac{v_E tan\phi}{N+h} & \frac{v_N}{M+h} \\ \omega_e sin\phi + \frac{v_E tan\phi}{N+h} & 0 & \omega_e cos\phi + \frac{v_E}{N+h} \\ -\frac{v_N}{M+h} & -\omega_e cos\phi - \frac{v_E}{N+h} & 0 \end{pmatrix} \tag{4.14}$$

Finally, the system matrix $\boldsymbol{F}$ can be composed as shown in formula 4.15.

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{F}_{\dot{r}r} & \boldsymbol{F}_{\dot{r}v} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{F}_{\dot{v}r} & \boldsymbol{F}_{\dot{v}v} & -\boldsymbol{f}^l & \boldsymbol{R}_b^l & \boldsymbol{R}_b^l \boldsymbol{f}_b & \mathbf{0} & \mathbf{0} \\ \boldsymbol{F}_{\dot{\epsilon}r} & \boldsymbol{F}_{\dot{\epsilon}v} & -\boldsymbol{\omega}_{il}^l & \mathbf{0} & \mathbf{0} & -\boldsymbol{R}_b^l & -\boldsymbol{R}_b^l \boldsymbol{\omega}_b \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{4.15}$$

The system transition matrix $\boldsymbol{T}_{k,k+1}$ is now computed by power-series expansion of the system matrix $\boldsymbol{F}$ (formula 3.16) by implying only the linear part ($i = 1$).

$$\boldsymbol{T}_{k,k+1} = \boldsymbol{I} + \boldsymbol{F}\Delta t \tag{4.16}$$

A prerequisite for the elimination of higher order terms is that the state variables are correspondingly small. It can also be seen that the system equations require measurements ($\boldsymbol{f}_b$, $\boldsymbol{\omega}_b$) for the system update. In the absence of IMU measurements the system cannot be updated any more, which is different from the classical Kalman filter. The big advantage of this approach is that the measurement equations are particularly simple [Blauensteiner, 2008].

In the system model, the system noise term generally describes the uncertainties of modelling the dynamic system behaviour (see equation 3.14). In this implementation the system noise vector $\boldsymbol{w}$ consists of the following unmodeled residual errors in position $\delta\boldsymbol{r}^e$, velocity $\delta\boldsymbol{v}^l$, attitude $\delta\boldsymbol{\epsilon}^l$, gravity model $\delta\boldsymbol{g}^l$ and IMU sensor error model $\delta\boldsymbol{b}_{a,g}^b$, $\delta\boldsymbol{s}_{a,g}^b$. Each of this residual error terms have the dimension of three, which subsequently leads to a total dimension of 24 for the system noise vector.

$$\boldsymbol{w} = \begin{pmatrix} \delta\boldsymbol{r}^e & \delta\boldsymbol{v}^l & \delta\boldsymbol{\epsilon}^l & \delta\boldsymbol{g}^l & \delta\boldsymbol{b}_a^b & \delta\boldsymbol{s}_a^b & \delta\boldsymbol{b}_g^b & \delta\boldsymbol{s}_g^b \end{pmatrix}^T \tag{4.17}$$

The corresponding noise coefficient matrix is represented below and has the dimension of 21×24:

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{F}_{\dot{r}v} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{4.18}$$

4.1.2 Measurement Model

The measurement vector of the IMU-GNSS integration describes the differences of the GNSS position and velocity and of the IMU derived position and velocity.

$$\boldsymbol{l} = \begin{pmatrix} \boldsymbol{r}^e_{GNSS} - \boldsymbol{r}^e_{INS} \\ \boldsymbol{v}^l_{GNSS} - \boldsymbol{v}^l_{INS} \end{pmatrix} \tag{4.19}$$

Since in this implementation the position is given in ellipsoidal coordinates, the differences in latitude and longitude in radians are extremely small. Therefore, in order to receive meters as units these two observations are transformed to the local-level frame by using formula 2.8. Thus, the measurement vector changes to:

$$\boldsymbol{l} = \begin{pmatrix} (M+h)\delta\phi \\ (N+h)cos\phi\delta\lambda \\ \delta h \\ \delta v_N \\ \delta v_E \\ \delta v_D \end{pmatrix} \tag{4.20}$$

Finally, the design matrix $\boldsymbol{A}$ is given by:

$$\boldsymbol{A} = \begin{pmatrix} (M+h) & 0 & 0 & 0 & 0 & 0 \\ 0 & (N+h)cos\phi & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.21}$$

4.1.3 Stochastic Model

In the stochastic model of the system and measurement equations, the uncertainty of the models will be taken into account. The variance information of the states are included in $\boldsymbol{\Sigma}_{xx}$ (see formula 4.22).

$$\boldsymbol{\Sigma}_{xx} = diag(\boldsymbol{\sigma}_r^2 \quad \boldsymbol{\sigma}_v^2 \quad \boldsymbol{\sigma}_\epsilon^2 \quad \boldsymbol{\sigma}_{b,a}^2 \quad \boldsymbol{\sigma}_{s,a}^2 \quad \boldsymbol{\sigma}_{b,g}^2 \quad \boldsymbol{\sigma}_{s,g}^2) \tag{4.22}$$

When starting the Kalman filter procedure, the variance information is delivered from the initialization step. Further, it is propagated in the time update phase using formula 3.20. The covariance matrix of the system noise $\boldsymbol{\Sigma}_{ww}$ includes the uncertainty of the system noise. It must be noted that the system noise variables are modelled as random walk, except for the scale factors of accelerometers and gyroscopes. These are modelled as a random constant. Therefore, the variance of the scale factors is zero.

$$\boldsymbol{\Sigma}_{ww} = diag(\boldsymbol{\sigma}_{\delta r}^2 \quad \boldsymbol{\sigma}_{\delta v}^2 \quad \boldsymbol{\sigma}_{\delta \epsilon}^2 \quad \boldsymbol{\sigma}_{\delta g}^2 \quad \boldsymbol{\sigma}_{\delta b,a}^2 \quad \mathbf{0} \quad \boldsymbol{\sigma}_{\delta b,g}^2 \quad \mathbf{0}) \tag{4.23}$$

The system noise matrix, which is requested at each time update at a time update is derived with the following formula 4.24.

$$\boldsymbol{Q} = \boldsymbol{S}\boldsymbol{\Sigma}_{ww}\boldsymbol{S}^T \tag{4.24}$$

The measurement covariance matrix $\mathbf{\Sigma}_{ll}$ is derived at every measurement update from the GNSS positioning process (see section 2.2).

$$\mathbf{\Sigma}_{ll} = diag(\sigma^2_{\delta\phi} \quad \sigma^2_{\delta\lambda} \quad \sigma^2_{\delta h} \quad \sigma^2_{\delta v_N} \quad \sigma^2_{\delta v_E} \quad \sigma^2_{\delta v_D}) \tag{4.25}$$

5 Preprocessing of the IMU Data

This chapter deals with analysing and preprocessing of the inertial observables. The raw inertial data is analysed and synthetic noise is added. Afterwards, the inertial data is preprocessed by filtering with the de-noising technique from section 3.1.3. These results are used in chapter 6 to analyse the multi-sensor performance.

5.1 Evaluation of the IMU Data

In real inertial navigation applications the IMU sensor outputs (acceleration and angular rates) contain the following signals:

- Vehicle motions
- Earth gravity components
- Sensor biases
- Sensor noise
- Other effects (e.g. engine vibrations)

It can be seen that the vehicle motion information wanted is superimposed by other components. The gravity component must be known and is considered in the navigation equation. Sensor biases can be determined in a calibration step, or estimated in real-time by using stochastic process models and a sensor integration. The other components are hard to split. Sensor noise, which has white noise characteristics, and other random processes like engine vibration are expected to disturb the measurement especially in the high frequency component. Therefore, if such noise components could be removed from the measurements, improvements in inertial navigation are expected and in turn, the overall

navigation accuracy will be improved [Nassar, 2003].

Wavelet techniques, which were presented in chapter 3.1.3, can be applied for removing the high frequency noise. Undesirable effects of sensor noise and other high frequency disturbances will be minimized.

5.1.1 *IMUWaveletDenoising*: Developed IMU Analysing Tool in Matlab

In order to detect and remove undesirable effects in the IMU data, the tool *IMUWavelet-Denoising* was developed during the course of this thesis. This tool was developed in *MATLAB* and can be used to analyse and de-noise IMU measurements. The user interface of *IMUWaveletDenoising* is presented in figure 5.1.
The tool consists of four main features:

(1) Import data: The tool imports IMU measurements from a csv-file where each line represents measurements at a certain time. Each line must contain 7 elements:

tab. 5.1: IMU file structure

time	accX	accY	accZ	gyroX	gyroY	gyroZ

With *importing data*, it is also possible to downsample the IMU measurements. IMUs are available which record at a very high data rate (e.g. 1000 Hz) and such a bulk of data is rarely required. Therefore, in order to decrease the file size, it is possible to downsample the IMU data.

(2) Analysing: To detect undesirable effects in the IMU data, the tool enables the possibility to analyse the measurements. It is possible to plot the *amplitude frequency spectrum*, *power spectral density* (*PSD*) or the *time series* for the different IMU measurements. This feature is subsequently used to analyse the characteristics of the different IMU measurements in section 5.1.2.

(3) Wavelet de-noising: *Wavelet thresholding*, which was presented in chapter 3.1.3, is applied to remove the noise term from the measurements. The user can select the

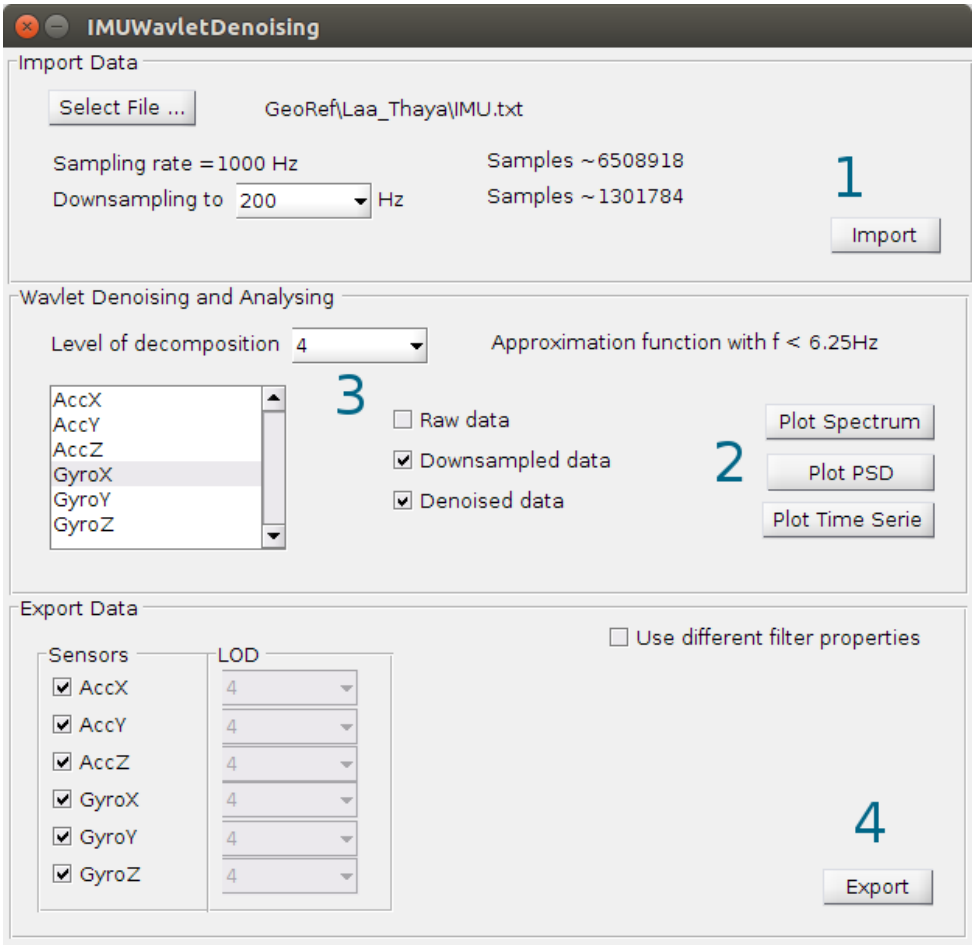


fig. 5.1: User interface of the developed analysing tool *IMUWaveletDenoising*

wavelet decomposition level and take a look on the filtering result by using the analysing plot options from (2).

(4) Export data: After analysing the measurements and choosing the appropriate level of decomposition for the filtering step, the filtered data can be exported. A csv-file with the chosen sampling frequency from (1) is created.

The tool *IMUWaveletDenoising* was used to analyse the different IMU datasets in section 5.1.2 and in addition, the preprocessing of the IMU data in chapter 5.2 was also performed with the help of this tool.

5.1.2 Raw Data

For this thesis, three data sets from different IMU measurements are analysed. Each of these datasets covers time series of accelerometer and gyroscope measurements with an high update rate. The following list gives a short overview of the data available:

- Helicopter flight *Laa an der Thaya*

  IMU: iNAV-FJI-001

  Data rate: 1000 Hz

  Data length: 100 min

- Helicopter flight *Lanzenkirchen*

  IMU: iNAV-FJI-001

  Data rate: 1000 Hz

  Data length: 150 min

- Train ride *ÖBB* from Vienna to Breclav

  IMU: AEROcontrol IMU-IIf

  Data rate: 400 Hz

  Data length: 4 h

As the different datasets show similar behaviour, just one dataset is chosen to be presented here. The following diagrams are related to the helicopter flight *Laa an der Thaya*. Figure 5.2 displays the accelerometer time series in x-direction, while figure 5.3 shows the respective gyroscope measurements. The data can be split into two parts.

- *static*: IMU is in rest relative to the earth surface. The data recorded include earth gravity, earth rotation and noise.
- *kinematic*: IMU is in motion. The data recorded include vehicle dynamics, earth gravity, earth rotation and noise.

During the first ten to twenty minutes of each measurement, only static data are recorded and subsequently used to determine the initial orientation (see section 2.1.4). The reason for the offset in the static accelerometer time series ($f_{x,static} \neq 0$) is because leveling of the IMU was imperfect. Therefore, a part of the local gravity is measured in x-direction.

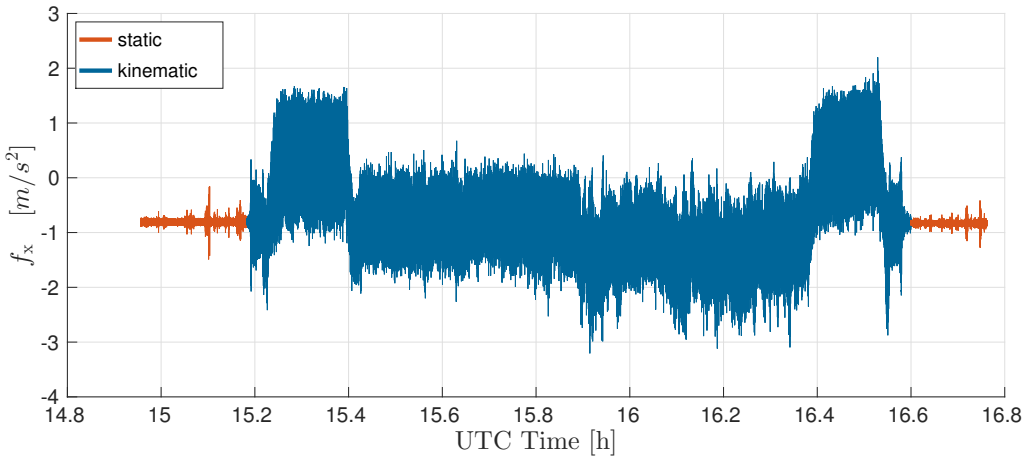


fig. 5.2: Accelerometer measurements in x-direction

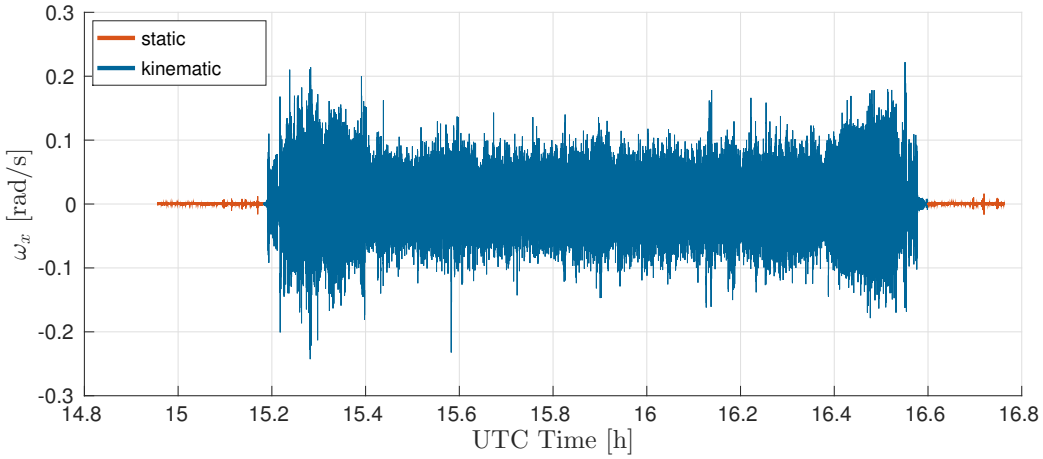
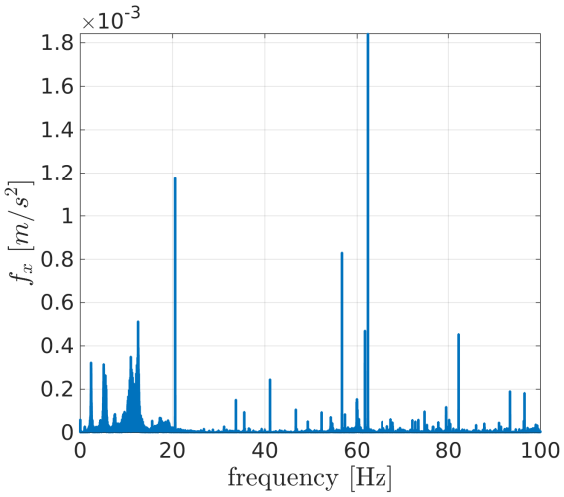


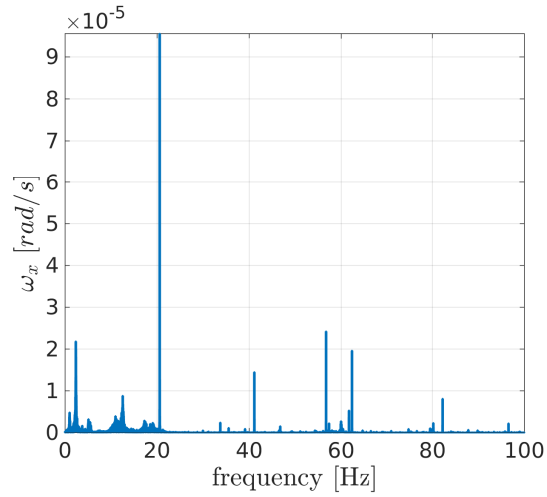
fig. 5.3: Gyroscope measurements in x-direction

In order to obtain information whether dominant frequency components are present in the time series a Fourier transform is computed to display the amplitude frequency spectrum. As one aim of this thesis is to analyse the signal's noise characteristics and to try to remove them, it is of interest which frequencies are included in the static and kinematic data. Figure 5.4 shows the frequency spectrum of ten minutes of static data in x-direction and z-direction. Referring to figure 2.1, the x-axis of the IMU points in front of the helicopter and the z-axis points downwards. Some pronounced peaks at certain frequencies are visible. One reason for this could be residual vibrations of the platform which the IMU was mounted onto and which were not compensated by the shock absorbers. In addition,

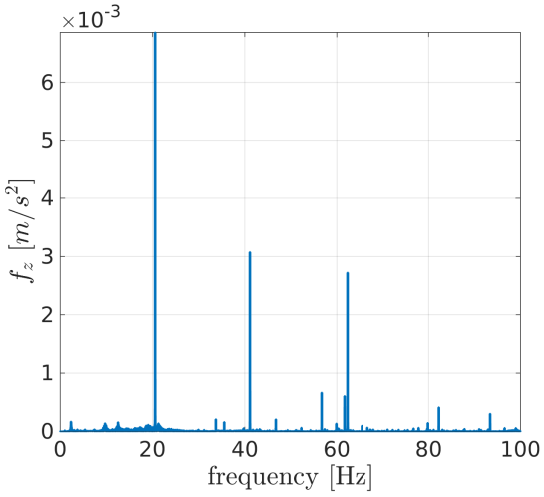
it seems that the peaks appear periodically at about 20, 40 and 60 Hz (especially visible in figure 5.4c). This constant frequency offset may be caused by an aliasing effect. The white noise term is not very dominant in these measurements.



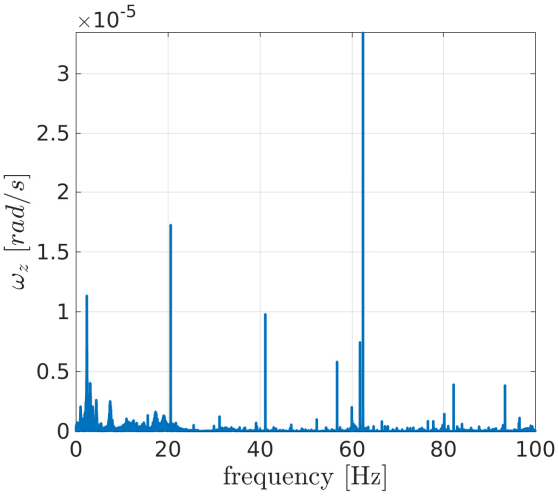
(a) Accelerometer measurements in x-direction



(b) Gyroscope measurments in x-direction



(c) Accelerometer measurements in z-direction

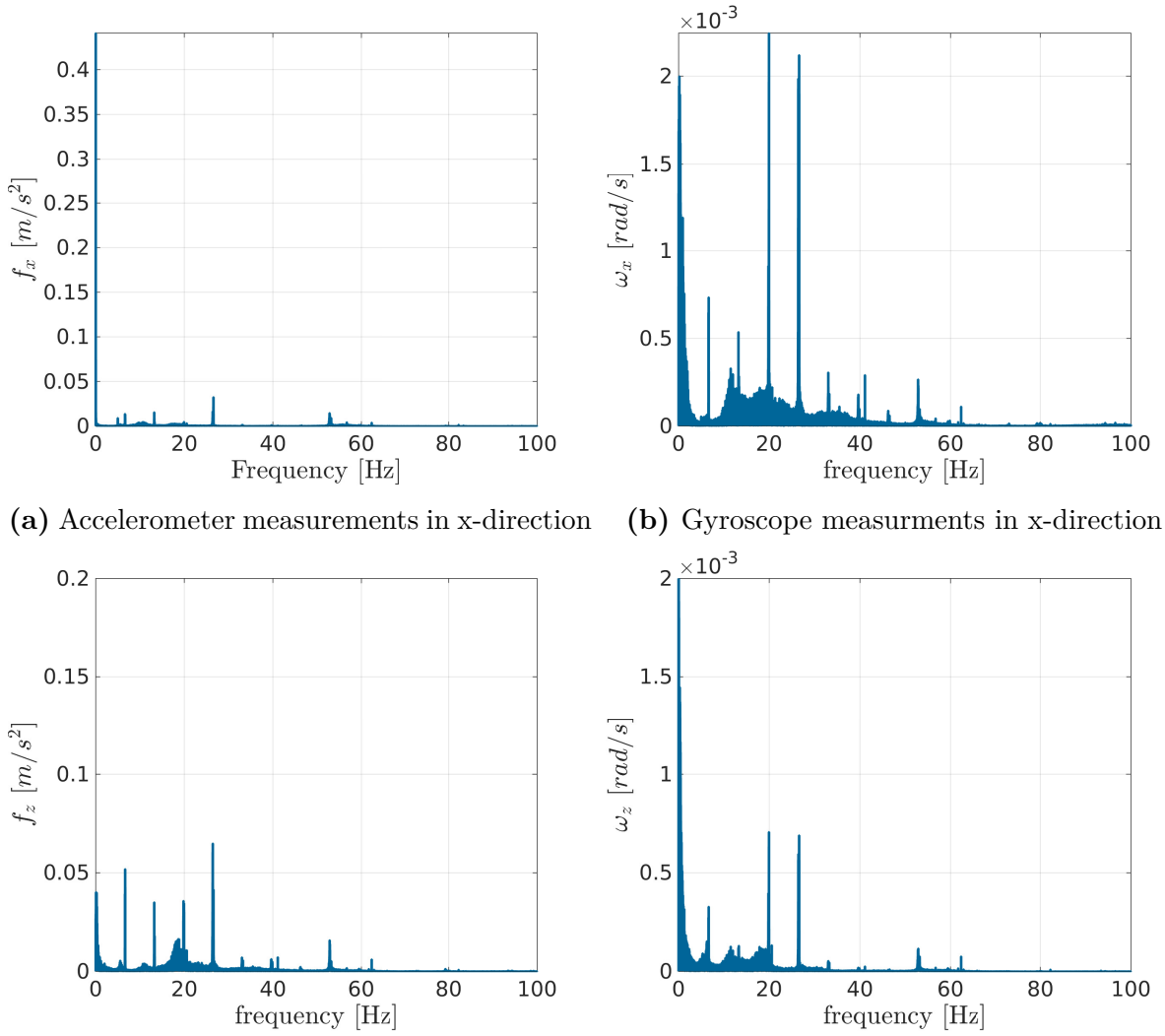


(d) Gyroscope measurments in z-direction

fig. 5.4: Frequency spectrum of static data of *Laa an der Thaya*

Within the frequency spectrum of the kinematic test data in figure 5.5 the bandwidth containing the majority of the motion dynamics falls into the low frequency band of the spectrum (< 7 Hz). An additional remark is that the disturbing frequency at 20 Hz is still present in the data (especially noticeable at gyroscope measurements) and in addition, a dominant frequency component at 26 Hz occurs now in the kinematic data set. This dominant frequency is not present in the spectrum of the static time series and appears only when the helicopter is in motion. The peaks at about 10 to 40 Hz are most probably

due to the vibrations of the engine or rotor motion. How such undesirable high-frequency components can be removed was introduced in chapter 3.1.3 and this technique is used to eliminate the noise from the data set in chapter 5.2.



(a) Accelerometer measurements in x-direction **(b)** Gyroscope measurments in x-direction

(c) Accelerometer measurements in z-direction **(d)** Gyroscope measurments in z-direction

fig. 5.5: Frequency spectrum of kinematic data of *Laa an der Thaya*

The temporal progress and the amplitude spectrum above are just shown for acceleration and gyroscope measurements in x-direction and z-direction, but the spectrum characteristics in y-direction of the IMU sensors are quite similar.

5.1.3 Adding Synthetic Noise

In the last section, the recorded raw data sets were presented. Some undesirable effects, which may come from engine or rotor vibration, are made visible. One point of interest is the following: Does this error source influence our multi-sensor system performance? Another approach is whether we can still deliver a useful navigation solution if the noise sources increase? To make this analysable, the measurements will be overlaid with synthetic noise. Two noise sources are chosen to be added to the original data. First, it is an important fact that low-grade IMUs have bad signal-to-noise ratio and suffer from extreme white noise. Thus, it is interesting to find out how the navigation solution will behave if the original measurements are overlaid with white noise. This means, a low-grade IMU is simulated. Secondly, it is worthwhile to know what will happen if vibrations of the engine increase. Therefore, a synthetic disturbing source at a certain frequency band will be added.

White Noise

In this thesis, medium-grade IMUs are used. The signal-to-noise ratio is quite good, compared to a low-grade IMU. With low-grade IMUs the expected signal is buried behind white noise. A white noise process is a random process of random variables that are uncorrelated over time, have zero mean and finite variance. Formally, w is a white noise process if:

$$E(w_t) = 0, \quad var(w_t) = \sigma^2, \qquad cov(w_t, w_s) = 0 \quad for \quad t \neq s \tag{5.1}$$

Normally, a white noise process is present over the whole frequency band. It is of interest how the navigation performance will behave if the signal-to-noise ratio of our IMU output is low. This can be evaluated by adding white noise to the original data. This is graphically demonstrated in figure 5.6. The magnitude of the noise is about one tenth of the maximally occurring amplitude in the frequency spectrum of the static raw data.
Figure 5.7 represents the noisy data set of static accelerometer and gyroscope measurements in x-direction.

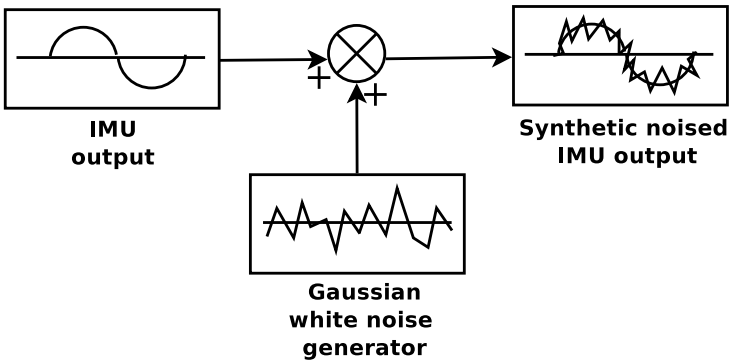
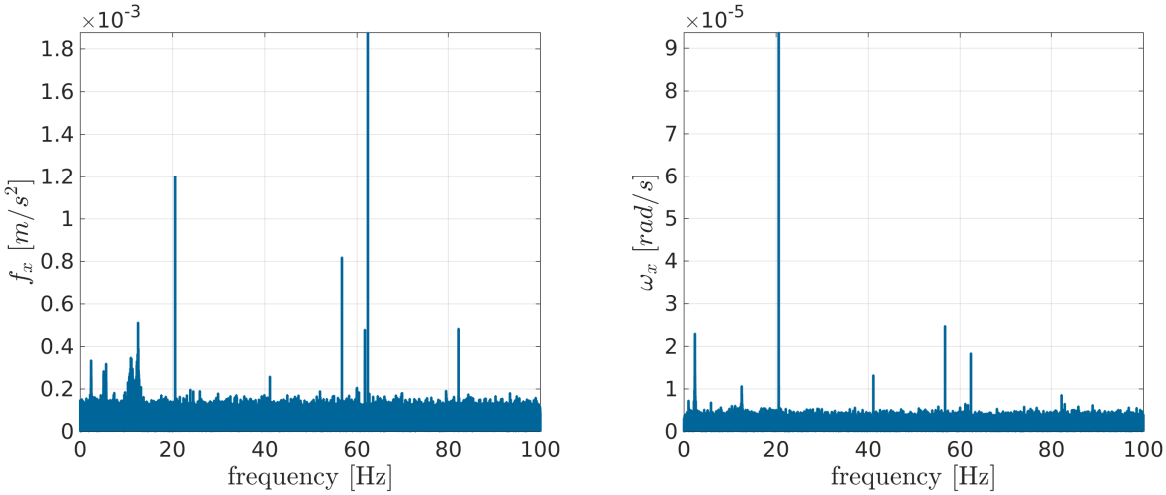


fig. 5.6: Adding synthetic white noise



(a) Accelerometer measurements in x-direction **(b)** Gyroscope measurments in x-direction

fig. 5.7: Spectrum of static accelerometer and gyro data with added white noise

Band-Limited Noise

In the frequency spectra 5.4 and 5.5 also a prominent disturbing signal at 20 Hz is visible. This is most probably due to engine or rotor vibration. This noise source will now be increased in order to make the influence more demonstrable. Therefore, the original signal will be overlaid with an oscillating noise at a defined frequency band. Figure 5.8 shows the IMU output superimposed with generated band-limited noise. One question deals with the scaling of the noise amplitudes. In order to generate realistic noise, the scaling of the noise amplitude depends on the static raw data. Therefore, the amplitude of the added noise adhere to the maximally occurring amplitude in the frequency spectrum of the static raw data (see figure 5.4). The maximum amplitude of the noise is twice of the original maximum amplitude. To implement a band-pass, a hamming window is used to cut out the frequency region wanted from the generated noise (20-22 Hz). This remaining

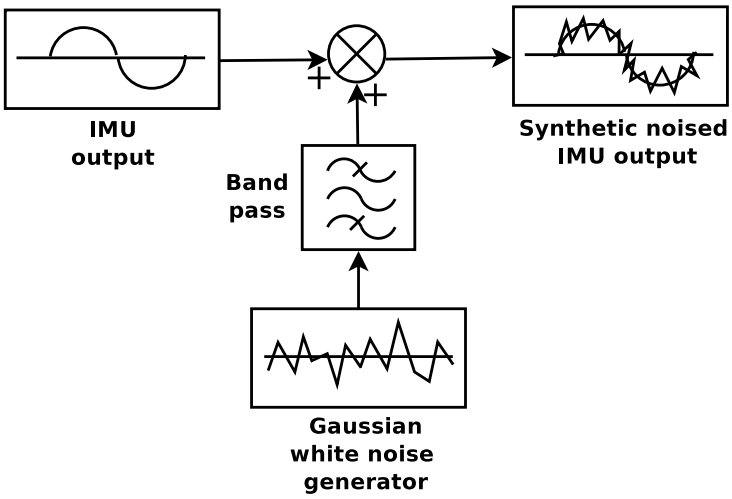
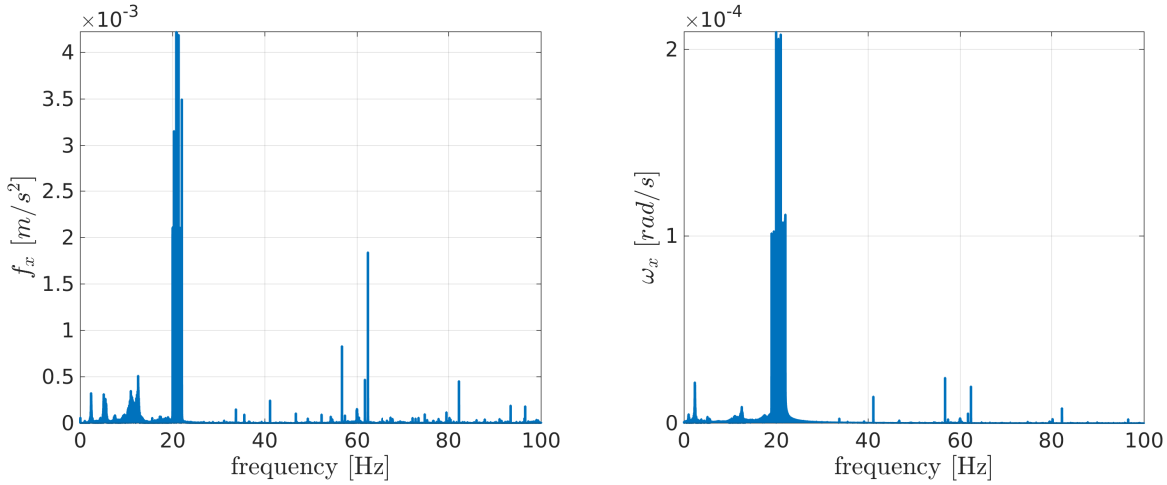


fig. 5.8: Adding synthetic white noise

frequencies are subsequently added to the static data. For example, figure 5.9 shows the spectrum of static accelerometer and gyroscope data with added synthetic noise. The superimposed noise is visible just between 20 and 22 Hz.



(a) Accelerometer measurements in x-direction **(b)** Gyroscope measurments in x-direction

fig. 5.9: Spectrum of static gyro data with added band-limited noise

In this section, the IMU output was overlaid with noise to deteriorate inertial signal quality. These measurements with superimposed synthetic noise are subsequently used to analyse the performance of the navigation system. How the noise affects the initial alignment is shown in chapter 6.1 and the behaviour at GNSS outages is presented in chapter 6.2. But, in practice, it is more important to improve the IMU output, therefore the next chapter deals with removing the noise component from the inertial data.

5.2 Denoising of the IMU Data

Inertial data suffer from different noise sources like sensor error noise, engine vibration or even white noise. If such noise components can be removed as well as possible, the overall navigation solution is expected to improve. This chapter deals with de-noising inertial data with *wavelet thresholding*, which was introduced in section 3.1.3.
First, it will be explained why wavelet thresholding has been chosen for de-noising inertial data instead of the classical low-pass filter techniques.

5.2.1 Wavelet Thresholding vs. Low-Pass Filter for Inertial Data

It was mentioned in chapter 5.1 that vehicle motion is superimposed by other signals like earth gravity, sensor noise or other effects like engine vibration. The aim is to remove the noise from the inertial data without affecting the wanted components. Classic low-pass filters have the characteristics to remove all frequencies above a certain frequency. If important information exists in the high-frequency component, it will be filtered out, which results in a loss of information. An example of inertial navigation are sudden changes in the vehicle accelerations. As it is shown in Kang et al. [2011] the low-pass filter distorts this change of the signal. As visible in figure 5.10 the red dashed line presenting the low-pass filtered data leads to a signal distortion. On the other hand, de-noised data using the wavelet threshold technique reduces the noise level with hardly any distortion (green line).
Therefore, using low-pass filters is not the best choice for processing IMU data. Instead, the approach *wavelet decomposition* and *thresholding* presented in chapter 3.1.3 overcomes the shortages of the dynamic behaviour of low-pass filters. Therefore, this filter technique is used in this thesis. It must be noted that the selection of an appropriate level of decomposition is based on removing the high-frequency noise while keeping all the useful information (especially the vehicle motion) contained in the signal. Therefore, the level of decomposition has to be chosen wisely.

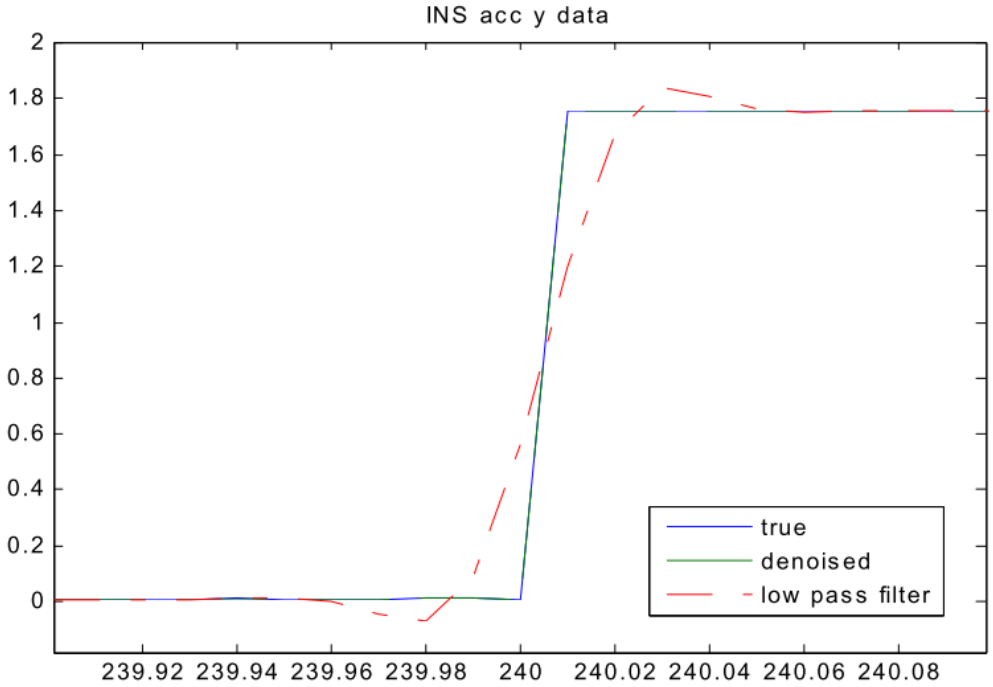


fig. 5.10: Change in signal using wavelet denoising and low-pass filter [Kang et al., 2011]

5.2.2 Selection of the Wavelet Level of Decomposition

In chapter 3.1.3, a technique to split a signal into two parts was presented. This is done by passing the signal through two complementary filters. The result is a low-frequency (*approximation*) and a high-frequency (*detail*) part. In a next step, the approximation part is further divided into two parts. The decomposition is done iteratively, so that the inertial sensor measurement is broken down into many lower-resolution components. Each level of decomposition has a maximally occurring frequency, which can be calculated with formula 3.8. The maximally occurring frequency for each level of decomposition of the data set available can be seen in table 5.2.

The question is: What is the appropriate level of decomposition? In practice, a suitable number of levels based on the nature of the signal would be selected. According to De Agostino [2008], in the case of static inertial data, the sensor outputs contain the physical components (earth gravity or earth rotation rate) and the sensor long-term errors (such as biases). For kinematic inertial data, effects of the actual vehicle motion dynamics and the sensor noise as well as some other effects, such as vehicle engine vibrations, are recorded. This is why the criterion for the selection of the appropriate level of decomposition for de-noising is distinguished for the static and kinematic cases in the following section.

tab. 5.2: Level of decomposition: Maximal frequency component

	Laa an der Thaya	**Lanzenkirchen**	**ÖBB**
	frequency [Hz]	frequency [Hz]	frequency [Hz]
Sample rate	1000	1000	400
LOD 0	500	500	200
LOD 1	250	250	100
LOD 2	125	125	50
LOD 3	62.5	62.5	25
LOD 4	31.25	31.25	12.5
LOD 5	15.625	15.625	6.25
LOD 6	7.813	7.813	3.125
LOD 7	3.906	3.906	1.563
LOD 8	1.953	1.953	0.781
LOD 9	0.977	0.977	0.391
LOD 10	0.488	0.488	0.195
LOD 11	0.244	0.244	0.975

Static Inertial Data

El-Sheimy et al. [2004] suggested that the cut-off frequency of static IMU data should be 0.5 Hz. The earth rotation rate and gravity as well as the long-term inertial sensor errors should be completely included in this approximation part. Wavelet decomposition is not able to separate the earth-related components from the long-term sensor errors. But El-Sheimy et al. [2004] also mentioned that such long-term errors have minor effects on the performance of inertial sensors during the alignment process. As a next step in this thesis, it is proved if this strict cut-off frequency can also be used for the IMU measurements available. Therefore, the signal-to-noise ratio (SNR) is calculated depending on the level of decomposition. The signal wanted is assumed to be the average of the measured static IMU data, because the earth rotation rate and earth gravity vector should remain constant in the static phase. The SNR at each filter level is calculated for the raw data set and the synthetically noised data set of *Laa an der Thaya* (see section 5.1.3). It is expected that at each filter level the SNR will improve.

$$SNR\ [dB] = 10 * log_{10}\left(\frac{signal}{noise}\right) \tag{5.2}$$

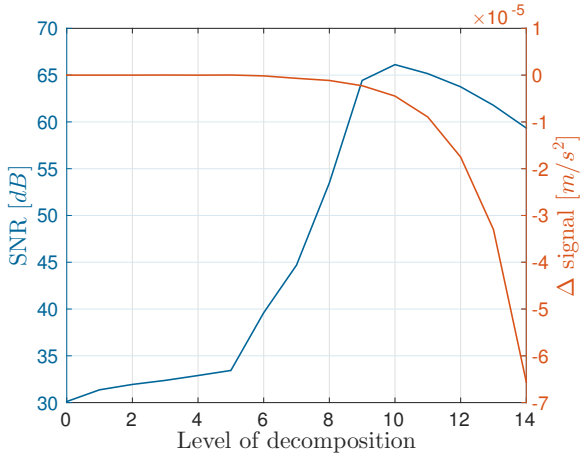
with: $$signal = \frac{1}{n}\sum_{i=1}^{n}(static\ raw\ data)$$

$$noise = (filtered\ static\ data)_{LOD} - \frac{1}{n}\sum_{i=1}^{n}(static\ raw\ data)$$

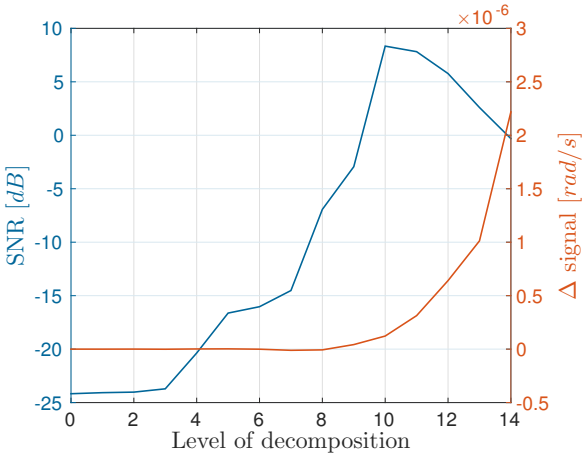
On the other hand, care has to be taken that the filter technique does not affect the original signal. Therefore, the difference of the mean value of the static raw data and the mean value of the wavelet filtered data at each decomposition level is calculated.

$$\Delta signal = \frac{1}{n}\sum_{i=1}^{n}(filtered\ static\ data)_{LOD} - \frac{1}{n}\sum_{i=1}^{n}(static\ raw\ data) \tag{5.3}$$

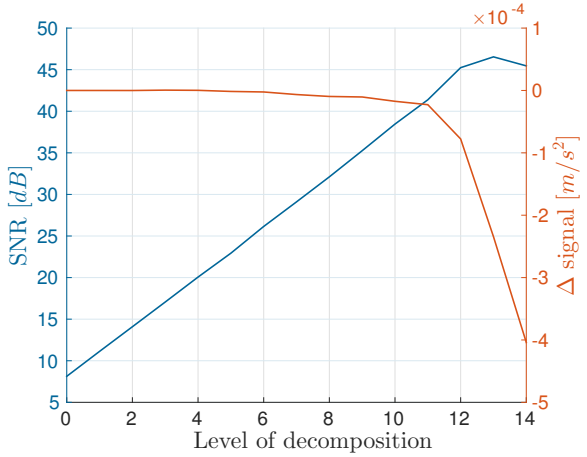
When the filter process affects the signal, the calculated mean value of the filtered data should differ from the value of the original data. The behaviour of the SNR and the mean of the static data are shown in figure 5.11.
What is striking in figure 5.11 is that the SNR improves with each decomposition level (blue line). This is expected, because wavelet tresholding should remove the noise term. When looking at the SNR progress of the raw data (5.11a, 5.11b) it can be seen that the maximum is at a decomposition level of ten. In table 5.2 the LOD of ten corresponds to 0.488 Hz. This value meets the cut-off frequency of 0.5 Hz proposed by El-Sheimy et al. [2004]. But on the other hand, after nine to ten levels of decomposition the value of the difference between the original mean and the filtered mean increases (red line). This increasing offset demonstrates a deterioration of the physical signal measured, which should be avoided. At the data set with synthetic white noise added (5.11c, 5.11d), also a deterioration of the signal is expected after a LOD of ten. With band-limited noise added (5.11e, 5.11f), the best performance is expected after seven levels of decomposition.



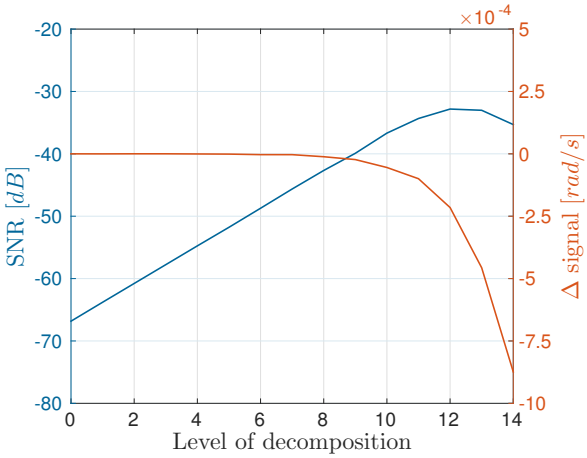
(a) Accelerometer measurement



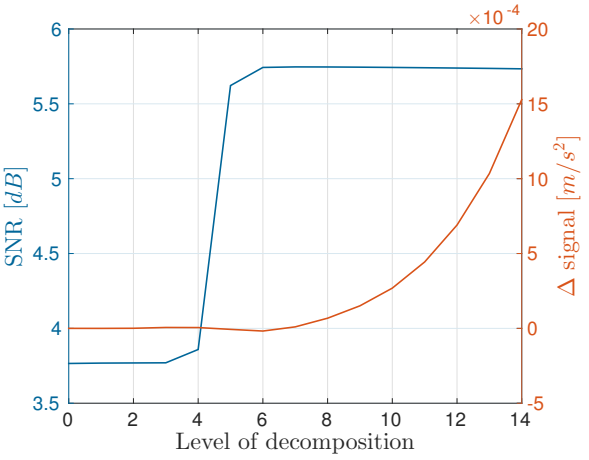
(b) Gyroscope measurement



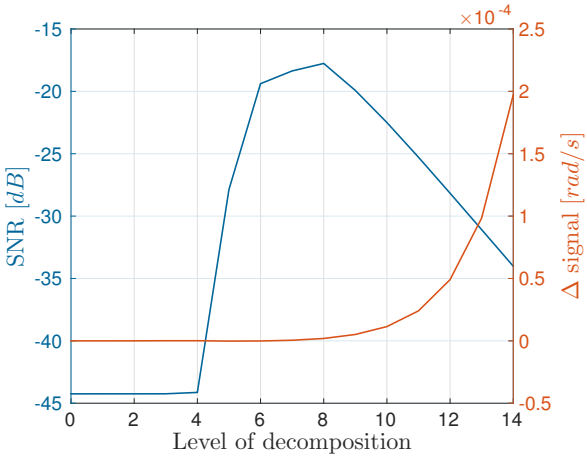
(c) Accelerometer measurement with synthetic white noise



(d) Gyroscope measurement with synthetic white noise

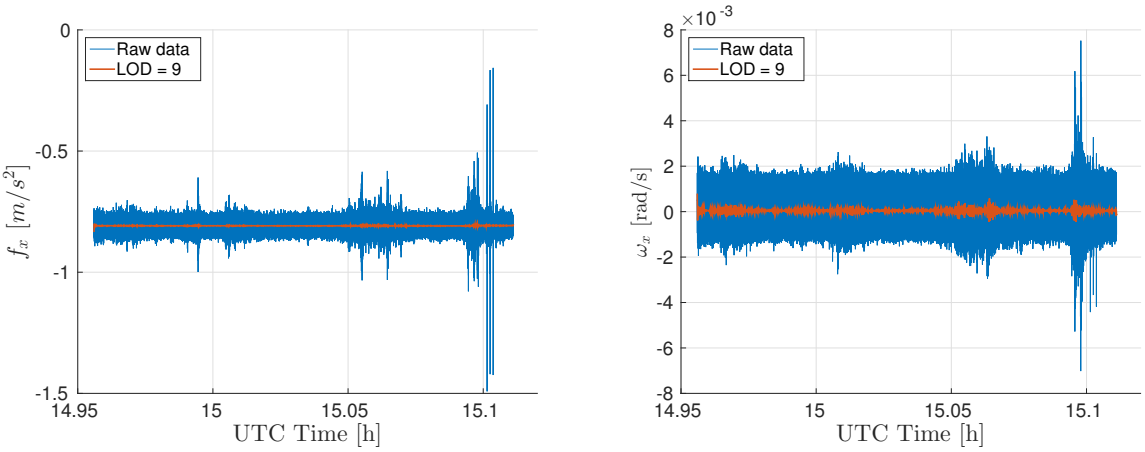


(e) Accelerometer measurement with band-limited noise



(f) Gyroscope measurement with band-limited noise

fig. 5.11: Calculated SNR and difference to the origin mean from data set *Laa an der Thaya*



(a) Accelerometer measurements **(b)** Gyroscope measurements

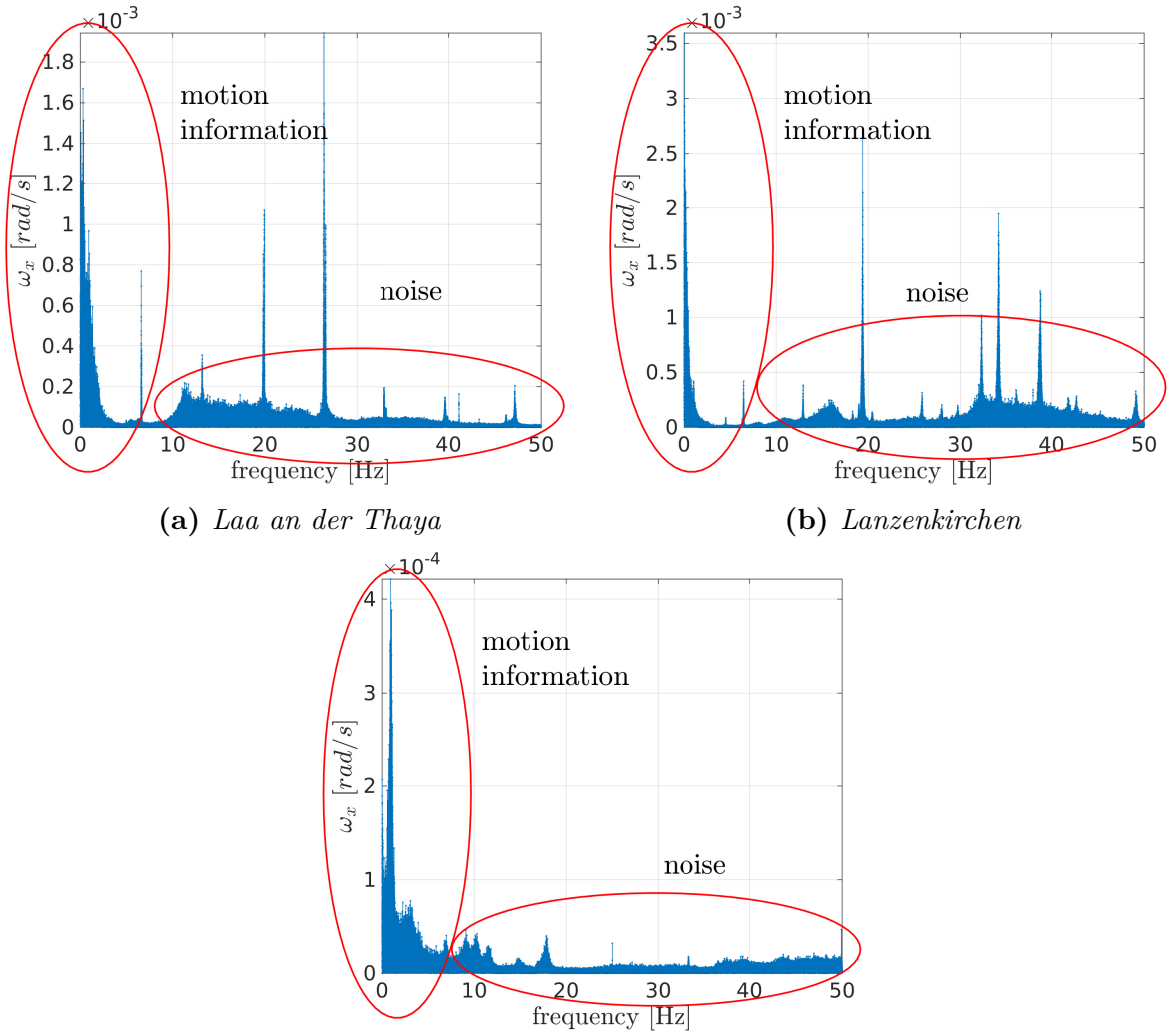
fig. 5.12: Static data of *Laa an der Thaya*

As an IMU de-noising example, figure 5.12 shows the signals measured of the static data of *Laa an der Thaya* in x-direction (blue signal). Furthermore, the wavelet de-noised signals after nine levels are shown (red signal). It is noticeable that nearly everything of the random noise term is removed and in addition, the peaks measured at time 15.1 h are almost totally mitigated. In section 6.1, it will be figured out if these filter values really improve the initialization phase of the multi-sensor system.

Kinematic Inertial Data

To obtain the cut-off frequency of kinematic data, a spectral analysis has to be performed to ensure that the decomposition process does not disturb any actual motion information. Figure 5.13 demonstrates gyroscope measurements in x-direction of the three data sets available. The frequency spectrum is just shown up to 50 Hz, because beyond that value there is just white noise. The figures clearly show that the motion term can be separated from the noise term visually. The cut-off frequency is at about 10 Hz. In the case of the *ÖBB* data, the value is a little bit higher, at about 15 Hz.

As soon as this spectral analysis has been completed and the approximate cut-off frequency is determined, information about the possible level of decomposition can be delivered. The appropriate level of decomposition for each data set can be found in table 5.3. As an exemplary result, figure 5.14 represents the filtered part of the gyroscope data after six decompositions.



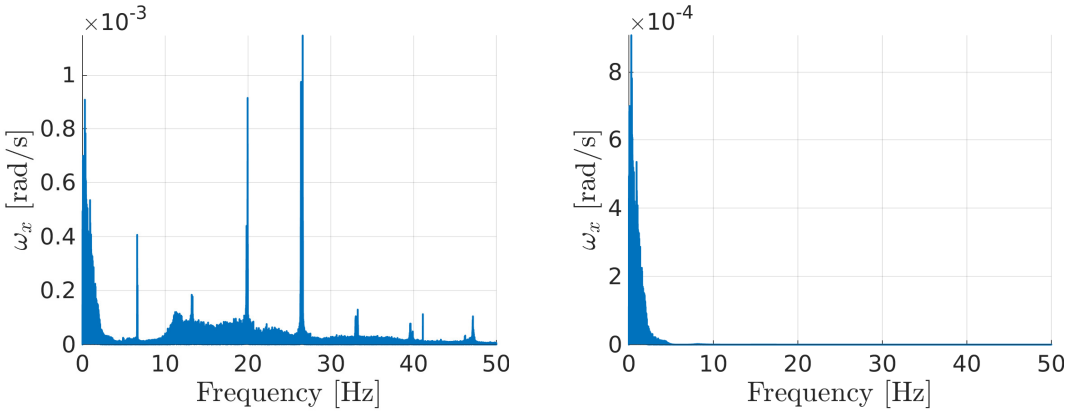
(a) *Laa an der Thaya* **(b)** *Lanzenkirchen*

(c) *ÖBB*

fig. 5.13: Amplitude spectrum of kinematic gyro measurements

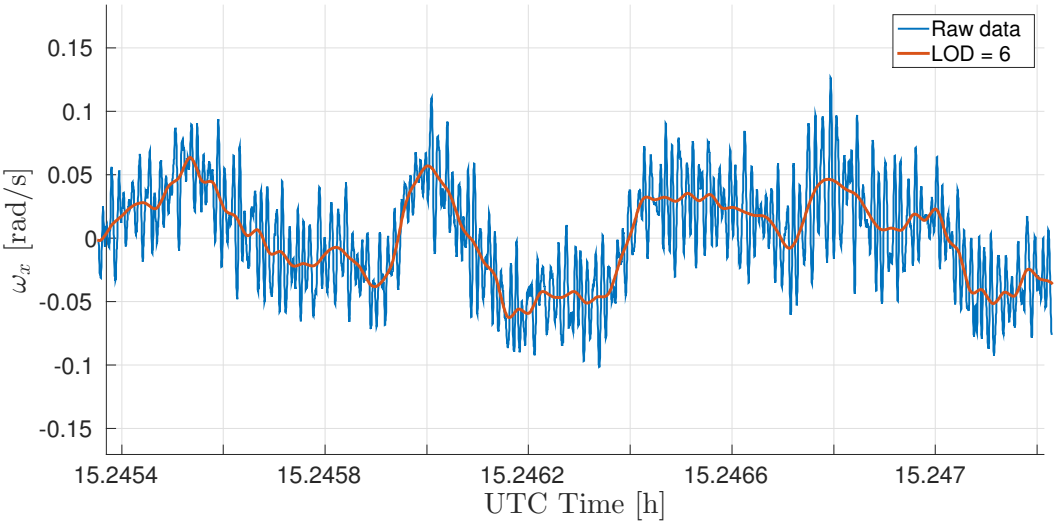
tab. 5.3: Possible level of decompostion

	Laa an der Thaya	**Lanzenkirchen**	**ÖBB**
static	LOD 10	LOD 10	LOD 8
kinematic	LOD 5-6	LOD 5-6	LOD 3-4



(a) Amplitude spectrum of the original data

(b) Amplitude spectrum after 6 decompositions



(c) Section of the temporal process

fig. 5.14: Gyroscope measurements of *Laa an der Thaya*

Of course, with this visual interpretation, it is not sure if the decomposition level chosen really improves the performance of the multi-sensor system, or if it influences the navigation performance negatively, because motion information is filtered out. Therefore, in chapter 6, it will be analysed how much the level of decomposition of the inertial data filtered influences the navigation performance.

6 Analyses and Results

In this chapter, the results of the analyses are presented. In section 3.1.3 a technique to decompose a signal was introduced. This procedure was implemented for the inertial measurements in chapter 5.2. It is now analysed in which way the de-noising technique influences the multi-sensor system performance. Therefore, the wavelet thresholding technique is used for the original and synthetically noised inertial data. Their effect is shown in two steps of the multi-sensor navigation process. Firstly, it is analysed how the noised IMU data affects the alignment process and subsequently, it is shown that the alignment procedure can be improved with the help of wavelet de-noising. The second analysing step takes the behaviour of the navigation system into account if GNSS updates are not available. It is of interest if the de-noising technique of the inertial data improves the overall position error. Figure 6.1 shows the navigation process with the analysis points chosen schematically.

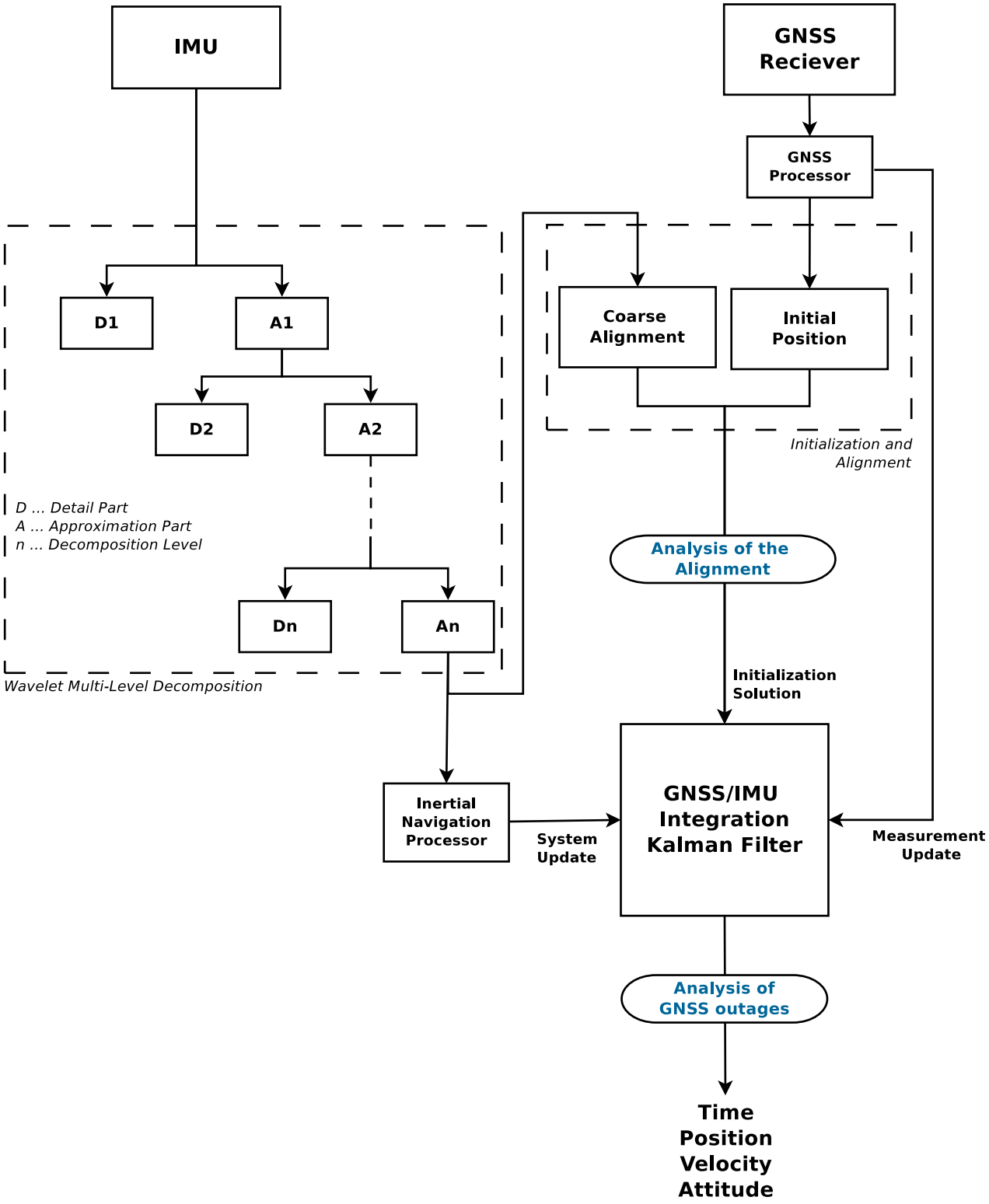


fig. 6.1: Block diagram of the modified multi-sensor system with the analysis points: *Analysis of the Alignment* and *Analysis of GNSS outages*

6.1 Analysis of the Alignment

A misalignment of the multi-sensor system can be an important error source for the overall navigation system. It was mentioned before in section 2.1.4 that during the alignment process an averaging of the calculated rotation matrix for some epochs has to be done. This leads to the situation that the high sensor noise and other periodic effects of the static data are reduced. The noise reduction depends on the time interval where static data is available. Usually, some minutes are required. To obtain a reference alignment, an alignment based on the raw data of the inertial data sets was performed. The mathematical background of the course alignment of IMU data was introduced in chapter 2.1.4. The final alignment orientation after 10 minutes of static data is calculated by using formula 2.13 and is presented in the following table (6.1).

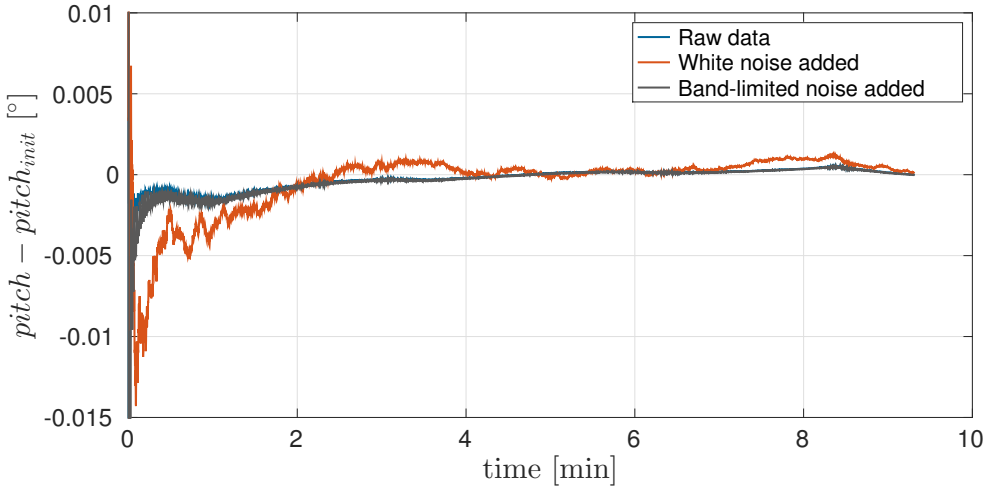
tab. 6.1: Computed initial alignment

	roll [°]	**pitch** [°]	**yaw** [°]
Laa an der Thaya	0.075	4.727	195.718
Lanzenkirchen	0.231	0.505	286.822
ÖBB	0.570	180.070	-45.483

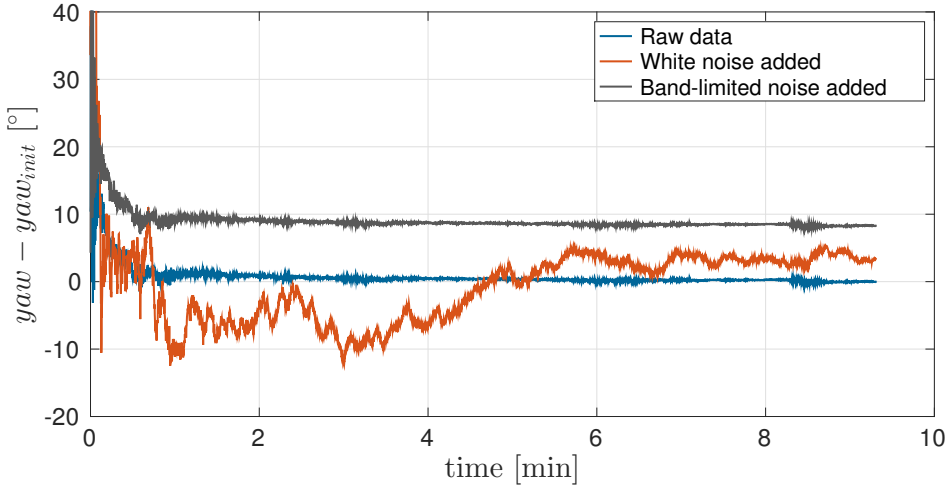
This initial alignment is now used as a reference. In the next section, it is shown how the added synthetic noise affects the result of the alignment.

6.1.1 Alignment with Noised Inertial Data

To demonstrate the effect of noise interfered inertial data at the initialization step, the course alignment is computed with the synthetically noised data shown in chapter 5.1.3. Due to the high sensor noise of the IMU, an averaging of the calculated rotation matrix has to be performed. The averaging process usually requires some minutes of static data. This averaging time is needed in order to reduce the noise components of the signal. Figure 6.2 points out how the choice of the averaging time of the initialization affects the final alignment. In this figure, the difference of the computed *pitch* and *yaw* angle of the noised data and the reference value (table 6.1) is shown. It is necessary to know that roll and pitch show similar behaviour and therefore, just the alignment of pitch is



(a) Alignment of the pitch angle



(b) Alignment of the yaw angle

fig. 6.2: Course alignment with the IMU observables of *Laa an der Thaya*

presented here. The reader should notice the different vertical axis scales in the pitch and yaw alignment progress.

In figure 6.2 it is demonstrated that ten minutes of course alignment process of the synthetically white noised data are not enough for the yaw alignment. The alignment of the Euler angle has not reached its final value. Some more minutes of recorded static data would be necessary. On the other hand, the alignment with added band-limited noise affects the final computed yaw angle. The band-limited noise influences the initialization process in such a way that the final yaw alignment differs from the reference alignment resulting in a wrong initialization. In this example, the difference of the final yaw angle and the initial one is 8.2°. In the case of roll or pitch alignment, no significant difference

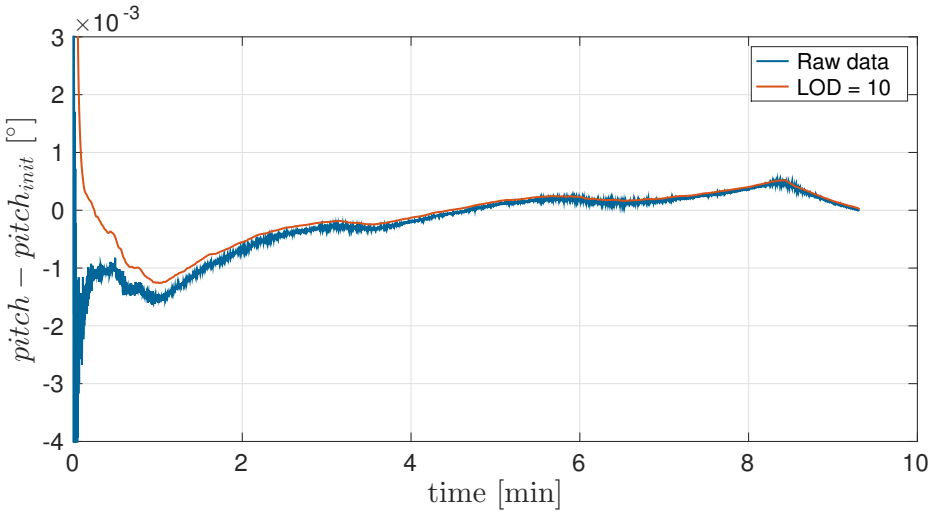
from the reference values can be observed.

Conclusion: The course alignment procedure is not suitable for data with a dominant noise term (low SNR). With this simple alignment procedure, the yaw angle cannot be determined in a sufficient accuracy and the averaging time would take more than 10 minutes to reach its final state. Therefore, for low-graded IMUs, the course alignment is not enough. The reason for this is that in this case usually gyroscopes with noise levels larger than the earth's rotation rate are used and as a result, low-graded IMUs cannot be aligned in static mode during the course alignment. Subsequently, either a technique called fine alignment or external yaw measurements has to be performed. External yaw measurements, which use a magnetic compass, provide the alignment (see pages 234-252 in [Jekeli, 2001]). A second remark is that in the case of inertial measurements with a band-limited noise term, the yaw angle converges to a wrong value. In this case, more averaging time would not eliminate the wrong alignment.

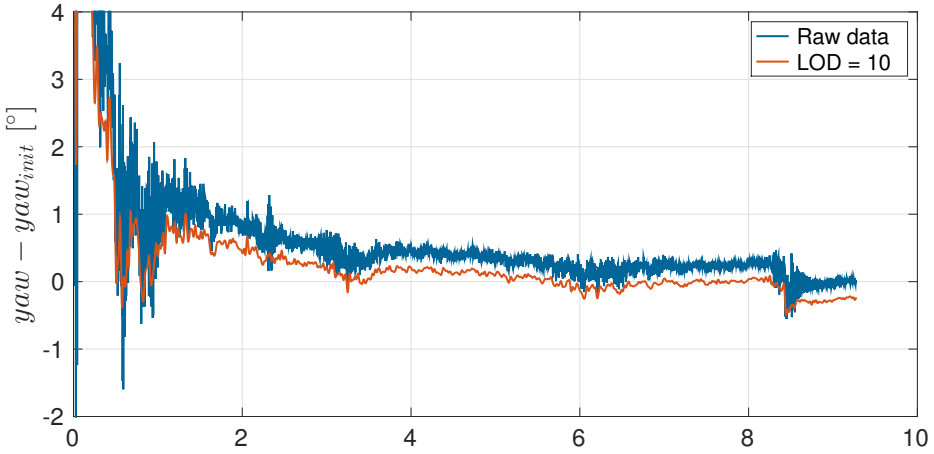
6.1.2 Alignment with Denoised Inertial Data

An improvement with regard to providing more accuracy and shortening the averaging time can be obtained by using wavelet de-noised inertial data. A strict cut-off frequency for static IMU data of 0.5 Hz was suggested in section 5.2.2, which leads to an approximation part of the signal with frequencies below 0.5 Hz. This approximation part should include only the earth-rotation rate and the local gravity component is used for the alignment process. Nearly all of the measurement noise which influenced the alignment process negatively should be eliminated.

a) Original inertial data: To prove whether the wavelet de-noising enhances the alignment result, the initialization of the multi-sensor system was performed with different filter steps. Ten levels of decomposition equal about 0.5 Hz (referring to table 5.2) and therefore, for example, figure 6.3 demonstrates that after de-noising with a level of decomposition of 10, the Euler angles align faster. This effect is especially noticeable in the yaw determination. A second interesting fact is that the final solution of the de-noised yaw alignment differs from the initial alignment solution at about 0.2 degree. A reason for this difference may be that the recorded inertial raw data is influenced by a noise source. It was mentioned and shown in section 5.1.2 that the Fourier transform of the static raw data points out a dominant frequency component at about 20 Hz. This error source is a



(a) Alignment of the pitch angle

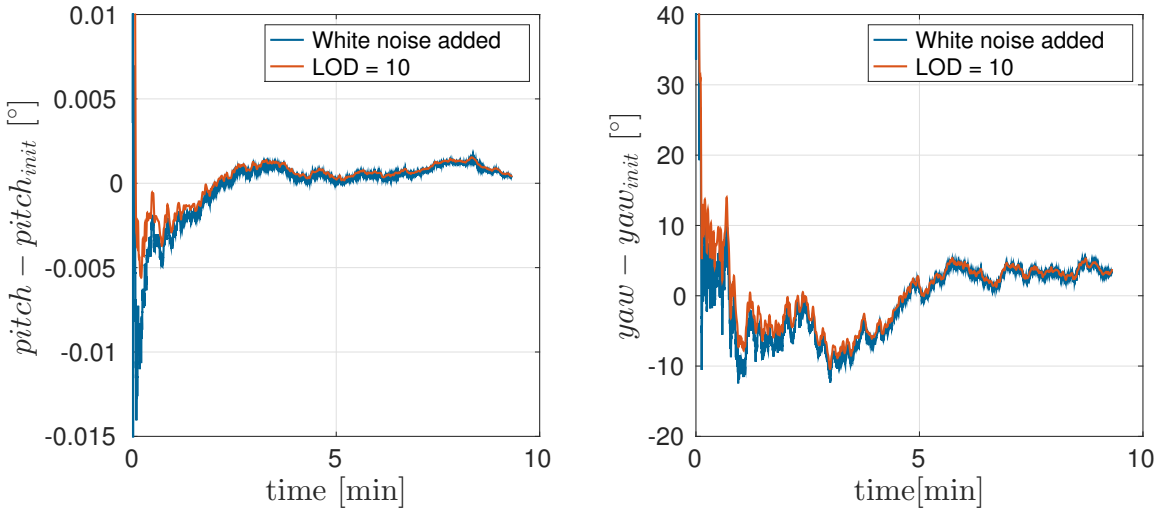


(b) Alignment of the yaw angle

fig. 6.3: Course alignment of *Laa an der Thaya* after ten levels of decomposition

kind of band-limited noise and it was shown in figure 6.2 that such noise leads to a different final alignment angle. At a decomposition level of ten, this noise is removed from the data and does not affect the alignment any more. Therefore, such noise characteristics might be responsible for the alignment differences.

b) Inertial data with added synthetic white noise: Additionally, it is of interest if the wavelet de-noising technique can remove the white noise term from the inertial data, which in turn should decrease the averaging time for the alignment procedure. Therefore, the inertial data with added synthetic white noise is de-noised with several levels of decomposition. It can be seen in figure 6.4 that after applying the wavelet filter,



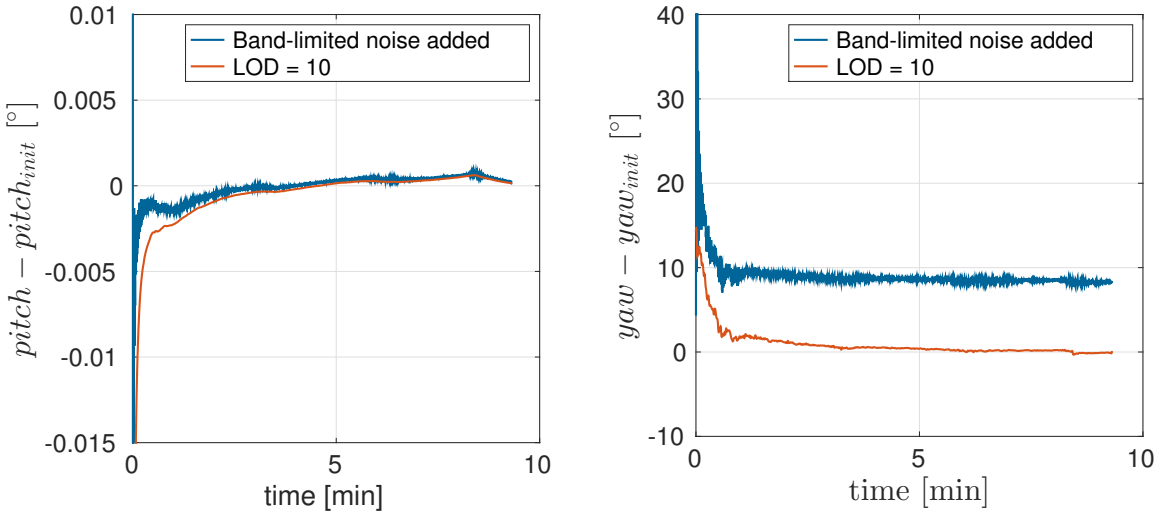
(a) Alignment of the pitch angle **(b)** Alignment of the yaw angle

fig. 6.4: Course alignment of *Laa an der Thaya* with added white noise

the Euler angles do not converge faster. For example, after ten decompositions the final state of the alignment is still not reached. The reason for this is that the white noise term influences information all over the whole frequency bandwidth. Therefore, the earth's rotation rate and gravity component are also superimposed by noise and the alignment needs more time to converge to its final state. More minutes of static measurements would be required in order to increase the averaging time and subsequently reduce the influence of the noise component. The de-noising filter has the effect that the alignment process is smoother.

c) Inertial data with added synthetic band-limited noise: As inertial data with a band-limited noise term is present, the wavelet de-noising technique should remove the noise term without manipulating the original information, which in turn should remove the large yaw alignment difference between the inertial raw data and noised data (cf. figure 6.2). Figure 6.5 presents that after ten levels of decomposition, the band-limited term is totally removed and the result of the alignment of the synthetically noised data equals the initial alignment from table 6.1.

Conclusion: The analysis of the course alignment of the raw inertial data has shown that the measurements recorded were influenced by a noise component at a certain frequency, which can be removed by applying wavelet thresholding. In the case of a huge

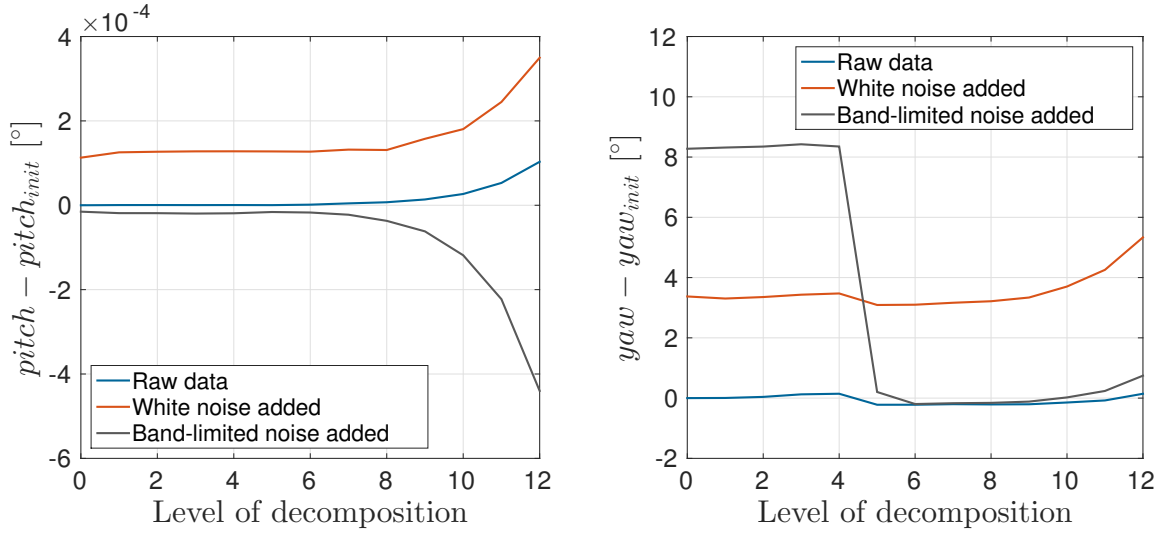


(a) Alignment of the pitch angle **(b)** Alignment of the yaw angle

fig. 6.5: Course alignment of *Laa an der Thaya* with added band-limited noise

white noise term in the inertial data, the convergence time stays the same, but the random noise level is lower, which leads to a smoother curve progression. The information wanted (earth's rotation rate and gravity component) is buried behind the white noise component, too, and this cannot be separated from the superimposed noise. If the residual noise component is too dominant, no improvement of shortening the convergence time can be obtained. Additionally, it was shown that disturbing noise which occurs at a certain frequency band can be removed by de-noising the inertial data.

As a final remark, it is of interest how the course alignment depends on the wavelet level of decomposition. Therefore, the course alignment is performed with different levels of decomposition (figure 6.6). It is obvious that the chosen decomposition level has nearly no impact in the case of inertial data with the added white noise term. The difference to the initial value stays nearly constant at each decomposition level. Compared to figures 5.11c / 5.11d, where the SNR increases at each decomposition step, the final alignment angle is not affected by the lower temporal resolution of the signal. Therefore, the earth's components cannot be separated from the noise any more. But the improved SNR might have advantages in the case of the convergence time when using a Kalman filter, when subsequently a fine alignment or kinematic alignment takes place. In the case of band-limited noise, a significant improvement of the difference of the yaw angle can be achieved after five levels of decomposition. After the fifth level, the signal is broken down to a lower resolution signal where the disturbing frequency band is excluded. When compared



(a) Difference of pitch angle

(b) Difference of the yaw angle

fig. 6.6: Differences of the alignment of *Laa an der Thaya* data after certain wavelet LODs

to the SNR pre-analysis shown in figure 5.11e / 5.11f, this result was expected, because the SNR increases rapidly after five decomposition levels. Additionally, it can be seen that after ten levels of decomposition, the difference of the alignment starts to increase. The reason for this is that with further decompositions, the filter procedure influences the earth's component signal wanted. It may be that the cut-off frequency, suggested by El-Sheimy et al. [2004], of 0.5 Hz is too strict.

6.1.3 Effect of an Inaccurate Alignment

The last section dealt with an analysis of the alignment of the multi-sensor navigation system and how undesirable effects like noise influence the course alignment solution. Partly, the undesirable noise can be removed or at least reduced by wavelet thresholding. In this section, the effect of an inaccurate initialization alignment is analysed. It is shown how the further navigation solution is affected by an inaccurate initial alignment. It was shown in section 2.1.4 that the determination of *yaw* is difficult and therefore more inaccurate, when compared to *roll* and *pitch*. Therefore, it is analysed how a wrong initialization of the yaw angle affects the navigation information, which is estimated by the IMU-GNSS Kalman filter. Table 6.2 represents the yaw alignment error used for this analysis.

tab. 6.2: Yaw alignment error

	[°]	[°]	[°]	[°]
initial yaw_{error}	0.2	1	4	8

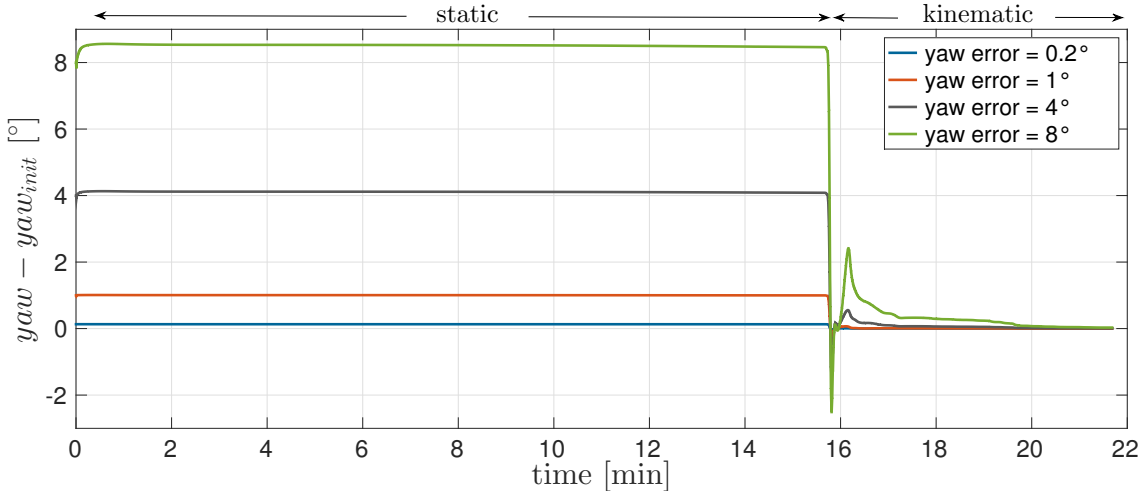


fig. 6.7: Difference of yaw with wrong initialization

If no GNSS updates are available in the multi-sensor navigation system, the system operates as a *INS stand-alone* system. In such a case the navigation system cannot eliminate the orientation error derived from the initial alignment. The alignment error is carried on to the trajectory estimation. As in the present multi-sensor system GNSS is available as a supporting sensor, a *kinematic alignment* takes place after the vehicle object starts moving. This leads in turn to an improvement of the vehicle alignment as long as the vehicle is moving. The following figure 6.7 shows the influence of the wrong initial yaw alignment. In the *static phase* (the vehicle is in rest), no GNSS based improvement of the wrong alignment can be achieved. The initial yaw error remains the same. As soon as the vehicle starts moving (*kinematic phase*), the GNSS measurements are used to support the current alignment of the vehicle, which in turn leads to a reduction of the yaw alignment error with time.

The length of the convergence time till the yaw error is eliminated depends on the initial alignment error and the GNSS measurement update rate. In this example, GNSS positions are available every 0.2 second. Table 6.3 shows the time till the yaw error falls below 0.1 degrees after the vehicle has started moving (*kinematic alignment*).

tab. 6.3: Convergence time until $|yaw - yaw_{init}| < 0.1°$

	[°]	[°]	[°]	[°]
initial yaw_{error}	0.2	1	4	8
convergence time [s]	3.7	10.2	74.4	240.7

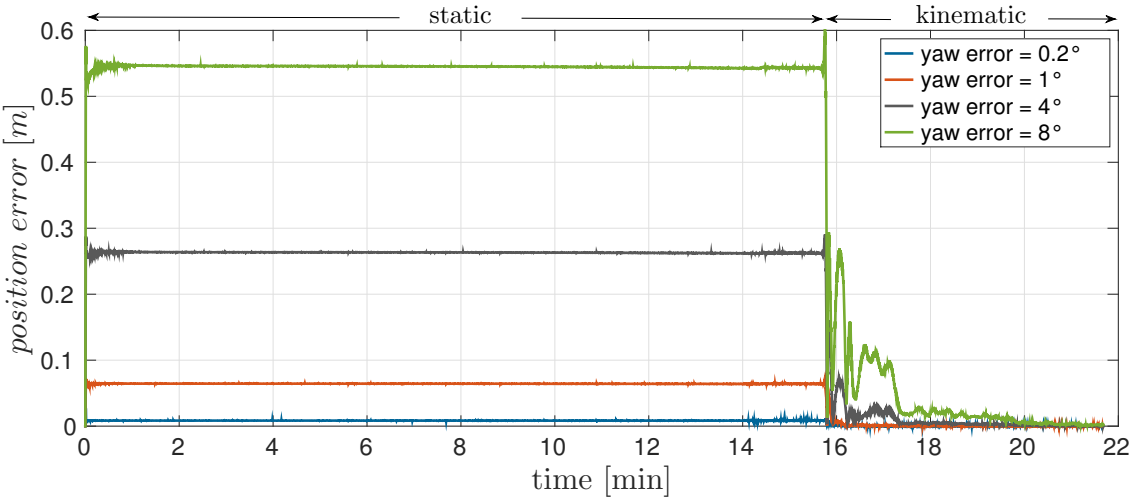


fig. 6.8: Position error after wrong initialization

The wrong yaw initialization angle does not only affect the orientation of a vehicle, but also subsequently the trajectory estimation. Figure 6.8 indicates the position error in the static phase and after the vehicle has started moving.

One thing that is striking is that a wrong initial alignment directly influences the position error. A yaw error of eight degrees causes a position error in the navigation frame of more than half a meter if the initial position is obtained using GNSS measurements. The reason for this error is the so-called *level-arm*. Normally, the IMU and GNSS receiver are two separate measurement systems and have different spatial positions. The spatial vector which describes the distance between IMU and GNSS in the body-system is called level-arm. The level-arm of the data set *Laa an der Thaya* is as follows (6.1).

$$\Delta \boldsymbol{r}^b = \begin{pmatrix} 3.727\ m \\ 0.141\ m \\ -0.592\ m \end{pmatrix} \tag{6.1}$$

The yaw angle describes the rotation of the body frame around the vertical axis with respect to the local-level frame (see section 2.1.1 about coordinate frames). If yaw is wrongly determined and includes a yaw error $\delta\alpha$, this error causes a rotation of the level-arm $\Delta\boldsymbol{r}^b$.

$$\Delta\boldsymbol{r}^b_{wrong} = \boldsymbol{R}_z(\delta\alpha)\Delta\boldsymbol{r}^b \tag{6.2a}$$

$$with: \qquad \boldsymbol{R}_z(\delta\alpha) = \begin{pmatrix} cos(\delta\alpha) & sin(\delta\alpha) & 0 \\ -sin(\delta\alpha) & cos(\delta\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.2b}$$

This rotation of the level-arm due to an inaccurate yaw estimation leads to the following position error $\delta\boldsymbol{r}$.

$$\delta\boldsymbol{r} = \sqrt{\left(\Delta\boldsymbol{r}^b_{wrong} - \Delta\boldsymbol{r}^b\right)^2} \tag{6.3}$$

Table 6.4 summarizes the position error due to a wrong yaw estimation.

tab. 6.4: Position error in static phase

	[°]	[°]	[°]	[°]
yaw_{error}	0.2	1	4	8
position error [cm]	1.3	6.5	26.0	52.0

As soon as the vehicle starts moving, the yaw determination improves but also the position, which can be seen in figure 6.8. Table 6.5, depicts the period after which the position error falls below 5 cm due to a wrong yaw estimation.

An error in the yaw determination causes a rotation around the z-axis of the body frame. Normally the z-axis of the body frame points down, which subsequently leads to a horizontal position error. Figure 6.9 indicates that especially latitude and longitude are affected by the yaw error.

tab. 6.5: Time until position error < 5 cm

	[°]	[°]	[°]	[°]
initial yaw_{error}	0.2	1	4	8
time [s]	0	6	29.4	93.6

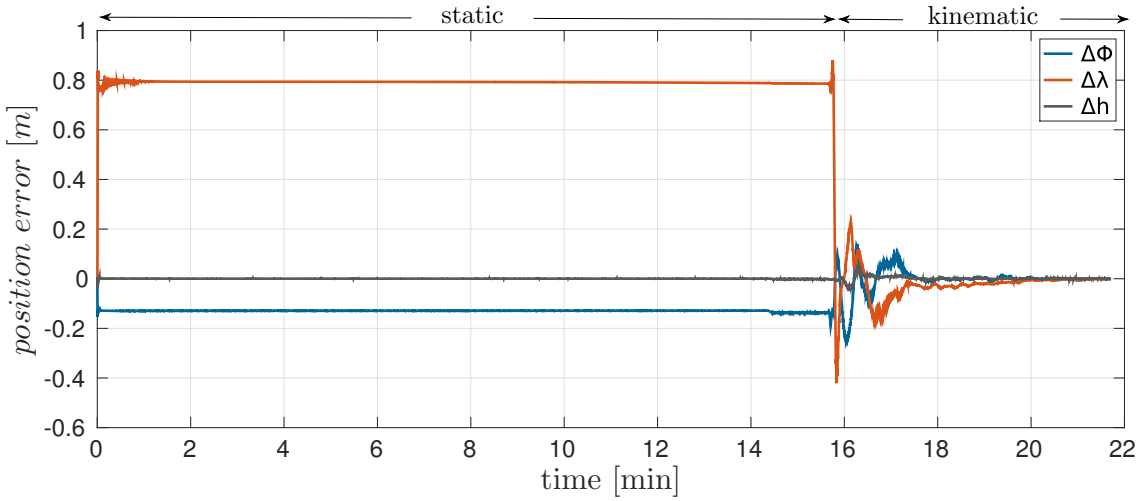


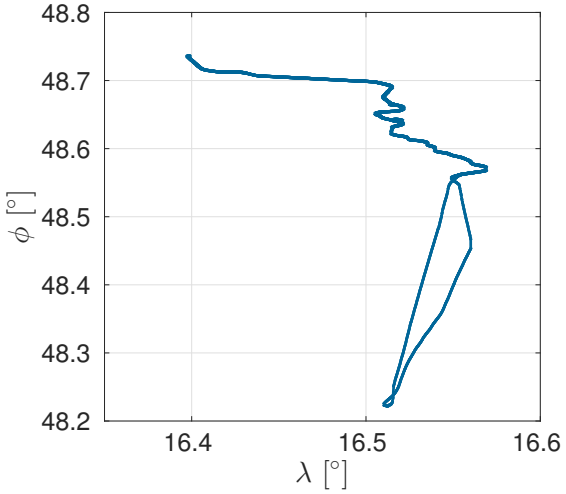
fig. 6.9: Position error after wrong initialization

Conclusion: It was shown that the convergence time of the Euler angles in kinematic alignment depends on the error magnitude. But in addition, the convergence time depends on the interval of the GNSS updates, too. In this example, GNSS measurements were available every 0.2 seconds, which leads to a very dense measurement update. When the time between GNSS measurements increases, the convergence time will also increase. As the vehicle is in rest, the position error depends both on the alignment error and the level-arm. If the yaw estimation is inaccurate, the horizontal coordinates are affected most.

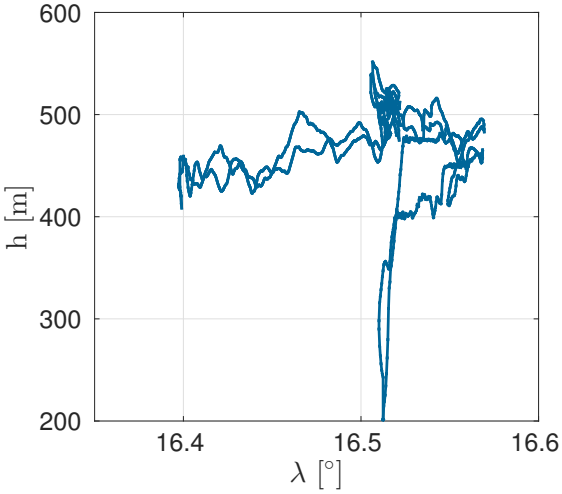
6.2 Simulating GNSS Outages

In this section, the usefulness of estimating the inertial sensor errors during the Kalman filter process is presented. It was shown in chapter 4 that the IMU-GNSS sensor integration was performed by a *loose coupled architecture* with a *closed-loop Kalman filter*. The closed-loop approach allows the Kalman filter to correct the system itself. At each iteration step, the modelled inertial sensor errors (see section 2.1.5) are estimated. At GNSS outages, no GNSS updates are possible, which means that GNSS does not aid the inertial sensor any more. The multi-sensor system operates as an INS stand-alone system. The navigation accuracy in such situations is mainly specified by the inertial sensor errors which were estimated in the Kalman filter process. Therefore, it is of interest at which accuracy the multi-sensor system can operate in such situations and, in addition, whether the wavelet filter technique can improve the navigation performance.

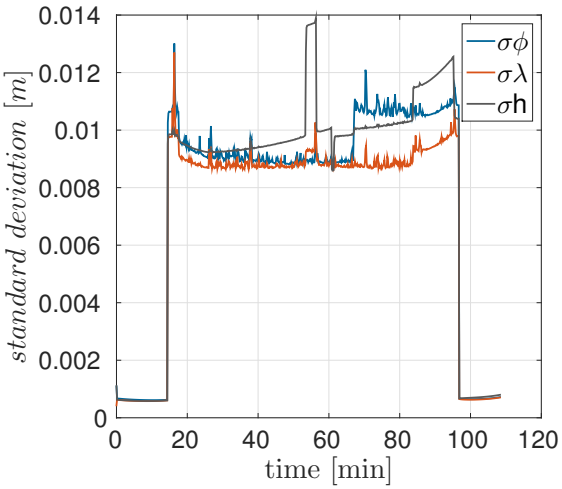
To achieve a reference, the trajectory of *Laa an der Thaya* is estimated using GNSS updates every 0.2 seconds and the original inertial data (without synthetic noise added). This is performed by using the sensor integration script which was developed by the research group *Advanced Geodesy* at *TU Wien*. The basic concept of the sensor fusion is explained in chapter 4. In figure 6.10, the estimated reference trajectory is shown in top and side view. Additionally, it can be seen that the standard deviation in latitude, longitude and height of the trajectory is between one to two centimeters. The reason for this precise position estimation is due to the fact that GNSS updates are available every 0.2 seconds and the trajectory is calculated in post-processing while fixing most of the GNSS ambiguities. The difference of the behaviour of the multi-sensor system compared to this estimated trajectory is shown further.



(a) Reference trajectory (top view)



(b) Reference trajectory (side view)



(c) Standard deviation of the estimated trajectory

fig. 6.10: Estimation of the trajectory *Laa an der Thaya*

It was mentioned before that a deeper look at the estimation of the inertial sensor errors is done in this section. This is accomplished by simulating GNSS outages, which means that GNSS does not aid the inertial navigation system any more and the performance of the navigation system is based on the last estimated sensor errors. It is expected that if these estimates were fairly appropriate, the resulting position errors at theses GNSS outages keep within limits. In total, 16 outages with the period of one minute each are simulated. Between each outage, there is a period of five minutes when GNSS measurements are available again, so that the navigation system can converge again. Therefore, the errors obtained during any outage are independent of the other outages. In figure 6.11, the resulting position errors at these simulated GNSS outages compared to the reference trajectory (6.10) are shown.

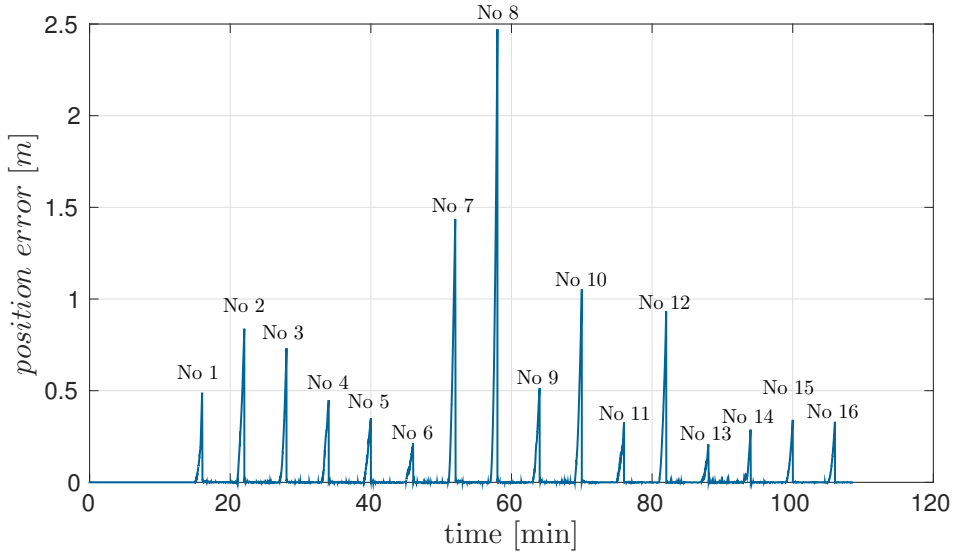


fig. 6.11: Position errors at sixteen simulated GNSS outages

The maximum resulting magnitude of the position error at the end of outage periods is about 2.5 meters. The average position error is 0.68 meter. Furthermore, before and after any GNSS outage, errors are reset almost to zero.

6.2.1 GNSS Outages With Noised Inertial Data

In the last section, a reference trajectory of the dataset *Laa an der Thaya* was computed. Additionally, sixteen GNSS outages with a period of one minute each were simulated. In this section, the effect of noisy inertial data at these GNSS outages is demonstrated.

Therefore, the trajectory of *Laa an der Thaya* is estimated with the synthetically noised inertial data from section 5.1.3. In addition, the same GNSS outages displayed in figure 6.11 are simulated. The resulting position error is calculated for added synthetic white noise (simulating a low-grade IMU) and added band-limited noise (simulating engine vibrations).

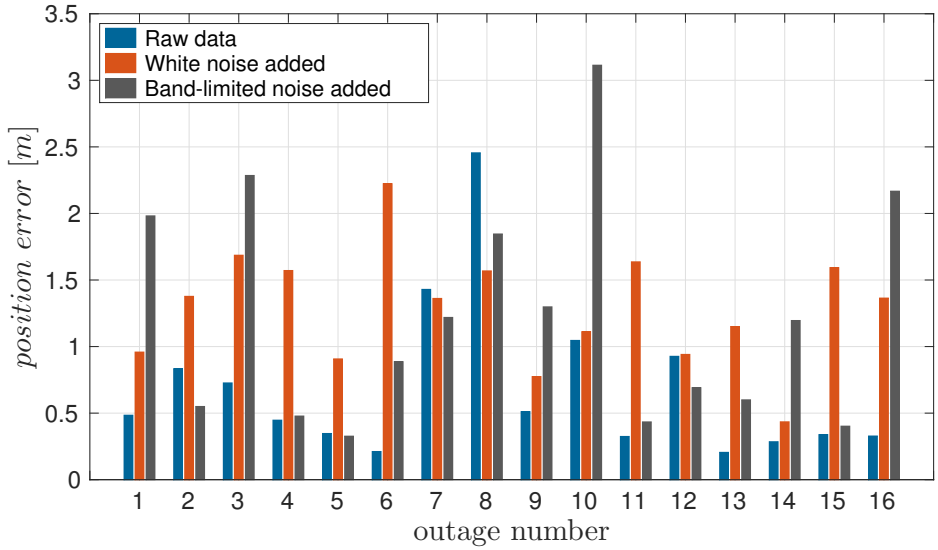


fig. 6.12: Position errors at each outage with different inertial data

In figure 6.12, the position errors at the end of the sixteen outage periods defined are plotted. As expected, it can be seen that the added noise affects the navigation performance of the multi-sensor system, which in turn leads to an increase of the position error at GNSS outages. Only at outage number seven and eight a different behaviour can be obtained. There the raw data has the highest error magnitude. This is probably due to an erroneous estimation of the sensor errors at the begin of the outage period. Table 6.8 summarizes the resulting position errors.

tab. 6.6: Position error at GNSS outages

	Mean [m]	**Max** [m]
Raw data	0.682	2.456
White noise added	1.291	2.224
Band-limited noise added	1.217	3.114

6.2.2 GNSS Outages With Denoised Inertial Data

As a next step, it is of interest if de-noising the inertial signal with wavelet thresholding at different decomposition levels can improve the inertial sensor error estimation, which in turn may reduce the resulting position errors.

a) Original inertial data: In figure 6.13, GNSS outages are simulated for inertial data with the decomposition levels of 3, 6 and 9.

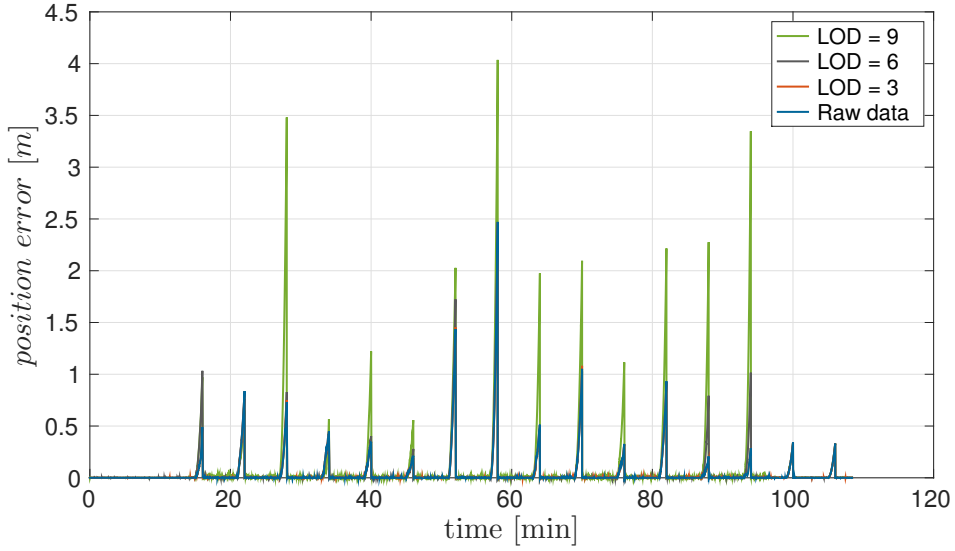


fig. 6.13: Position differences with respect to the reference trajectory

Figure 6.13 demonstrates that the level of decomposition does not affect the trajectory estimation significantly when GNSS is available (between outages). On the other hand, figure 6.13 shows directly that the magnitude of the resulting position error changes with each LOD, especially for a level of decomposition of nine. To illustrate the dependence of the position error for each decomposition level, the position difference to the reference trajectory at the end of the outage periods is plotted (6.14). It seems that an LOD of 9 manipulates the original signal, because motion information is filtered out and, subsequently, the position error at each GNSS outage increases. A second aspect is depicted in figure 6.14: the decomposition levels of 3 and 6 have nearly no effect on the maximum position error. The position error equals the results of the unfiltered inertial data.
It seems that de-noising the signal does not really improve the sensor error estimations. It is now of interest to find out whether the choice of the decomposition level affects the inertial sensor error estimation. For that, the temporal progress of the estimated sensor

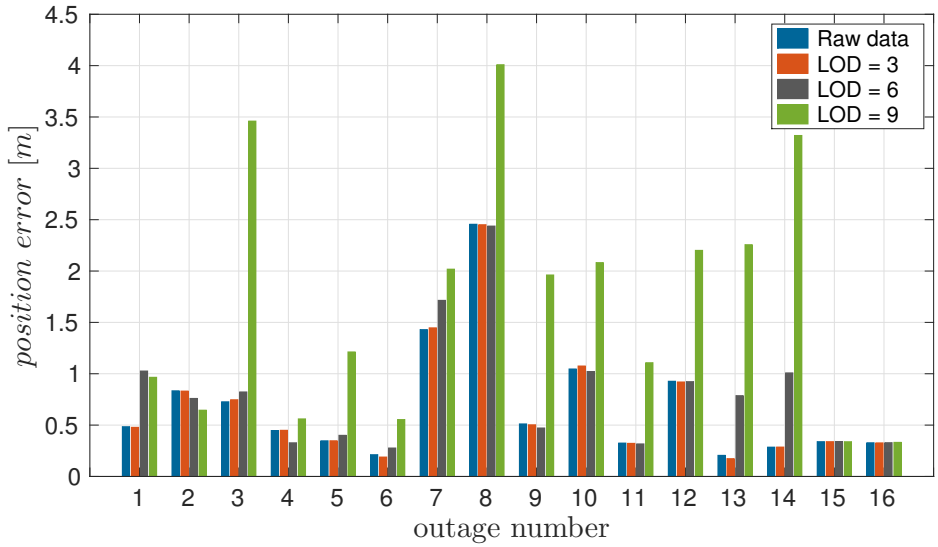
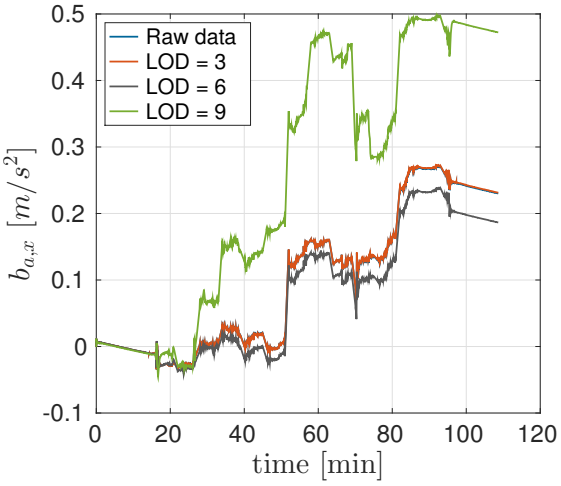


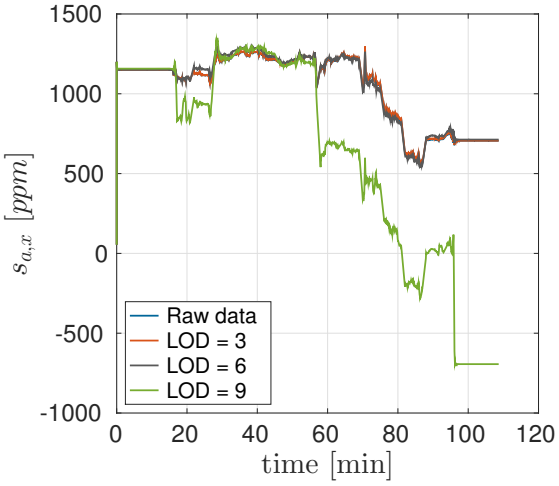
fig. 6.14: Maximum position error for different decomposition levels

errors (biases and scale factors) is represented in figure 6.15. This figure shows that the decomposition level directly affects the estimated sensor errors. The estimated sensor biases and scale factors are clearly correlated with the decomposition level. The result of de-noising inertial data leads to different sensor error estimations. It can also be seen that after an LOD of nine the sensor errors cannot be obtained in a sufficient accuracy any more and the values start drifting away.

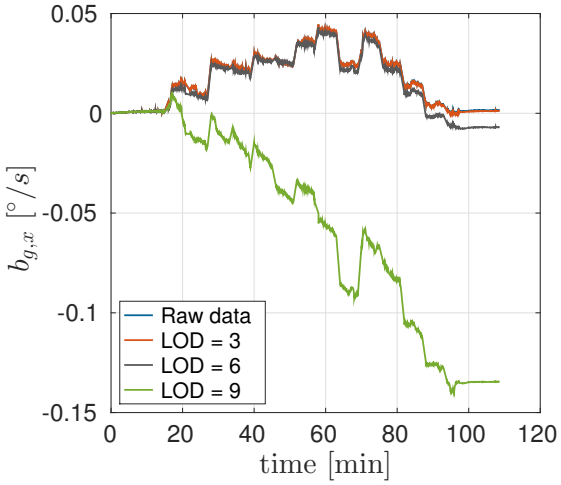
As a last step in this section, the magnitude of the position error at the simulated outage periods is computed, using the original and de-noised inertial data. The wavelet de-noising is performed by applying decomposition levels from one to nine. The results are summarized in table 6.7. Here it can be seen that the best result is obtained at a decomposition level of three, even though the effect is minimal. The average values of these errors are presented in figure 6.16. It seems that after six levels of decomposition, the inertial sensor error estimations start to deteriorate, which may be rebalanced by adjusting the stochastic model of the Kalman filter.



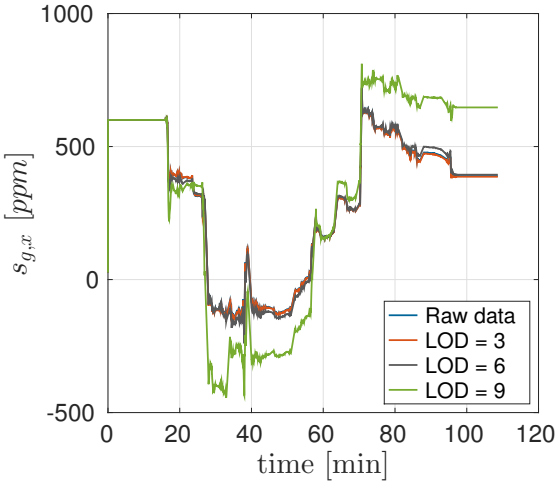
(a) Bias of accelerometer in x-direction



(b) Scale factor of accelerometer in x-direction



(c) Bias of gyroscope in x-direction



(d) Scale factor of gyroscope in x-direction

fig. 6.15: Estimated inertial sensor errors

tab. 6.7: Position error for each GNSS outage and with different filter steps

Outage No	**Raw data** [m]	**LOD 1** [m]	**LOD 2** [m]	**LOD 3** [m]	**LOD 4** [m]	**LOD 5** [m]	**LOD 6** [m]	**LOD 7** [m]	**LOD 8** [m]	**LOD 9** [m]
1	0.485	0.485	0.484	0.479	0.492	1.058	1.026	1.034	0.995	0.966
2	0.834	0.833	0.834	0.831	0.882	0.621	0.760	0.767	0.770	0.645
3	0.727	0.730	0.728	0.747	0.726	0.774	0.823	0.906	1.0682	3.460
4	0.447	0.447	0.451	0.449	0.390	0.373	0.328	0.319	0.317	0.560
5	0.347	0.347	0.347	0.348	0.391	0.399	0.400	0.386	0.416	1.212
6	0.212	0.209	0.208	0.189	0.212	0.364	0.277	0.263	0.252	0.554
7	1.430	1.429	1.430	1.447	1.485	1.655	1.715	1.706	1.688	2.018
8	2.456	2.455	2.454	2.451	2.510	2.436	2.438	2.394	2.423	4.009
9	0.512	0.511	0.512	0.503	0.475	0.507	0.472	0.458	0.477	1.962
10	1.046	1.048	1.046	1.075	1.021	0.977	1.022	1.029	0.987	2.082
11	0.325	0.326	0.326	0.322	0.333	0.354	0.317	0.318	0.312	1.107
12	0.927	0.925	0.925	0.920	0.919	0.906	0.923	0.914	0.891	2.201
13	0.205	0.205	0.207	0.173	0.302	0.764	0.787	0.777	0.889	2.257
14	0.285	0.286	0.285	0.286	0.302	1.106	1.008	0.981	1.149	3.320
15	0.339	0.339	0.339	0.339	0.339	0.340	0.340	0.340	0.341	0.338
16	0.328	0.328	0.327	0.327	0.328	0.329	0.329	0.329	0.329	0.332
Mean	0.682	0.682	0.681	0.680	0.694	0.810	0.810	0.808	0.832	1.689

Conclusion: It has been shown that the wavelet level of decomposition has nearly no influence on the resulting position errors at GNSS outages. The reason for this is that the inertial sensor errors, which are estimated by the Kalman filter, are adjusted in every case. A mere manipulation of the original signal can be achieved by choosing a too large decomposition level, which of course should be avoided. Improvements may be obtained if the stochastic model of the Kalman filter, especially the system noise, is adjusted for every decomposition level.

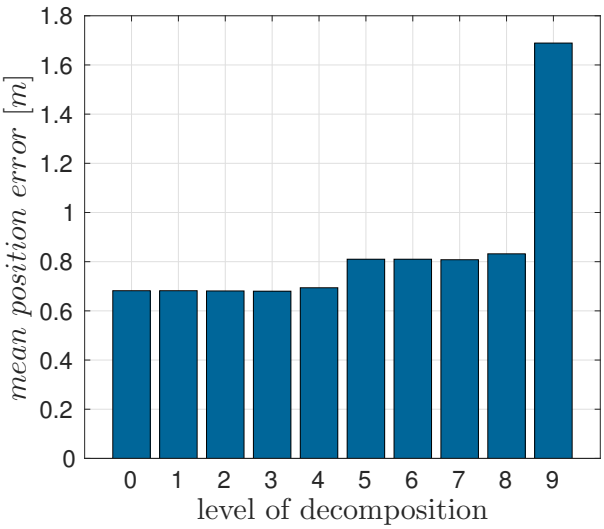


fig. 6.16: Mean position error for different decomposition levels (see table 6.7)

b) Synthetically noised inertial data: The resulting position difference is specified by the estimated inertial sensor errors. The better the sensor errors are estimated, the smaller is the resulting position error at the end of the outage periods. It is shown in table 6.8 that using inertial data with a larger noise term increases the average position error, because the noise term influences the estimation of the sensor errors. The sensor error estimation is more difficult and therefore more inaccurate. Next, it is of interest if wavelet decomposition can actually remove or reduce the negative effect of this synthetic noise.

Therefore, the trajectory inclusive GNSS outages is estimated with different levels of decomposition and by using the synthetically noised data from section 5.1.3. When inertial data with huge white noise is used, an improvement cannot really be obtained by filtering the data (see figure 6.17). The reason for this was already mentioned in *alignment with denoised inertial data* (section 6.1.2). The white noise term influences the information all over the whole frequency band and superimposes the motion term of the vehicle. Wavelet thresholding does reduce this effect but the residual noise influences the sensor error estimations. As mentioned before, after an LOD of nine the filter technique starts to downgrade significantly the signal. If band-limited noise is added to the inertial data, an improvement can be obtained (see figure 6.18). At the decomposition levels of three and six, the position error decreases at most GNSS outages. Therefore, the influence of the band-limited noise added has been reduced or even eliminated.

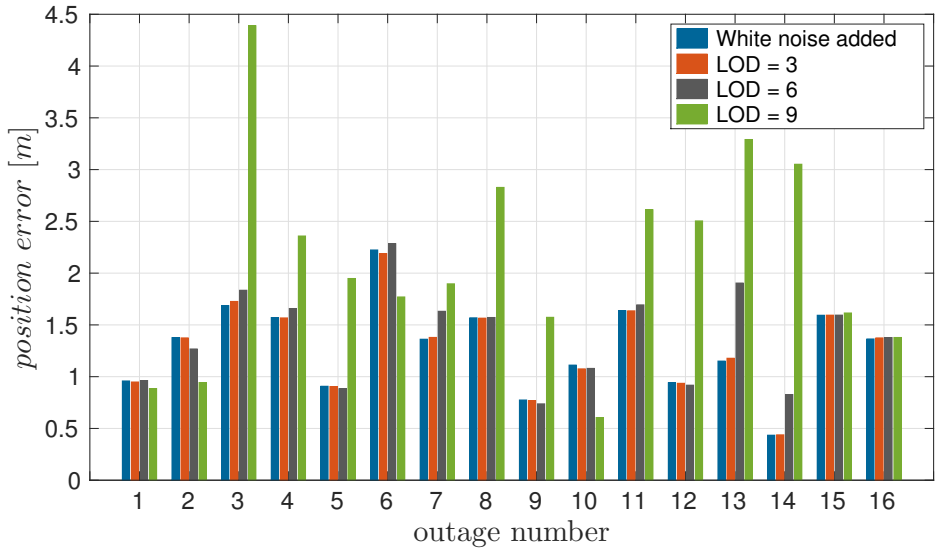


fig. 6.17: Position errors at different decomposition levels by using inertial data with added white noise

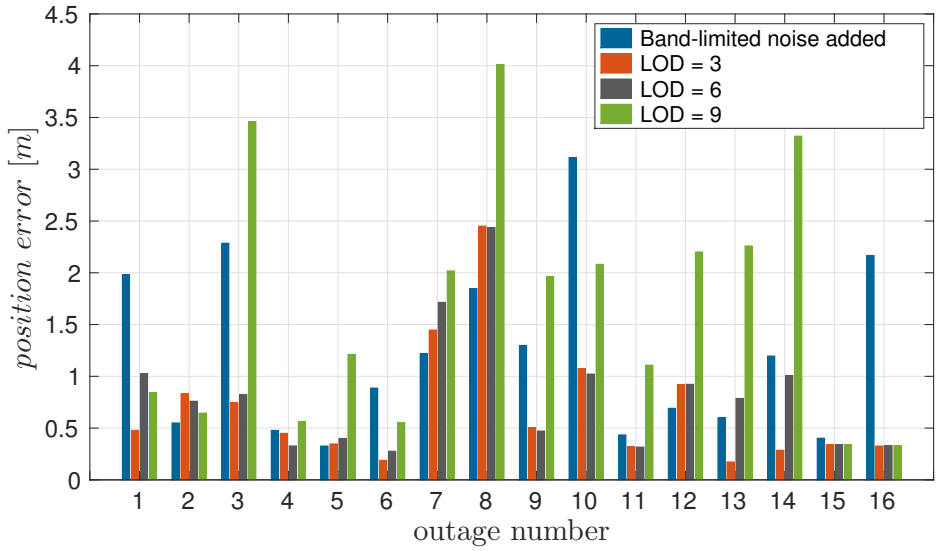
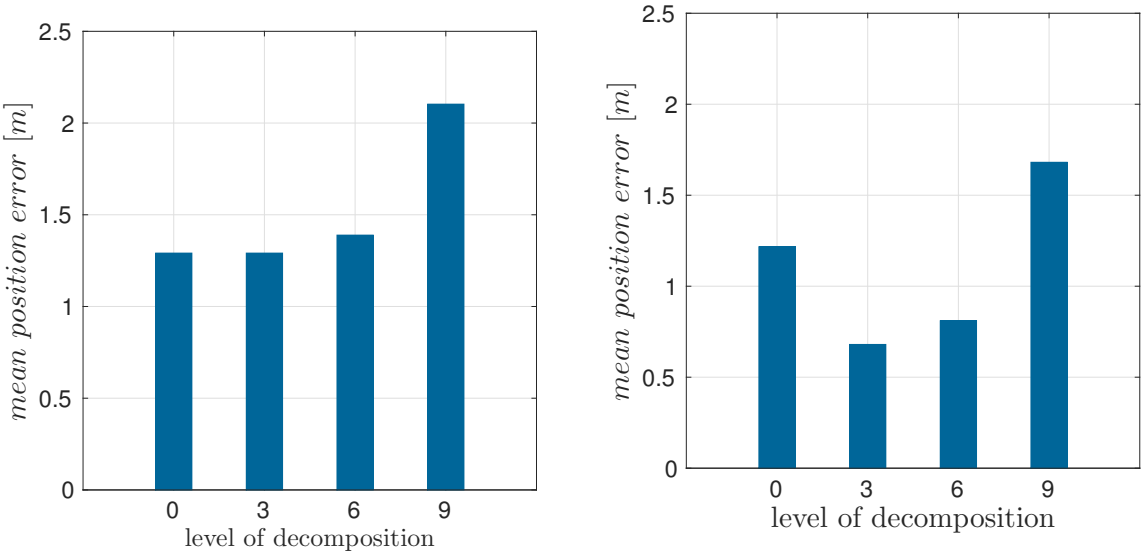


fig. 6.18: Position errors at different decomposition levels by using inertial data with added band-limited noise

Table 6.8 summarizes the maximum position error at different decomposition levels for white and band-limited synthetically noised inertial data. The average values of these errors are presented in figure 6.19.

tab. 6.8: Resulting position errors in the case of synthetically noised inertial data for each GNSS outage and different filter steps

Outage No	**White noise added**	**LOD 3**	**LOD 6**	**LOD 9**	**Band-limited noise added**	**LOD 3**	**LOD 6**	**LOD 9**
	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]
1	0.959	0.949	0.963	0.885	1.982	0.479	1.026	0.842
2	1.377	1.375	1.268	0.943	0.550	0.831	0.760	0.645
3	1.686	1.727	1.835	4.389	2.285	0.747	0.823	3.460
4	1.572	1.568	1.658	2.359	0.478	0.449	0.328	0.560
5	0.907	0.905	0.884	1.948	0.327	0.348	0.400	1.212
6	2.224	2.188	2.284	1.770	0.887	0.189	0.277	0.554
7	1.362	1.379	1.632	1.897	1.219	1.447	1.715	2.018
8	1.569	1.565	1.572	2.828	1.846	2.451	2.438	4.009
9	0.775	0.769	0.738	1.573	1.299	0.503	0.472	1.961
10	1.112	1.075	1.080	0.606	3.114	1.075	1.022	2.082
11	1.637	1.636	1.694	2.612	0.434	0.322	0.317	1.107
12	0.941	0.936	0.918	2.504	0.691	0.920	0.923	2.201
13	1.151	1.178	1.904	3.291	0.600	0.173	0.787	2.260
14	0.435	0.438	0.828	3.052	1.196	0.286	1.008	3.320
15	1.593	1.594	1.594	1.615	0.402	0.339	0.340	0.338
16	1.364	1.374	1.378	1.379	2.167	0.327	0.329	0.332
Mean	1.291	1.291	1.389	2.103	1.217	0.680	0.810	1.681



(a) Synthetic white noise added inertial data

(b) Synthetic band-limited noise added inertial data

fig. 6.19: Average position errors at different decomposition levels

Conclusion: It has been shown that the wavelet level of decomposition has nearly no influence on the resulting position errors at GNSS outages, when dealing with synthetically white noised data. The reason for this is that white noise influences information all over the whole frequency band and therefore superimposes the motion term wanted. Wavelet thresholding cannot separate the noise from the real motion information in the low frequency field. This residual noise is directly adopted in the inertial sensor errors, which are estimated by the Kalman filter. But they cannot be computed in the same quality as it is the case when dealing with the original data. In the case of band-limited noise added, an improvement can be obtained. The decomposition limits the noise and the sensor error estimation can be achieved in a sufficient accuracy again.

7 Conclusions

This thesis discussed an *IMU-GNSS multi-sensor system* in the field of navigation. At the beginning, the basic concept of inertial navigation was introduced and how GNSS can aid the navigation system. The integration type chosen for the sensor fusion was the *loose architecture* with a *closed-loop Kalman filter*. In addition, it is a well-known fact that the IMU sensor suffers from temporarily increasing uncertainties due to sensor errors. Therefore, in order to achieve navigation information in a cm-range these sensor errors were estimated within the Kalman filter process. Three datasets with IMU and GNSS measurements collected were analysed:

- Helicopter flight in *Laa an der Thaya*
- Helicopter flight in *Lanzenkirchen*
- Train ride *ÖBB*

The main emphasis was put on the helicopter flight *Laa an der Thaya*. Another central question in this thesis was to take a specific look on the quality of the IMU output. Therefore, the Matlab tool *IMUWaveletDenoising* was created to analyse the IMU observables. With the help of this tool, the frequency band describing the motion information of the moving object can be determined. The bandwidth containing the motion dynamics fell into the low frequency band of the spectrum (< 10 Hz). In addition, some undesirable disturber present in the dataset can be detected. Thereby, noise sources were revealed in the dataset *Laa an der Thaya*: A disturber at 20 Hz, and in addition, with the helicopter starting to move, another disturber at about 26 Hz are present (cf. 5.5). One suggestion was that the entire multi-sensor navigation solution will most likely be improved as soon as these disorders get eliminated. Therefore, the *wavelet thresholding* technique was introduced which allows, by decomposition of the signal, eliminating noise sources. This technique was also implemented in the Matlab tool in order to analyse and de-noise IMU measurements.

A next step was stressing the multi-sensor system. Therefore, the IMU measurements were superimposed by synthetic noise. Two kinds of noise were chosen and added: White noise (simulating a low-grade IMU) and band-limited noise (simulating vehicle vibrations). It was shown that band-limited noise can be eliminated by wavelet de-noising. The white noise term cannot be totally eliminated, because white noise influences information all over the whole frequency band and therefore also superimposes the motion term wanted. Alongside with this, an improvement of the signal-to-noise ratio was achieved (cf. 5.11). The effect of de-noised and synthetically noised IMU data was analysed in two steps of the multi-sensor navigation process. First, it was analysed how the noised IMU data affects the alignment process and subsequently, it was shown that the alignment procedure can be improved by applying wavelet de-noising. Band-limited noise can be removed, which would lead to an incorrect alignment determination and subsequently to a longer convergence time in the case of kinematic alignment. The second central point to be worked out was the question if preprocessing IMU data helps the Kalman filter to estimate the sensor errors. Therefore, GNSS outages were simulated. In such a situation the navigation system works as an INS stand-alone system and its performance is specified by the last estimated sensor errors. It has been shown that wavelet decomposition has nearly no influence on the resulting position errors at GNSS outages. The reason for this is that the inertial sensor errors, which are estimated by the Kalman filter, are adjusted in every case (cf. 6.15). Only a mere manipulation of the original signal can be achieved by choosing a too large decomposition level, which of course should be avoided. In the case of synthetically noised data, the resulting position error at the GNSS outages increased. With wavelet decomposition, band-limited noise can be excluded and the sensor error estimation can be achieved in a sufficient accuracy again. When dealing with white noise added data (low-grade IMU), wavelet thresholding cannot completely separate the noise from the real motion information wanted. This residual noise is directly adopted in the inertial sensor error estimates, but they cannot be computed in the same quality as it is the case when dealing with the original data.

7.1 Prospects of Future Work

Below follows a brief overview of possible further assignments to be considered.

- It would be of interest whether improving the SNR during wavelet de-noising might help the Kalman filter to converge faster in kinematic alignment.
- The de-noising of the IMU observables was just performed in post-processing. Can this technique also be applied in near real-time?
- Backward smoothing (reverse filtering) would improve the performance during GNSS outages.
- To improve the initial alignment also some external information such as magnetometer can be used.
- The multi-sensor system can be extended with another navigation sensor like an odometer.
- A prerequisite of Kalman filtering is that the input does not have any time correlation. Does the preprocessing of the IMU data thus influence the Kalman filter performance?
- Improvements may be obtained if the stochastic model of the Kalman filter, especially the system noise, is adjusted for every decomposition level.
- A gain of robustness and accuracy can be achieved if the sensor integration is performed as tight integration, because the navigation system will extract useful GNSS information even if less than four satellites are available.

Bibliography

Abdel-Hamid, W. *Accuracy Enhancement of Integrated MEMS-IMU/GPS Systems for Land Vehicular Navigation Applications*. PhD thesis, University of Calgary, 2005.

Blauensteiner, F. M. GPS/IMU Integration für die Präzise Bestimmung von Hoch Kinematischen Flugtrajektorien. Master's thesis, TU Wien, 2008.

De Agostino, M. A Multi-Frequency Filtering Procedure for Inertial Navigation. *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 1115–1121, 2008. doi: 10.1109/PLANS.2008.4569998.

Donoho, D. L. and Johnstone, I. M. Threshold Selection for Wavelet Shrinkage of Noisy Data. *Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1994. doi: 10.1109/IEMBS.1994.412133.

El-Diasty, M. and Pagiatakis, S. Calibration and Stochastic Modelling of Inertial Navigation Sensor Erros. *Journal of Global Positioning Systems*, 7(2):170–182, 2008. ISSN 14463156. doi: 10.5081/jgps.7.2.170.

El-Sheimy, N., Nassar, S., and Noureldin, A. Wavelet De-noising for IMU Alignment. *IEEE Aerospace and Electronic Systems Magazine*, 19(10):32–39, 2004. ISSN 08858985. doi: 10.1109/MAES.2004.1365016.

Gebre-Egziabher, D., Petovello, M., and Bevly, D. *GNSS Applications and Methods*, chapter Integration of GNSS and INS. Artech House, 2009. ISBN 9781596933309.

Goswami, J. C. and Chan, A. K. *Fundamentals of Wavelets: Theory, Algorithms and Applications*. Wiley, 2011. ISBN 9780470484135.

Grap, A. An Introduction to Wavelets. *IEEE Computational Science and Engineering*, 2 (2):50–61, 1995. ISSN 1070-9924. doi: 10.1109/99.388960.

Groves, P. D. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008. ISBN 978-1-58053-255-6.

Hinterberger, F., Eichhorn, A., and Weber, R. GNSS/IMU integration for the Precise Determination of Highly Kinematic Flight Trajectories. *Österreichische Zeitschrift für Vermessung und Geoinformation (VGI)*, (2):110–113, 2011.

Hofmann-Wellenhof, B., Legat, K., and Wieser, M. *Navigation: Principles Of Positioning And Guidance*. Springer Vienna, 2003. ISBN 9783211008287.

Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J. *Global Positioning System: Theory and Practice*. Springer Vienna, 2012. ISBN 9783709161999.

Jekeli, C. *Inertial Navigation Systems with Geodetic Applications*. Walter de Gruyter, 2001. ISBN 3110159031.

Kalman, R. A New Approach to Linear Filtering and Prediction Problems. *Journal of basic Engineering*, 82(1):35–45, 1960. ISSN 00219223. doi: 10.1115/1.3662552.

Kang, C. H., Kim, S. Y., and Park, C. G. Improvement of a Low Cost MEMS Inertial-GPS Integrated System Using Wavelet Denoising Techniques. *International Journal of Aeronautical and Space Sciences*, 12(4):371–378, 2011. doi: 10.5139/IJASS.2011.12.4.371.

Keller, W. *Wavelets in Geodesy and Geodynamics*. Walter de Gruyter, 2004. ISBN 9783110198188.

Merry, R. *Wavelet Theory and Applications: A Literature Study*. Technische Universiteit Eindhoven, 2005. URL `https://pure.tue.nl/ws/files/4376957/612762.pdf`.

Misiti, M. *Wavelet Toolbox: For Use with MATLAB® : User's Guide*, 2006.

Nassar, S. *Improving the Inertial Navigation System (INS) Error Model for INS and INS/DGPS Applications*. PhD thesis, University of Calgary, 2003.

Nassar, S. and El-Sheimy, N. Wavelet Analysis For Improving INS and INS/DGPS Navigation Accuracy. *Journal of Navigation*, 58:119–134, 2005. ISSN 0373-4633. doi: 10.1017/S0373463304003005.

Petovello, M. G. *Real-Time Integration of a Tactical-Grade IMU and GPS for High-Accuracy Positioning and Navigation*. PhD thesis, University of Calgary, 2003.

Reinstein, M. *Use of Adaptive Filtering Methods in Inertial Navigation Systems.* PhD thesis, Czech Technical University In Prague, 2010.

Rogers, R. M. *Applied Mathematics in Integrated Navigation Systems.* American Institute of Aeronautics and Astronautics, 2003. ISBN 9781563476563.

Skaloud, J., Bruton, A. M., and Schwarz, K. P. Detection and Filtering of Short-Term ($1/f^{\gamma}$) Noise in Inertial Sensors. *Journal of the Institute of Navigation*, 46(2):97–107, 1999. ISSN 0028-1522.

Titterton, D. H. and Weston, J. L. *Strapdown Inertial Navigation Technology, 2nd Edition.* Institution of Electrical Engineers, 2004. ISBN 0863413587.

STEFAN SCHAUFLER

PERSONAL INFORMATION

Born in Braunau am Inn, 4. October 1988

nationality Austria

adress Alxingergasse 27-29/19, 1100 Vienna

email schaufler.stefan@gmail.com

EDUCATION

2013–2016 Technical University of Vienna

Master of Geodesy and Geomatics Engineering

Thesis: *Preprocessing of IMU Observables and Multi-Sensor Kalman Filtering*

2009–2013 Technical University of Vienna

Bachelor of Geodesy and Geomatics Engineering

Thesis: *Dekodierung von RTCM 3.x Datenströmen*

2008–2009 Lebenshilfe Oberösterreich

Community Service

2003–2008 HTL Braunau am Inn

Matura

Field: *Electronics and Computer Engineering*

9. Mai 2016