Matthias Kalkgruber BSc

Opto-Inertial-Sensor Fusion for 6-DOF Tracking: Application in Low Cost Interaction Device

MASTER'S THESIS

to achieve the university degree of

Master of Science

Master's degree programme: Electrical Engineering

submitted to

Graz University of Technology

Supervisor

Dipl.-Ing. Dr. techn. Markus Neumayer

Institute of Electrical Measurement and Measurement Signal Processing

Graz, January 2016

AFFIDAVIT

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

Date

Signature

Abstract

This research concerns the development of a sensor fusion framework, which provides real time 6 degree-of-freedom tracking. The framework is based on the Unscented Kalman Filter and uses data from a low cost inertial measurement unit and an optical measurement system. For the optical measurement system a monocular position tracking algorithm based on sphere tracking was developed. A simplified version of the navigation equation was derived. Furthermore deterministic and stochastic error models of inertial sensors were in the fusion framework integrated. Also methods for calibration and identification of stochastic parameters for inertial sensors are provided and demonstrated. Finally a low cost 3D interaction device was implemented. For a complex task (combined fast/slow rotations and translation over a range of $\approx 0.5\,\mathrm{m}$) a RMSE of about 1.5 mm for the X and Y direction and a RMSE of about 3 mm for the Z direction, with respect to the camera, were achieved. For the orientation a RMSE of about 1.3 deg was obtained.

Zusammenfassung

Diese Arbeit befasst sich mit der Entwicklung eines Sensorfusion Frameworks. Das Framework benutzt einen Unscented Kalman Filter, der auf Basis von low cost Inertial- und optischen Positionsdaten, Orientierungs - und Positionsinformation in Echtzeit zur Verfügung stellt. Des Weiteren wurde ein Monokular-Position-Tracking Algorithmus implementiert, welcher die Position von kugelförmigen Objekten schätzt. Eine vereinfachte Version der Navigationsgleichung wurde in Kombination mit stochastischer und deterministischer Modellierung der Inertialsensoren in das Framework integriert. Geeignete Verfahren für die Kalibrierung und für die Bestimmung der stochastischen Parameter sind dargestellt und demonstriert. Schlussendlich wurde ein 3D Interaktionstool entwickelt, das in einem komplexen Anwendungsfall (zusammengesetzte schnelle/langsame Rotationen und Translationen in einem Bereich von $\approx 0.5\,\mathrm{m}$) einen RMSE von 1.5 mm für die X und Y-Richtung, und einen RMSE von 3 mm für die Z-Richtung erzielte. Die Orientierung erreichte einen RMSE von 1.3 deg.

Acknowledgment

Let me begin by thanking the people, who helped me in one way or another to write this thesis.

This thesis has been conducted in cooperation with PS-Tech in Amsterdam. During my stay in Amsterdam I got to know Dutch culture, of which I am really thankful for.

Thank you Arjen and Mark for giving me the opportunity of writing my thesis at their company and for sharing your wisdom with me (e.g. how to make perfect cappuccinos, singing the right carnival songs, what to do in case I win the lottery, . . . , ☺).

To Wim and Martijn. I am thankful for your support in coding and sharing your knowledge with me. Your efforts in introducing me to good Dutch food such as 'Oude kaas salade', 'Kip-kerriesalade', . . . , has been appreciated.

Thanks to Erwin for helping me out with hardware problems, as well as for all the fun we had casting silicon balls.

Also, I am grateful to Mays and Pedro for discovering Burgerbar and great beers with me.

Special thanks to my supervisor Markus for all his great help and supporting me with long lasting Skype conversation until late at night.

Thank you Joachim and Thomas for reading and correcting my thesis.

To all my wonderful friends and study colleagues for all the memories, friendship and support.

To Emma for all your support, love and unforgettable time in Amsterdam.

Schlussendlich möchte ich mich bei meiner Familie bedanken. Bei meinen Eltern Johann und Elfriede und Geschwistern Joachim und Michalea, die mich immer unterstützt haben.

Contents

1. Introduction

1.1. Motivation

What is Tracking / Why Tracking?

The aim of tracking is to continuously determine the states of a moving object over time [1]. Tracking orientation and position in 3D space is called 6 degree of freedom (DOF) tracking. The first 3 DOF refer to the translation in three perpendicular axes and the other 3 DOF refer to the rotation about three perpendicular axes.

Nowadays, the field of Virtual Reality (VR) and Augmented Reality (AR) is continuously growing [2]. Considering the recent developments of VR / AR glasses and the availability of affordable 3D screens even further increases can be expected.
With the growing access to the 3D virtual world, also the demand of 3D interaction devices is growing. Where the computer 'mouse' was developed to 'optimally' interact with a 2D screen, a similar device should be developed to interact with this virtual 3D world in an optimal way. Intuitively, the best way to do so would be by moving a real object in the 3D space. By tracking the position and orientation of that object, these movements can serve as input to the virtual world. Tracking is the technique to provide precise real-time information of the object's path and offers an intuitive way for 3D interaction.

A field where tracking is well developed is navigation. A well-known application is the (Global Navigation Satellite System) GNSS navigation system for cars. However, if GNSS signals are not available, other approaches have to be used. A smartphone is usually equipped with many different sensors such as accelerometers, magnetometers, gyroscopes, GNSS, cameras and many more. Combining those information sources in a smart way could provide accurate tracking information. Possible applications would be pedestrian navigation [3], emergency evacuations [4] and fireman guiding systems [5], [6].

To summarize, a strong demand for 6 DOF tracking exists.

How to get Position and Orientation Information?

Many methods to gather information of an object's position and orientation are available. A quick but not complete overview of available technologies will be presented [7]–[12]:

- Radio frequency based technologies such as Bluetooth, Wireless LAN, Ultra Wide Band or RFID techniques provide position information based on fingerprinting and triangulation. From all of these techniques Ultra Wide Band shows the most accurate and robust behavior.
- Ultrasound methods provide distance and angle measurements in order to determine the position of an object.
- By measuring the strength and orientation of an artificial excited magnetic field or the earth magnetic field, the orientation and also the position (active excited field) can be obtained. However, those methods are very sensitive to local distortions of the magnetic field.
- Infrared laser scanners provide very accurate position information, but the measurement of many points is slow and the equipment is fairly expensive.
- Tracking using depth cameras show great potential, however their accuracy is still limited with current low cost cameras.
- The accuracy of image based methods strongly depends on the algorithm and the properties of the tracked object. Marker based stereo optical tracking systems provide very accurate position and orientation information for nearly any kind of object. Unfortunately those systems are quite expensive.
- Recent developments in micro-electro-mechanical systems led to inexpensive ($5 - 15$ €) and miniaturized inertial sensors with increasing accuracy and performance. Although inertial sensors provide position and orientation information with a high update rate and high short term accuracy, their accuracy drastically decreases over time. For that reason standalone applications for position and orientation tracking are difficult to realize with current technologies.
- Global Navigation Satellite System (GNSS) such as Global Positioning Systems GPS are widely used for outdoor navigation. However, due to the need of direct sight and possible multipath propagations the performance for indoor application is poor.
- Automotive sensors such as steering encoders, odometers and velocity encoders provide information about the vehicles state. These sensors are mostly combined with GNSS systems.

Why Fusion?

Each of the above mentioned approaches has advantages and disadvantages. While one sensor provides very accurate position information at a low update rate, another sensor provides position information at a very high update rate, but only has a good short term accuracy and drifts over time. By combining those two sensors using sensor fusion techniques, position information with a high accuracy over long term and a high update rate can be provided. Sensor fusion is a model based strategy to combine available information in order to obtain an optimal estimation of the position and orientation.

A more detailed introduction to sensor fusion is given in chapter 5.

Opto-Inertial-Sensor Fusion for 6 DOF Tracking!

An inexpensive way to create a 6 DOF user interaction device could be based on low cost cameras in combination with inertial sensors. Nowadays, nearly every smartphone or computer is equipped with a decent camera and could provide position information based on a simple sphere tracking algorithm. Inertial sensors would provide orientation information and improve the short term accuracy. This combination presents a promising approach for an accurate and low cost tracking system for user interaction.

1.2. Aim of this Thesis

The aim of this thesis is to develop, demonstrate and test a low cost 3D interaction device with the use of a low cost inertial measurement unit (IMU) in combination with an inexpensive monocular position tracking system. In figure 1.1 the concept of the 3D interaction device is demonstrated.

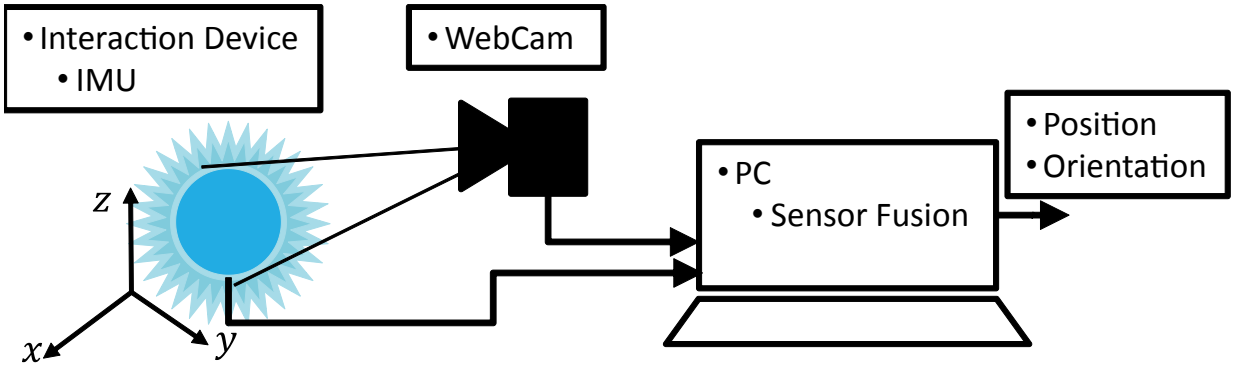


Figure 1.1: Concept for a low cost 3D interaction device.

In order to provide optimal 6-DOF estimations, a sensor fusion framework should be developed. This sensor fusion framework should assume measurements of an IMU and a position measurement system as basic inputs. However, it should also be adaptable to any other measurement device, so that the sensor fusion framework can be applied to other applications in the field of tracking. The application and hence the sensor fusion framework should operate on a consumer graed PC and provide 6 degree of freedom information in real time. In order to incorporate the IMU in the sensor fusion framework, the navigation equations should be derived. Furthermore to increase the performance, the characteristics of the IMU with respect to errors should be modeled and incorporated in the sensor fusion framework.

In order to provide 3D position information, a monocular position tracking algorithm should be developed and implemented. This algorithm should provide position information of a spherical ball using only one low cost camera.

Note: The original goal of this thesis was to only develop the sensor fusion framework. However, the existing low cost optical system did not provide the desired performance. As a result the development of the monocular position system was added as a side path.

1.3. Outline of this Thesis

The thesis is organized as follows:

Chapter 2 gives an introduction to the basics of inertial navigation such as reference frames, the notation of rotations and vectors. Furthermore, different attitude presentations and their properties are introduced. Finally the navigation equation for orientation and position are derived.

Chapter 3 provides characterization of the sensors, which are used for inertial navigation such as accelerometer, magnetometer and gyroscope. Furthermore, calibration procedures of deterministic errors are provided and demonstrated. A proper error model for the characterization of random errors will be introduced. By using Allan variance analysis, a method to provide parameters to model random errors is provided. In addition numerical values for the IMU, which has been used for the experiments in this thesis, are given.

In **Chapter 4** the monocular position tracking system will be presented. General problems within this task including the approaches of solving these problems will be discussed in detail.

Chapter 5 starts with the introduction to sensor fusion and gives a brief overview of possible methods. After explaining the selected fusion strategy in detail, the complete system model for the optical-inertial sensor fusion based on Chapter 2/3/4 will be explained. Finally several adaptations, in order to obtain a reasonable functioning fusion framework are described.

In **Chapter 6** the application of the sensor fusion framework for a low cost 3D interaction device is presented.

Chapter 7 depicts the results of the low cost 3D interaction device for different tracking tasks.

In **Chapter 8** the performance of the low cost 3D interaction device will be discussed, including possible improvements and potential for future work.

2. Short Introduction to Inertial Navigation

This chapter starts with a brief overview of Inertial Navigation Systems (INS) and is followed by an introduction to reference coordinate systems, which are used for the derivation of the strapdown navigation equations. Subsequently the notation for vectors and rotations will be presented, including a brief overview of different attitude representations, which will be used in this thesis. Finally the strapdown navigation equation for INS will be derived.

The aim of 6 DOF Inertial Navigation is to continuously provide the position and orientation of a moving object with respect to an inertial frame of reference. The information is provided by an inertial measurement unit (IMU). A typical IMU consists of three orthogonal gyroscopes and three orthogonal acceleration sensors [13].
INS systems can be categorized as Gimbaled navigation systems or Strapdown navigation systems [13]. In Gimbaled navigation systems, illustrated in figure 2.1, the inertial sensors are mounted on a stabilized platform to isolate the accelerometer from the rotational motions of the object to be tracked. This simplifies the position calculation.

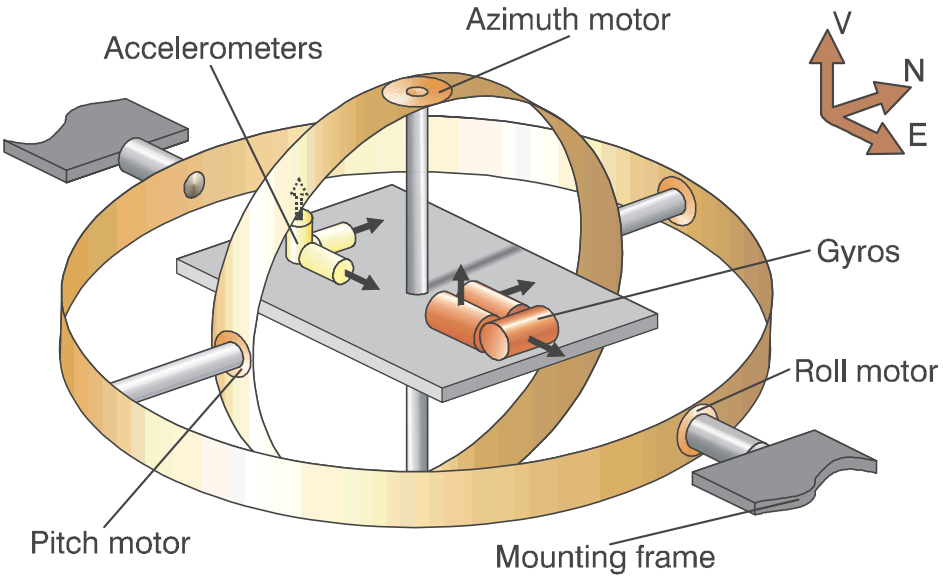


Figure 2.1: Gimbaled platform for Inertial Navigation [13].

In strapdown navigation systems the inertial sensors are rigidly fixed to the moving object and thus the sensors measure the same angular rate and accelerations as the moving object.

Compared to Gimbaled platforms the position and orientation extraction is more complicated. However, due to the simpler mechanical construction and the minimization of the inertial sensors, strapdown designs are commonly used in modern INS [13].

2.1. Reference Frames

For the derivation of the navigation equations the following reference coordinate systems will be intensively used. In the field of navigation these coordinate systems are also called frames. The reference frames will be explained according to [14], [15] and are illustrated in figure 2.2.

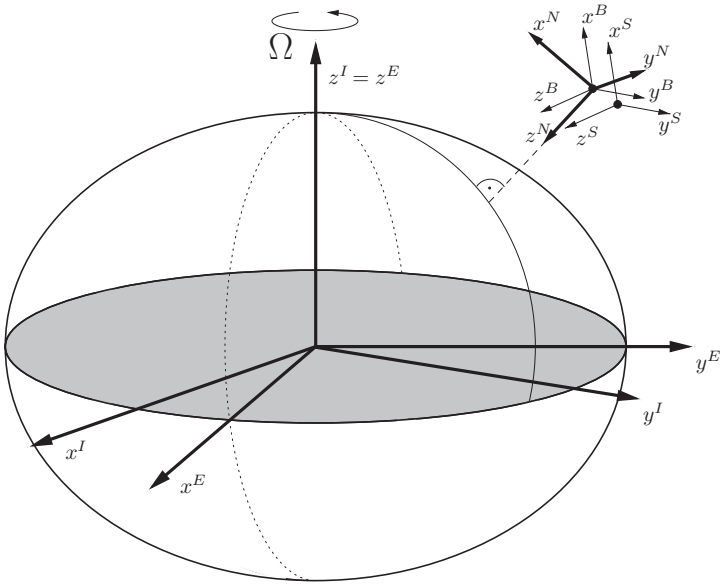


Figure 2.2: Coordinate frames for navigation adapted from [14].

- **Earth-Centered-Inertial-Frame (I-frame)**:
  As the name indicates, the earth-centered-inertial-frame (ECI) is a frame, whose origin is located at the center of the earth. The z^I axis is aligned with the rotational axis of the earth. The coordinate system is fixed with respect to fixed stars and therefore the earth rotates with respect to this frame. The inertial sensors measure acceleration and angular rotation with respect to the inertial-frame.
- **Earth-Centered-Earth-Fixed-Frame (E-frame)**:
  The earth-centered-earth-fixed-frame (ECEF) also has its origin at the center of the earth

and its z^E axis is aligned with the rotational axis of the earth. The coordinate axes are fixed with respect to the earth. The earth and E-frame rotate with an earth angular rotation of $\Omega = 7.291x10^{-5}$ rad/s $\approx 1\frac{1}{\text{day}}$, with respect to the I-frame.

- **Navigation-Frame (N-frame)**

  The navigation-frame is defined by the tangent plane of the earth's surface located at a chosen reference point or at the initial position of the object. For the alignment of the axes two different definitions exist. The axes of the east-north-up-frame (ENU) point to the east, north and up for the x^N, y^N, z^N axes, respectively. On the other hand the axes of the north-east-down-frame (NED) point north, east and down for the x^N, y^N, z^N axes, respectively. The latter representation is commonly used in navigation.

- **Body-Frame (B-Frame)**

  The axes of the body-frame are fixed with respect to the moving object to be tracked and usually point along the longitudinal axis, lateral axis and down for the x^B, y^B, z^B axes, respectively. Furthermore the origin is usually located at the object's center of gravity.

- **Sensor-Frame (S-Frame)**

  The sensor-frame is fixed with respect to the body-frame. Preferably its axes are aligned with the body-frame and orthogonal to each other. The origin is located at the center of the sensor, which is also desired to be at the origin of the body-frame.

  Due to constructive restrictions this is not always possible. This displacement, also known as lever arm, has to be considered in the strap-down navigation-equations or in the measurement function (see section 5.4.2.2).

Furthermore the location of an object can be described with latitude, longitude and height (LLH) with respect to the earth ellipsoid. However, this will not be covered in this thesis. The interested reader is referred to [14], [15].

2.2. Notation

To clarify the representation of position, orientation and their time derivatives the following notation will be introduced according to the definitions in [14].

- **Vectors**

  In the field of navigation two different categories of vectors are used.

  - **Position Vectors**

    A position vector represents the position from a point in space with respect to the origin of the reference frame. The position vector pointing from reference frame A to frame B is notated as follows:

    $$\mathbf{r}_B^A, \tag{2.1}$$

    where the vector $\mathbf{r}_B^A$ can also be seen as the location of frame B in frame A.

  - **Relative Vectors**

    Relative vectors (e.g. velocity $\mathbf{v}$, acceleration $\mathbf{a}$ or angular rotation vectors $\boldsymbol{\omega}$) are vectors, which do not have a specific starting or ending point. They only have a direction and magnitude. For their representation three indexes are needed and will be demonstrated with the following velocity vector:

    $$\mathbf{v}_{EB}^N. \tag{2.2}$$

    The upper index stands for the reference frame (e.g. N for navigation-frame) in which the velocity is measured. The two lower indexes represent the velocity of the B-frame with respect to the E-frame.

    This example represents the velocity measured in the N-frame of the B-frame with respect to the E-frame.

- **Rotation**

  To represent rotations, the rotation-matrix, also called Direction-Cosine Matrix (DCM)[1] is used. The rotation of a vector is given as the following matrix-vector multiplication:

  $$\mathbf{v}_{EB}^B = \mathbf{R}_N^B \mathbf{v}_{EB}^N. \tag{2.3}$$

  The rotation-matrix $\mathbf{R}_N^B$ rotates the vector $\mathbf{v}_{EB}^N$ from the N-frame to the B-frame. As result the velocity of the B-frame with respect to the E-frame is now represented in the B-frame.

[1] In the authors view the DCM represents the nicest and most compact notation.

Note that for transformations of position vectors, the origin of the reference frame has to be considered. Therefore, the transformation of the body's position $\mathbf{p}_B^E$ from the E-frame to the N-frame is given as:

$$\mathbf{p}_B^N = \mathbf{R}_E^N(\mathbf{p}_B^E - \mathbf{p}_N^E). \tag{2.4}$$

2.3. Attitude Representation

Several attitude representations are available, of which the following are used in this thesis:

- Euler Angles
- Rotation Matrix / Direction-Cosine-Matrix (DCM)
- Rotation Vectors / Axis-Angle
- Quaternion

Their mathematical background will be explained according to [14], [16].

2.3.1. Euler Angles

Euler angles describe the rotation of a rigid body or a reference-frame by three separate sequential rotations around a specific axis. In the field of navigation the ZYX-Euler angles are commonly used. An example of Euler angles, representing the rotation from N-frame to B-frame, is illustrated in figure 2.3 and can be defined as follows:
The first angle ψ, called yaw, describes the rotation around the local Z-axis. Then the rotation of the frame around the newly obtained Y-axes is given by the pitch θ. Finally, the roll ϕ gives the rotation around the newly obtained X-axis.
One advantage of Euler angles is the intuitive representation. However, the disadvantage is the Gimbal-lock, which is a singularity at a pitch of $\phi = \pm 90\,^\circ$ [14].

2.3.2. Rotation Matrix

The rotation matrix, also called Direction-Cosine-Matrix (DCM), has already been introduced in the previous sections. It is a 3x3-orthogonal matrix. An intuitive interpretation of the rotation matrix is given as follows: The first, second and third rows represent the x, y, z-axes of the transformed coordinate system, seen from the old coordinate system. As an example the rotation

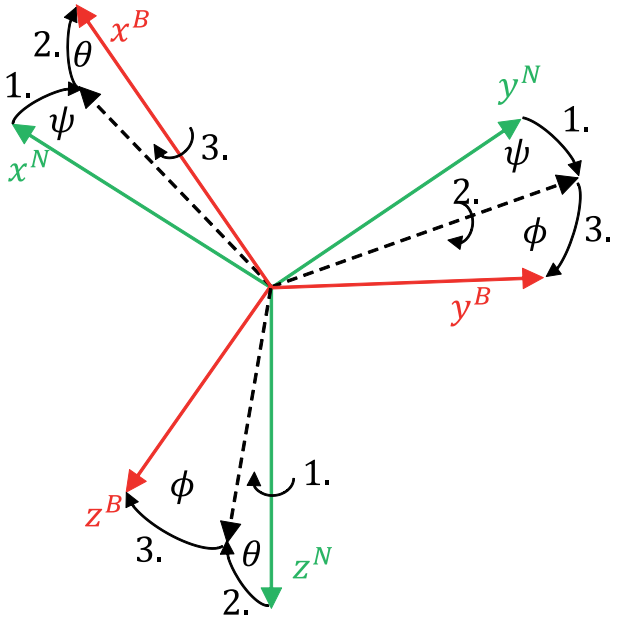


Figure 2.3: Attitude representation in Euler angles, adapted from [14].

from N-frame to B-frame is given as:

$$\mathbf{R}_N^B = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} \left(\mathbf{x}_B^N\right)^T \\ \left(\mathbf{y}_B^N\right)^T \\ \left(\mathbf{z}_B^N\right)^T \end{bmatrix}. \tag{2.5}$$

The rotation of a vector is carried out by a matrix-vector multiplication:

$$\mathbf{v}_{EB}^B = \mathbf{R}_N^B \mathbf{v}_{EB}^N. \tag{2.6}$$

Due to the orthogonality of the matrix, its inverse rotation can be given as its transposed.

$$\mathbf{R}_N^B = (\mathbf{R}_B^N)^{-1} = \mathbf{R}_B^{N^T}. \tag{2.7}$$

Compositions of rotations can be calculated by a matrix-matrix multiplication:

$$\mathbf{R}_1^3 = \mathbf{R}_2^3 \mathbf{R}_1^2. \tag{2.8}$$

Compared to Euler angles rotation matrices do not have singularities, but are known to cause difficulties maintaining the orthogonality. For storing nine parameters are used. Therefore more storage compared to other attitude representations is needed.

2.3.3. Rotation Vectors / Axis-Angle

Any rotation can also be described as a rotation θ around a fixed axis $\mathbf{n}$, as illustrated in figure 2.4.

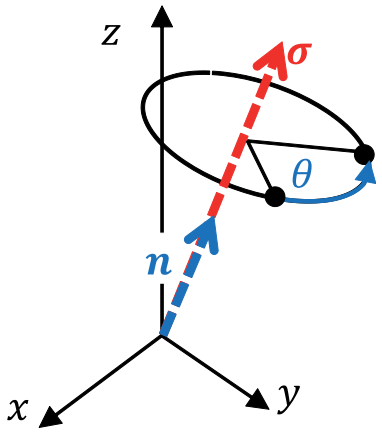


Figure 2.4: Rotation around rotation vector σ, $\theta = \|\sigma\|$.

For this rotation two equivalent notations exist. If axis and angle of the rotation are given separately, it is referred to axis-angle notation. On the other hand, when using rotation vectors the length of the vector represents the angle of rotation. Rotation vectors are denoted as:

$$\sigma = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \end{bmatrix}. \tag{2.9}$$

For the axis-angle notation, the rotation-axis $\mathbf{n}$ is defined by direction of the vector:

$$\mathbf{n} = \frac{\sigma}{\|\sigma\|}. \tag{2.10}$$

The angle of rotation θ is given by the magnitude of the rotation vector:

$$\theta = \|\sigma\|. \tag{2.11}$$

Rotation vectors offer a clear physical interpretation, e.g. the data obtained from IMU for gyroscopes is provided as rotation vectors. Furthermore no Gimbal locks can occur. However, a disadvantage of rotation vectors is the fairly complicated calculation of sequential rotations.

2.3.4. Quaternion

A quaternion $\mathbf{q}_A^B$, representing the rotation from frame A to frame B, is a four dimensional hyper complex vector. It is similar to complex numbers. However, instead of one real and one imaginary part j, quaternions have one scalar part s and an imaginary vector part $\mathbf{v}$. The imaginary vector part has three elements i, j and k. A quaternion is denoted as follows:

$$\mathbf{q}_A^B = q_1 + q_2 \cdot i + q_3 \cdot j + q_4 \cdot k = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} s \\ \mathbf{v} \end{bmatrix}. \tag{2.12}$$

Since a rotation can be described with three parameters, the representation using quaternions (4 parameters) seems to be over-determined. Introducing the unity norm constrain solves the over-determinedness:

$$\|\mathbf{q}\| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1. \tag{2.13}$$

Quaternions show a strong analogy to rotation vectors $\boldsymbol{\sigma}$. The scalar part s can be interpreted as the amount of rotation around the axis defined by the vector part $\mathbf{v}$. The exact relation is given as follows:

$$\mathbf{q} = \begin{bmatrix} \cos(\frac{\|\boldsymbol{\sigma}\|}{2}) \\ \frac{\sin(\frac{\|\boldsymbol{\sigma}\|}{2})}{\|\boldsymbol{\sigma}\|} \cdot \sigma_x \\ \frac{\sin(\frac{\|\boldsymbol{\sigma}\|}{2})}{\|\boldsymbol{\sigma}\|} \cdot \sigma_y \\ \frac{\sin(\frac{\|\boldsymbol{\sigma}\|}{2})}{\|\boldsymbol{\sigma}\|} \cdot \sigma_z \end{bmatrix} = \begin{bmatrix} \cos\left(\dfrac{\|\boldsymbol{\sigma}\|}{2}\right) \\ \dfrac{\boldsymbol{\sigma}}{\|\boldsymbol{\sigma}\|} \cdot \sin\left(\frac{\|\boldsymbol{\sigma}\|}{2}\right) \end{bmatrix}. \tag{2.14}$$

The composition of rotations can be calculated as :

$$\mathbf{q}_A^C = \mathbf{q}_B^C \bullet \mathbf{q}_A^B, \tag{2.15}$$

where $\bullet$ represents the quaternion multiplication, which can be written as a matrix-vector multiplication as follows:

$$\mathbf{q}_A \bullet \mathbf{q}_B = \begin{bmatrix} +q_{1,A} & -q_{2,A} & -q_{3,A} & -q_{4,A} \\ +q_{2,A} & +q_{1,A} & -q_{4,A} & +q_{3,A} \\ +q_{3,A} & +q_{4,A} & +q_{1,A} & -q_{2,A} \\ +q_{4,A} & -q_{3,A} & +q_{2,A} & +q_{1,A} \end{bmatrix} \cdot \begin{bmatrix} q_{1,B} \\ q_{2,B} \\ q_{3,B} \\ q_{4,B} \end{bmatrix}. \tag{2.16}$$

The inverse of a quaternion is obtained by inverting the imaginary part:

$$(\mathbf{q}_A^B)^{-1} = \mathbf{q}_B^A = \begin{bmatrix} q_{1,A}^B \\ -q_{2,A}^B \\ -q_{3,A}^B \\ -q_{4,A}^B \end{bmatrix}. \tag{2.17}$$

The rotation of a vector can be calculated as a left and right quaternion multiplication:

$$\begin{bmatrix} 0 \\ \mathbf{v}_{EB}^B \end{bmatrix} = \mathbf{q}_N^B \bullet \begin{bmatrix} 0 \\ \mathbf{v}_{EB}^N \end{bmatrix} \bullet (\mathbf{q}_N^B)^{-1}. \tag{2.18}$$

Although quaternions do not offer an intuitive representation, they offer some superior characteristics and are therefore widely used for navigation. The following list summarizes the advantages compared to other attitude representations:

- Quaternions do not have singularities e.g. Gimbal lock.
- Quaternions are easy to construct from IMU such as gyroscope data.
- Maintaining the unity constrain of quaternions is much easier than re-orthogonalization of rotation matrices.
- Rotating vectors and combining sequential rotations is fairly simple.
- Compared to rotation matrices (9-parameters) less storage (4-parameter) is needed.

In this thesis quaternions will be used for represent rotations in the strapdown navigation equations. For a more detailed explanation, as well as the calculation instructions for the transformation between the mentioned attitude representations, the interested reader is referred to [14], [16].

2.4. Strapdown Navigation Equations

In this section the strapdown navigation equations will be derived for the NED-frame (north-east-down). The navigation equations describe the motion of a rigid body in the three dimensional space. It starts with presenting the continuous time differential equations for orientation and position. Based on assumptions of the sensors, these equations are simplified. In order to perform the numerical integration, their time-discrete approximations are given. In the derivation it is assumed that the obtained sensor data from IMU is error free and without noise.
The derivation scheme in the following sections is based on [14].

2.4.1. Orientation

The differential equation for the rotation vector $\boldsymbol{\sigma}_N^B$, representing the rotation from the N-frame to the B-frame, is given by the Bortz equation [17]:

$$\begin{aligned}\dot{\boldsymbol{\sigma}}_N^B =& \boldsymbol{\omega}_{NB}^B + \frac{1}{2}\boldsymbol{\sigma}_N^B \times \boldsymbol{\omega}_{NB}^B \\ &+ \frac{1}{\left\|\boldsymbol{\sigma}_N^B\right\|^2}\left(1 - \frac{\left\|\boldsymbol{\sigma}_N^B\right\| \sin(\left\|\boldsymbol{\sigma}_N^B\right\|)}{2(1-\cos(\left\|\boldsymbol{\sigma}_N^B\right\|)}\right)\boldsymbol{\sigma}_N^B \times (\boldsymbol{\sigma}_N^B \times \boldsymbol{\omega}_{NB}^B).\end{aligned} \tag{2.19}$$

$\boldsymbol{\omega}_{NB}^B$ represents the rotation rate of the B-frame with respect to the N-frame. However, the IMU measures the rotation rate $\boldsymbol{\omega}_{IB}^B$ with respect to the I-frame. The relation between measured rotation $\boldsymbol{\omega}_{IB}^B$ and $\boldsymbol{\omega}_{NB}^B$ is given as [14]:

$$\boldsymbol{\omega}_{NB}^B = \boldsymbol{\omega}_{IB}^B - \mathbf{R}_N^B(\boldsymbol{\omega}_{IE}^N + \boldsymbol{\omega}_{EN}^N). \tag{2.20}$$

The terms $\boldsymbol{\omega}_{IE}^N$ and $\boldsymbol{\omega}_{EN}^N$ are the earth rotation rate and transport rate, respectively. The earth rotation $\boldsymbol{\omega}_{IE}^N$ represents the rotation of the earth with respect to the inertial-frame. Its representation in the navigation-frame depends on the location of the object and can be calculated with latitude φ and earth rotation $\Omega = 7.291x10^{-5}$ rad/s according to [14]:

$$\boldsymbol{\omega}_{IE}^N = \begin{bmatrix} \Omega \cdot \cos(\varphi) \\ 0 \\ \Omega \cdot \sin(\varphi) \end{bmatrix}. \tag{2.21}$$

The transport rate $\boldsymbol{\omega}_{EN}^N$ considers the rotation of the navigation-frame due to movement on the earth's curvature and depends on the actual velocity in the northern $v_{EB,\text{north}}^N$ and eastern $v_{EB,\text{east}}^N$

direction [14]:

$$\boldsymbol{\omega}_{EN}^{N} = \begin{bmatrix} \dfrac{v_{EB,\text{east}}^{N}}{R_e - h} \\ -\dfrac{v_{EB,\text{north}}^{N}}{R_n - h} \\ \dfrac{v_{EB,\text{east}}^{N} \tan(\varphi)}{R_e - h} \end{bmatrix}. \tag{2.22}$$

R_e represents the radius of the curvature in the prime vertical, R_n gives the meridian radius of the curvature and h is the height.

Assuming the maximum velocity $v_{\max}$ of the object $v_{\max} \approx 5\,\text{m/s}$ and the term $(R_e - h)$ is approximated with the mean earth radius $r_m = 6.371 \times 10^6\,\text{m}$, the maximum transport rate would be:

$$\omega_{EN,\max}^{N} = \frac{v_{\max}}{R_e - h} = \frac{5}{6.371 \times 10^6} \approx 1 \times 10^{-6}\,\text{rad/s}. \tag{2.23}$$

The influence of the transport rate and the earth rotation $\Omega = 7.291 \times 10^{-5}$ rad/s is smaller than the resolution ($\omega_{\text{LSB}} = 1.33 \times 10^{-4}\,\text{rad/s}$ [18]) of the IMU, which has been used for the experiments in this thesis. For that reason their influence will be neglected. The second cross-product term in equation (2.19) is also dropped due to its marginal influence [14]. This yields to the following simplified differential equation:

$$\dot{\boldsymbol{\sigma}}_{N}^{B} \approx \boldsymbol{\omega}_{IB}^{B} + \frac{1}{2}\boldsymbol{\sigma}_{\boldsymbol{N}}^{\boldsymbol{B}} \times \boldsymbol{\omega}_{IB}^{B}. \tag{2.24}$$

The rotation vector, which rotates the B-frame from time t_{k-1} to t_k, can be approximated according to [19], as:

$$\Delta\boldsymbol{\sigma}_{B,k-1}^{B,k} = \int_{t_{k-1}}^{t_k} \left(\boldsymbol{\omega}_{IB}^{B} + \frac{1}{2}\boldsymbol{\sigma}_{\boldsymbol{N}}^{\boldsymbol{B}} \times \boldsymbol{\omega}_{IB}^{B} \right) dt \approx \Delta\boldsymbol{\theta}_k + \frac{1}{12}\left(\Delta\boldsymbol{\theta}_{k-1} \times \Delta\boldsymbol{\theta}_k\right) \tag{2.25}$$

with

$$\Delta\boldsymbol{\theta}_k = \int_{t_{k-1}}^{t_k} \left(\boldsymbol{\omega}_{IB}^{B} \right) dt \approx \boldsymbol{\omega}_{IB,k}^{B} \cdot \Delta T \tag{2.26}$$

and

$$\Delta T = t_k - t_{k-1}. \tag{2.27}$$

ΔT is the sampling time and $\boldsymbol{\omega}_{IB,k}^{B}$ is the sensed rotation rate at time t_k.

The cross-term in equation (2.25) is called 1^{st} order coning term and is due to the non-commutativity of finite rotations. It should be noted that other approximations with higher and lower order coning terms exist. Only lower order terms approximations are also frequently used

in order to minimize the computational burden.
The rotation vector $\Delta\boldsymbol{\sigma}_{B,k-1}^{B,k}$ can be expressed as a quaternion:

$$\mathbf{q}_{B,k-1}^{B,k} = \begin{bmatrix} \cos\left(\frac{\left\|\Delta\boldsymbol{\sigma}_{B_{k-1}}^{B_k}\right\|}{2}\right) \\ \frac{\Delta\boldsymbol{\sigma}_{B_{k-1}}^{B_k}}{\left\|\Delta\boldsymbol{\sigma}_{B_{k-1}}^{B_k}\right\|} \cdot \sin\left(\frac{\left\|\Delta\boldsymbol{\sigma}_{B_{k-1}}^{B_k}\right\|}{2}\right) \end{bmatrix}. \tag{2.28}$$

Finally the quaternion $\mathbf{q}_N^{B,k}$, which represents the rotation from the N-frame to the B-frame at time step k, can be calculated using the quaternion chain rule as follows:

$$\mathbf{q}_N^{B,k} = \mathbf{q}_{B,k-1}^{B,k} \bullet \mathbf{q}_N^{B,k-1}. \tag{2.29}$$

2.4.2. Velocity

The differential equation for the velocity of the body-frame with respect to the navigation-frame, expressed in the navigation-frame, is given as [20]:

$$\dot{\mathbf{v}}_{NB}^N = \mathbf{R}_B^N \mathbf{a}_{IB}^B - \left(2\boldsymbol{\omega}_{IE}^N + \boldsymbol{\omega}_{BN}^N\right) \times \mathbf{v}_{EB}^N + \mathbf{g}^N, \tag{2.30}$$

where $\mathbf{R}_B^N$ represents the rotation from the B-frame to the N-frame. $\mathbf{a}_{IB}^B$ stands for the acceleration sensed from the IMU and is also called specific force. The second and third term reflect the influence of the Coriolis and the gravity force, respectively.

To appraise the maximum acceleration $\dot{\mathbf{v}}_{\text{Cor,max}}$ due to the Coriolis force, the following assumptions are made. The maximum velocity of the object is $v_{\max} \approx 5\,\text{m/s}$, the maximum transport rate is $\omega_{EN,\max}^N \approx 1 \times 10^{-6}\,\text{rad/s}$ (see equation (2.23)) and the earth rotation is $\omega_{IE,\max}^N = \Omega \approx 7 \times 10^{-5}\,\text{rad/s}$. Therefore the maximum acceleration is given as:

$$\begin{aligned} \dot{\mathbf{v}}_{\text{Cor,max}} &= (2 \cdot \omega_{IE,\max}^N + \omega_{EN,\max}^N) \cdot v_{\max} \\ &= (2 \cdot 7 \times 10^{-5} + 10^{-6}) \cdot 5 \\ &\approx 7 \times 10^{-4}\,\text{m/s}^2. \end{aligned} \tag{2.31}$$

The maximum acceleration due to the Coriolis force $\dot{\mathbf{v}}_{\text{Cor,max}}$ lies in the range of one or two LSB ($a_{\text{LSB}} = \frac{9.81}{16384}\,\text{m/s}^2 = 5.9875 \times 10^{-4}\,\text{m/s}^2$ [18]) of the IMU, which has been used for the

experiments in this thesis. For that reason the Coriolis term is neglected, which results in the following simplified differential equation:

$$\dot{\mathbf{v}}_{NB}^{N} = \mathbf{R}_{B}^{N}\mathbf{a}_{IB}^{B} + \mathbf{g}^{N}. \tag{2.32}$$

The digital velocity integration algorithm is then formulated as:

$$\mathbf{v}_{NB,k}^{N} = \mathbf{v}_{NB,k-1}^{N} + \int_{t_{k-1}}^{t_k} \left(\mathbf{R}_{B}^{N}\mathbf{a}_{IB}^{B} + \mathbf{g}^{N}\right) dt \tag{2.33}$$

$$= \mathbf{v}_{NB,k-1}^{N} + \Delta\mathbf{v}_{NB,k}^{N} \tag{2.34}$$

$$= \mathbf{v}_{NB,k-1}^{N} + \Delta\mathbf{v}_{NB,a,k}^{N} + \Delta\mathbf{v}_{NB,g,k}^{N}. \tag{2.35}$$

The velocity increment due to the specific force is approximated according to [21]:

$$\Delta\mathbf{v}_{NB,a,k}^{N} = \int_{t_{k-1}}^{t_k} \left(\mathbf{R}_{B,k}^{N}\mathbf{a}_{IB}^{B}\right) dt \tag{2.36}$$

$$\approx \mathbf{R}_{B}^{N}\left(\Delta\mathbf{v}_{IB,k}^{B} + \frac{1}{2}\Delta\boldsymbol{\theta}_k \times \Delta\mathbf{v}_{IB,k}^{B} + \frac{1}{12}\left(\Delta\boldsymbol{\theta}_k \times \Delta\mathbf{v}_{IB,k-1}^{B} + \Delta\mathbf{v}_{IB,k}^{B} \times \Delta\boldsymbol{\theta}_{k-1}\right)\right), \tag{2.37}$$

with

$$\Delta\mathbf{v}_{IB,k}^{B} = \int_{t_{k-1}}^{t_k} \mathbf{a}_{IB}^{B} dt \approx \mathbf{a}_{IB,k}^{B} \cdot \Delta T. \tag{2.38}$$

The cross-terms in equation (2.36) are called 1^{st} and 2^{nd} order sculling terms and are due to simultaneous linear and angular motions.
It should be noted, that analog to the coning terms also other approximations with higher and lower order sculling terms exist. However, only lower order terms are frequently used, in order to minimize the computational burden. The velocity increment due to gravity is given by:

$$\Delta\mathbf{v}_{NB,g,k}^{N} = \int_{t_{k-1}}^{t_k} \mathbf{g}^{N} dt = \mathbf{g}^{N} \cdot \Delta T. \tag{2.39}$$

$\mathbf{a}_{IB,k}^{B}$ represents the sensed acceleration by the IMU at time t_k and $\mathbf{g}^{N}$ gives the gravity-vector in the NED-frame.

2.4.3. Position

The differential equation for the position $\mathbf{p}_B^N$ of the B-frame with respect to the N-frame can be written as:

$$\dot{\mathbf{p}}_B^N = \mathbf{v}_{NB}^N. \tag{2.40}$$

The digital position integration is obtained using a zero-order hold (ZOH) approximation according to [22]:

$$\mathbf{p}_{B,k}^N = \mathbf{p}_{B,k-1}^N + \int_{t_{k-1}}^{t_k} \mathbf{v}_{NB}^N dt \tag{2.41}$$

$$\approx \mathbf{p}_{B,k-1}^N + \mathbf{v}_{NB,k-1}^N \Delta T + \mathbf{a}_{NB,k}^N \frac{\Delta T^2}{2} \tag{2.42}$$

$$= \mathbf{p}_{B,k-1}^N + \mathbf{v}_{NB,k-1}^N \Delta T + \Delta \mathbf{v}_{NB,k}^N \frac{\Delta T}{2} \tag{2.43}$$

with

$$\Delta \mathbf{v}_{NB,k}^N = \Delta \mathbf{v}_{NB,a,k}^N + \Delta \mathbf{v}_{NB,g,k}^N. \tag{2.44}$$

3. Characterization and Calibration of Sensors in Inertial Navigation

A traditional inertial sensor only provides information about acceleration and angular rate of a moving body with respect to the inertial-frame. However, since the development of micro-electro-mechanical systems (MEMS) also other sensors are usually added to the IMU. The number of sensors in an IMU, which can be used to determine position or orientation, is called degree of freedom DOF. This definition should not be confused with the DOF of a moving rigid body. A 10-DOF-IMU (MPU-9150) has been used for this thesis and is composed of a 3-axes gyroscope, 3-axes accelerometer, 3-axes magnetometer and a barometer. Except for the barometer, all other 9-DOF are used for navigation.
In the following a list of typical errors of an IMU is given [18], [23]–[25]:

- **Constant Bias:** Constant offset of a measurement.
- **Bias drift:** Change of the bias over time.
- **Turn-on to Turn-on Bias:** Different initial bias for each power up cycle.
- **Sensor Noise:** Zero mean random process, which corrupts the measurement.
- **Scale-Errors:** Error of the slope defined by the scale-factor between assumed and real slope.
- **Miss-Alignment / Cross-Coupling:** Error due to non-orthogonal sensitive axes and sensitivity to orthogonal axes.
- **Scale-Nonlinearity:** Deviation from the straight line, which is defined by the scale-factor
- **Temperature Effects:** The above mentioned effects such as bias, scale-factor and noise are further. influenced by temperature changes, which are caused by environmental changes or self-heating.
- **Hard-Iron-Distortion:** Distortion in the magnetic field, due to objects which create magnetic fields.
- **Soft-Iron-Distortion:** Distortion in the magnetic field, due to objects which deflect or alternate the existing magnetic field.
- **G-Sensitivity:** Observed bias in gyroscopes as a function of acceleration.

- **Supply Voltage:** Change of scale-factor due to changes in supply-voltages.
- **Aging:** Change of bias, scale-factor or noise characteristics due to aging effects.
- **Size-Effect:** Acceleration during rotations, due to non-co-located accelerometers.

These errors can be categorized in deterministic/systematic and in random/stochastic errors. In the following section the models of deterministic and stochastic errors, which have the largest effect on the integrity, are described for each sensor. Calibration procedures for deterministic errors will be presented and demonstrated. Furthermore methods to determine the parameters for stochastic error models are presented.

3.1. Deterministic-Errors

Measurement errors cause the measured value to deviate from the true value. Deterministic errors are characterized in such a way, that they can be reproduced and therefore be compensated. The deterministic errors with the largest effect on the integrity of the IMU are bias, scale-factor-errors, misalignment and cross coupling effects. They are due to imperfection of the device itself [26]. Their influence can be modeled with the following model [26]:

$$\tilde{\mathbf{x}}_c = \mathbf{M}_{\text{Dev}} \cdot \left(\tilde{\mathbf{x}}_d^B - \mathbf{b}_{\text{Dev},d} \right) \tag{3.1}$$

with

$$\mathbf{M}_{\text{Dev}} = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{pmatrix}, \tag{3.2}$$

$$\mathbf{b}_{\text{Dev},d} = \begin{pmatrix} b_{\text{Dev},d_x} \\ b_{\text{Dev},d_y} \\ b_{\text{Dev},d_z} \end{pmatrix}. \tag{3.3}$$

$\tilde{\mathbf{x}}_c$ presents the calibrated measurement and $\tilde{\mathbf{x}}_d$ is the measurement, which is corrupted by deterministic-errors. The scale factor matrix $\mathbf{M}_x$ represents scale, misalignment and cross coupling errors, whereas $\mathbf{b}_{\text{Dev},d}$ defines the deterministic bias of the device 'Dev'. 'Dev' is a placeholder for the specific device, such as 'a' for accelerometer. The diagonal elements are called the scale-factors and the off-diagonals are called cross-axes factors. For an ideal sensor the deterministic bias and the cross-axes factors would be zero and the scale-factors would be equal to 1.

3.1.1. Accelerometer

3.1.1.1. Accelerometer Calibration Method

To obtain the compensated accelerometer measurement $\tilde{\mathbf{a}}^B_{IB,c}$ from the deterministic error affected measurement $\tilde{\mathbf{a}}^B_{IB,d}$, the scale factor matrix $\mathbf{M}_a$ and deterministic bias $\mathbf{b}_{a,d}$ of the accelerometer have to be estimated. For the calibration, an adapted version of the iterative calibration method described in [26] is used and will be explained:
For this calibration method, the user has to place the IMU in at least nine different static orientations. In these static conditions it is assumed, that the magnitudes of the acceleration is equal to the gravity g:

$$\sqrt{\left(\tilde{a}^B_{IB,c}\right)^2_x + \left(\tilde{a}^B_{IB,c}\right)^2_y + \left(\tilde{a}^B_{IB,c}\right)^2_z} = g. \tag{3.4}$$

It can be shown [27] that the in general asymmetric scale factor matrix $\mathbf{M}_a = \mathbf{M}_{\text{asym}}$ can be decomposed in a pure rotational part $\mathbf{R}$ and a symmetrical part $\mathbf{M}_{\text{sym}}$:

$$\mathbf{M}_a = \mathbf{M}_{\text{asym}} = \mathbf{R} \cdot \mathbf{M}_{\text{sym}}. \tag{3.5}$$

The equation (3.4) can be interpreted, such that the acceleration vector describes a sphere located at the origin with the radius g. Any rotation of a sphere does not affect the function of a sphere, hence the rotational part $\mathbf{R}$ cannot be estimated. For that reason only the symmetrical part of $\mathbf{M}_a$ can be calculated and the symmetry constrain on the scale factor matrix, according to [26], is introduced:

$$S_{xy} = S_{yx}, S_{xz} = S_{zx}, S_{yz} = S_{zy}. \tag{3.6}$$

The rotational part of the scale factor matrix will be calculated in the gyroscope calibration (see section 3.1.3.1). It aligns the sensor axes of the gyroscope and the accelerometer.
To form an optimization criterion the following error term $\mathbf{r}$ is introduced:

$$\mathbf{r} = [r_1, \ldots, r_m]^T \tag{3.7}$$

with:

$$r_n = \left(\tilde{\mathbf{a}}^B_{IB,c}\right)^2_x + \left(\tilde{\mathbf{a}}^B_{IB,c}\right)^2_y + \left(\tilde{\mathbf{a}}^B_{IB,c}\right)^2_z - g^2 \tag{3.8}$$

$$= \sum_{i=x,y,z} \left\{ \sum_{j=x,y,z} \left[S_{ij} \cdot \left(\left(\tilde{\mathbf{a}}^B_{IB,d}\right)_{j,n} - b_{j,d} \right) \right]^2 \right\} - g^2 \tag{3.9}$$

where $\left(\tilde{\mathbf{a}}^B_{IB,d}\right)_{j,n}$ is the j-th (x/y/z) output of the error affected measurement for the n-th orientation. The accumulative error E^t at iteration t can be calculated by the summation of r_n^2 over all orientations:

$$E^t = E(b_{x,d}, b_{x,d}, b_{z,d}, S_{xx}, S_{yy}, S_{zz}, S_{xy}, S_{xz}, S_{yz})^t = \sum_{k=1}^{m} r_n^2. \tag{3.10}$$

The parameter vector at iteration t is composed of bias and scale matrix elements:

$$\mathbf{p}^t = [p_1, \ldots, p_9]^T = \left[b_{x,d}, b_{y,d}, b_{z,d}, S_{xx}, S_{yy}, S_{zz}, S_{xy}, S_{xz}, S_{yz}\right]^T. \tag{3.11}$$

Starting with an initial guess, the parameter vector $\mathbf{p}^t$ can be iteratively updated using a Gauss-Newton type method in order to minimize the accumulative error E. Note: In the original algorithm Newton's method was proposed [26]. The update-step is then given as:

$$\mathbf{p}^{t+1} = \mathbf{p}^t - \alpha^t \cdot \delta^t. \tag{3.12}$$

The correction term δ^t can be found by solving the following equation:

$$\mathbf{J}^t (\mathbf{J}^t)^T \delta^t = -\mathbf{J}^t \mathbf{r}^t. \tag{3.13}$$

The Jacobian matrix $\mathbf{J}^t$ is given as:

$$\mathbf{J}^t = \left[\nabla r_1(p^t), \nabla r_2(p^t), \ldots, \nabla r_m(p^t)\right]. \tag{3.14}$$

The damping parameter $\alpha^t \leq 1$ is updated as follows:

$$\alpha^{t+1} = \begin{cases} \alpha^t - \tau \cdot \alpha^t, & E^{t+1} \leq E^t \\ \tau \cdot \alpha^t, & \text{else} \end{cases}. \tag{3.15}$$

with τ the fading factor chosen $\tau = 0.01 \ldots 0.05$. The optimization method is stopped, if the following criterion is achieved:

$$max \left\{ \left| \frac{p^t - p^{t-1}}{(p^t - p^{t-1})/2} \right| \right\} < \epsilon, \tag{3.16}$$

with ϵ as an empirical chosen threshold. e.g. $\epsilon = 1 \times 10^{-10}$.

3.1.1.2. Identification of Static Conditions

In general, a recorded time sequence of accelerometer measurements $\tilde{\mathbf{a}}_k$ contains sequences in static (device is lying still) and non-static conditions (device is moved from one to another orientation). However, for the accelerometer calibration only measurements of different orientations in static conditions are needed. Therefore an algorithm which automatically detects these static conditions will be presented. The concept to identify those static conditions is based on the following assumption:
The local variance of the accelerometer measurements in static conditions is much smaller than in non-static conditions. The variance in static-conditions is assumed to be caused by sensor-noise and is considered to be the same for all orientations. Therefore static-conditions can be identified, if the local variance falls under a defined threshold. The local variance $\sigma^2_{L,k}$ at step k can be calculated using a moving window as follows:

$$\sigma^2_{L,k} = \frac{1}{N-1} \cdot \sum_{i=k-N/2}^{k+N/2-1} \left(\tilde{\mathbf{a}}_i - \boldsymbol{\mu}_{L,k}\right)^2, \tag{3.17}$$

$$\boldsymbol{\mu}_{L,k} = \frac{1}{N} \cdot \sum_{i=k-N/2}^{k+N/2-1} (\tilde{\mathbf{a}}_i), \tag{3.18}$$

where N is the window length, which should be an even number. $\boldsymbol{\mu}_{L,k}$ is the local mean and $\tilde{\mathbf{a}}_i$ is the measured acceleration-vector at time step i.
A static condition candidate is found, if the following criterion is satisfied:

$$||\sigma^2_{L,k}|| \leq \beta \cdot \sigma^2_{L,\min}, \tag{3.19}$$

where $\sigma^2_{L,\min}$ represents minimal local variance defined as:

$$\sigma^2_{L,\min} = min\left(||\sigma^2_{L,k}||\right). \tag{3.20}$$

β is a tunning factor. $||\sigma^2_{L,k}||$ represents the euclidean norm defined by:

$$||\sigma^2_{L,k}|| = \sqrt{(\sigma^2_{L,k,x})^2 + (\sigma^2_{L,k,y})^2 + (\sigma^2_{L,k,z})^2}. \tag{3.21}$$

The next step is to identify starting points $\mathbf{k}_{\text{SP}}$ and ending points $\mathbf{k}_{\text{EP}}$ for all static conditions. A starting point $k_{\text{SP},n}$ for the n-th orientation is found, if the point at time step k is a candidate and the point at time step $k-1$ does not satisfy the criterion in equation (3.19). Similar an ending point $k_{\text{EP},n}$ is found, if point $k+1$ does not satisfy the criterion in equation (3.19). In order to

reject isolated points, starting and ending points which do not satisfy the following criterion are removed:

$$k_{\mathrm{EP},n} - k_{\mathrm{SP},n} \geq L_{\min}, \tag{3.22}$$

where $L_{\min}$ is the minimal static condition length.

Finally the mean value for the n-th orientation $\tilde{\mathbf{a}}_n$ can be calculated as:

$$\tilde{\mathbf{a}}_n = \frac{1}{k_{\mathrm{EP},n} - k_{SP,n}} \cdot \sum_{i=k_{\mathrm{SP},n}}^{k_{\mathrm{EP},n}} \tilde{\mathbf{a}}_i. \tag{3.23}$$

3.1.1.3. Accelerometer Calibration

Due to the lack of a turntable or equivalent measurement equipment and in order to allow in-field calibrations, a simple, inexpensive and innovative calibration procedure was invented. Since the IMU was rigidly mounted in a ball only and a cup was needed for the calibration procedure. It will be explained hereinafter:

The ball containing the rigidly mounted IMU is placed in the cup to ensures that no movements or rotations would perturb the signals during measurements. In order to collect measurements of different orientations, the ball is placed pointing in various orientations. The calibration setup is illustrated in figure 3.1.

(a)Theoretical calibration setup. (b)Real calibration setup.

Figure 3.1: Illustration of the accelerometer calibration setup.

For the identification of static conditions (see section 3.1.1.2) a window length of $N = 200$, a threshold for accepting candidates $\beta = 2$ and a minimal static condition length $L_{\min} = 50$ was chosen. In figure 3.2 an exemplary calibration sequence for the identification of static conditions is illustrated. Figure 3.3 depicts a zoomed section. The blue, dark-green and red data represents the measured acceleration in x, y, z direction of the IMU. The bright-green data marks identified static conditions. Both figures show the successful identification of static conditions.

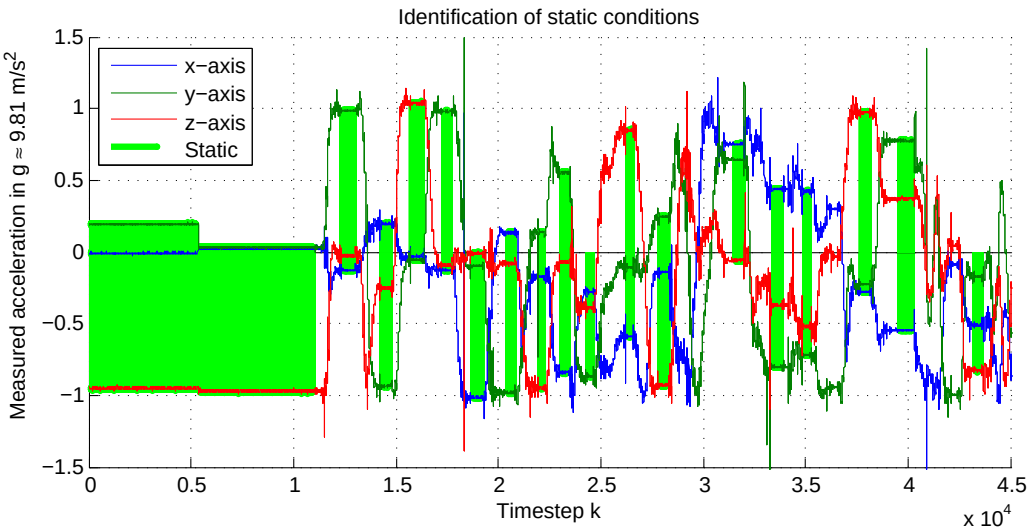


Figure 3.2: Signals of an exemplary calibration sequence and demonstration of the identification of static conditions. The blue, dark-green and red data represents the measured acceleration in x, y, z direction of the IMU. The bright-green data marks the identified static conditions.

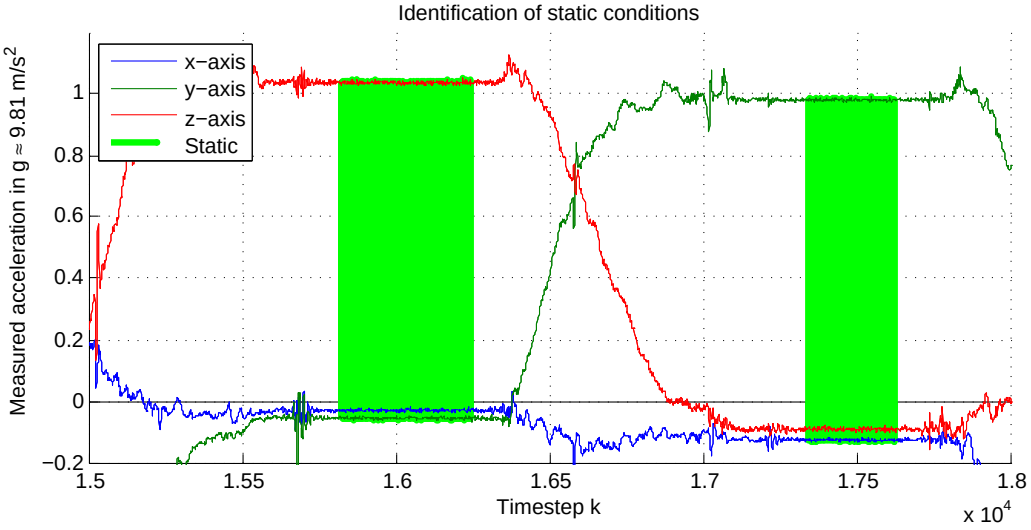


Figure 3.3: Zoomed section: Signals of an exemplary calibration sequence and demonstration of the identification of static conditions. The blue, dark-green and red data represents the measured acceleration in x, y, z direction of the IMU. The bright-green data marks the identified static conditions.

The obtained measurements for the n-th orientation $\tilde{\mathbf{a}}_n$ are used for the calibration method described in subsection 3.1.1.1. For the IMU, which has been used for the experiments of this thesis, the following calibration parameters for the accelerometer were found:

$$\mathbf{M}_a = \begin{bmatrix} +0.9914 & -0.0001 & -0.0014 \\ -0.0001 & +1.0036 & -0.0046 \\ -0.0014 & -0.0046 & +0.9947 \end{bmatrix}, \tag{3.24}$$

$$\mathbf{b}_{a,d} = \begin{pmatrix} -0.0982 \\ -0.0075 \\ +0.3313 \end{pmatrix} \mathrm{m/s^2}. \tag{3.25}$$

To demonstrate effect of the calibration procedure the difference between the measured and the true gravity will be introduced as new error-term:

$$\Delta l_k = g - ||\mathbf{a}_k|| = g - \sqrt{\left(\tilde{a}^B_{IB,c}\right)^2_{x,k} + \left(\tilde{a}^B_{IB,c}\right)^2_{y,k} + \left(\tilde{a}^B_{IB,c}\right)^2_{z,k}}. \tag{3.26}$$

The error for each orientation is presented in figure 3.4 and shows the improvement due to the calibration method. Furthermore the MSE before and after calibration is given in table 3.1 and confirms the successful calibration. The MSE is defined as:

$$\mathbf{MSE}(\Delta \mathbf{l}) = \frac{1}{m} \sum_{k=1}^{m} (\Delta l_k)^2 . \tag{3.27}$$

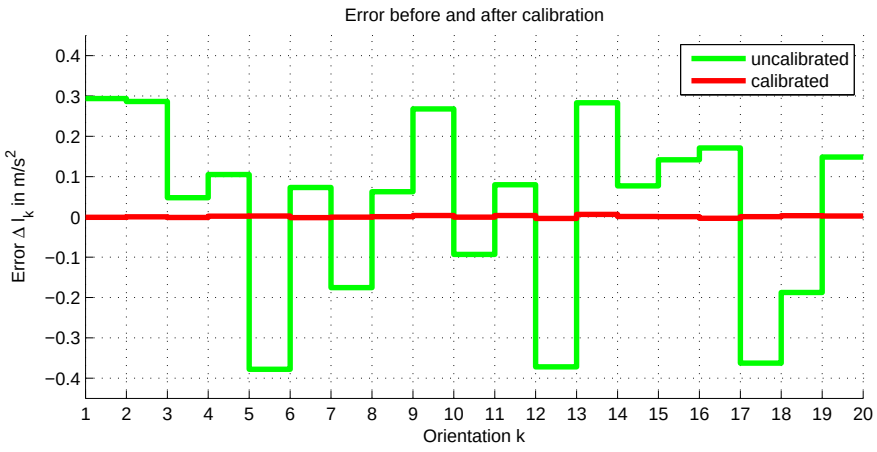


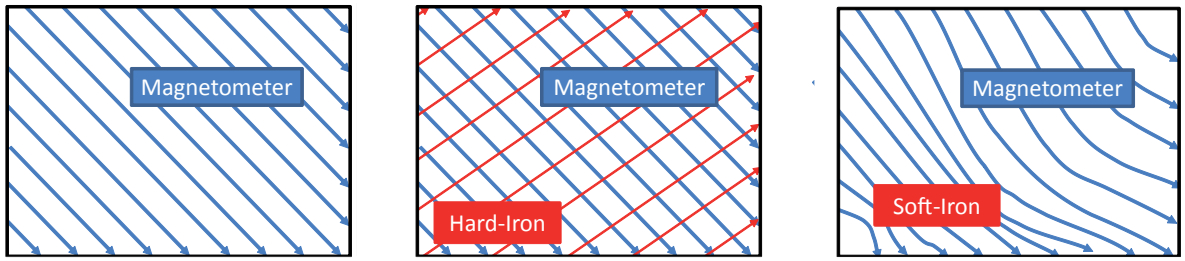
Figure 3.4: Comparison of error Δl_k before and after accelerometer calibration.

Table 3.1: Comparison of MSE between uncalibrated and calibrated accelerometer.

Method	**MSE** (Δ**l**)
uncalibrated	$51.4 \times 10^{-3} \left(\frac{\mathrm{m}}{\mathrm{s}^2}\right)^2$
calibrated	$7.76 \times 10^{-6} \left(\frac{\mathrm{m}}{\mathrm{s}^2}\right)^2$

3.1.2. Magnetometer

Measurement errors of magnetometers are not only due to imperfections of the device, but also due to the magnetic properties of its environment, which perturbs the magnetic field. These effects are called hard-iron and soft-iron distortions. In figure 3.5 the magnetic fields for a distortion-free, hard-iron and soft-iron distorted environment are illustrated.



(a) (b) (c)

Figure 3.5: Illustration of the hard and soft-iron distortions.(a) Distortion free magnetic field. (b) Hard-iron distorted magnetic field. (c) Soft-iron distorted magnetic field.

Hard-iron distortions are due to objects on the Printed Circuit Board (PCB) or in the vicinity of the sensor, which produce magnetic fields such as permanent magnets in speakers. Soft-iron distortions are due to unmagnetized ferromagnetic components on the PCB, which deflect or distort the magnetic field such as battery-packs. In sum these error sources act in the same way as bias and scaling errors [28]. For that reason the same model as defined in section 3.1 can be used to describe their influence.

3.1.2.1. Magnetometer Calibration Method

In order to compensate the erroneous measurement $\tilde{\mathbf{m}}_d^B$, the scale factor matrix $\mathbf{M}_m$ and deterministic bias $\mathbf{b}_{m,d}$ can to be estimated by using a similar assumption as in section 3.1.1.1. The magnitude of the magnetic field will be assumed to be constant. Furthermore, since the real

magnitude is not exactly known and because it does not contain information for the navigation, it will be normalized to 1:

$$\sqrt{\left(\tilde{m}_d^B\right)_x^2 + \left(\tilde{m}_d^B\right)_y^2 + \left(\tilde{m}_d^B\right)_z^2} = \text{const} = 1. \tag{3.28}$$

The error term $\mathbf{r}$ is introduced:

$$\mathbf{r} = [r_1, \ldots, r_m]^T, \tag{3.29}$$

with:

$$r_k = \left(\tilde{m}_{d,k}^B\right)_x^2 + \left(\tilde{m}_{d,k}^B\right)_y^2 + \left(\tilde{m}_{d,k}^B\right)_z^2 - 1. \tag{3.30}$$

Since the magnetometer is not perturbed by linear acceleration as it was the case for the accelerometer calibration, no preselection of static conditions is needed. The rest of the calibration method is analog to the accelerometer calibration in section 3.1.1.1.

3.1.2.2. Magnetometer Calibration

For calibration purposes, the IMU was slowly rotated by hand, such that the vector of the magnetometer measurement would cover the whole sphere. For the used IMU, the following parameters were found:

$$\mathbf{M}_m = \begin{bmatrix} 0.9393 & 0.0095 & 0.0115 \\ 0.0095 & 0.9467 & 0.0048 \\ 0.0115 & 0.0048 & 0.9920 \end{bmatrix}, \tag{3.31}$$

$$\mathbf{b}_{m,d} = \begin{pmatrix} -0.1857 \\ +0.0712 \\ -0.0564 \end{pmatrix}. \tag{3.32}$$

The difference between the magnitude of the corrected magnetometer measurement and the unity vector is introduced as new error term:

$$\Delta l_k = 1 - ||\mathbf{m}_k|| = 1 - \sqrt{(\tilde{m}_c^B)_x^2 + (\tilde{m}_c^B)_y^2 + (\tilde{m}_c^B)_z^2}, \tag{3.33}$$

$$\textbf{MSE}\ (\Delta \mathbf{l}) = \frac{1}{m} \sum_{k=1}^{m} (\Delta l_k)^2. \tag{3.34}$$

The errors of the whole calibration sequence is presented in figure 3.6 and shows the improvement due to the calibration method. Furthermore the MSE before and after calibration iss illustrated in table 3.2 and confirms the successful calibration.

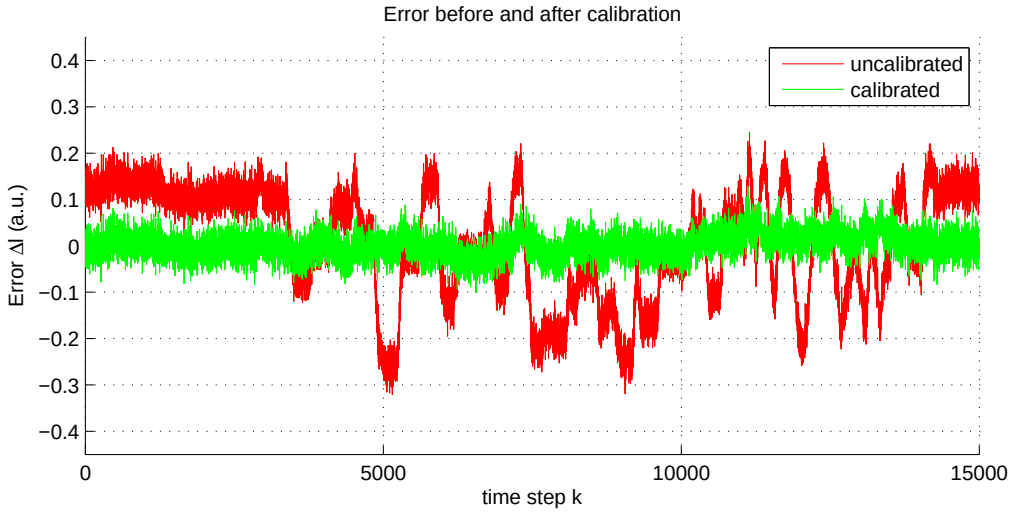


Figure 3.6: Comparison of error Δl_k before and after magnetometer calibration.

Table 3.2: Comparison of MSE between uncalibrated and calibrated magnetometer.

Method	**MSE** (Δ**l**)
uncalibrated	14.3×10^{-3} (a.u.)
calibrated	0.777×10^{-3} (a.u.)

3.1.3. Gyroscope

3.1.3.1. Gyroscope Calibration Methods

In order to obtain the compensated gyroscope measurement $\tilde{\boldsymbol{\omega}}^B_{IB,c}$ from the deterministic error affected measurement $\tilde{\boldsymbol{\omega}}^B_{IB,d}$, the scale factor matrix $\mathbf{M}_\omega$ and the deterministic bias $\mathbf{b}_{\omega,d}$ for the gyroscope have to be estimated.

Bias:

The bias $\mathbf{b}_{\omega,d}$ can be estimated by calculating the mean of the gyroscope measurements $\tilde{\boldsymbol{\omega}}^B_{IB,d}$, while the IMU is placed in a static condition:

$$\mathbf{b}_{\omega,d} = \frac{1}{N} \cdot \sum_{k=1}^{N} \tilde{\boldsymbol{\omega}}^B_{IB,d,k}. \tag{3.35}$$

Scale Factor Matrix:

For the sake of readability the indexes will be omitted such that $\boldsymbol{\omega}^B_{IB} = \boldsymbol{\omega}$.

The estimation of the scale factor matrix $\mathbf{M}_\omega$ is based on the following assumption:

The IMU is only exposed to pure rotations and no acceleration. Therefore the accelerometer only measures gravity. While rotating the IMU from time t_1 to t_2, the accelerometer observes a rotating gravity vector, with respect to the body-frame. The same rotation is measured by the gyroscope. If the coordinate axes of the gyroscope and the accelerometer are aligned and the IMU delivers perfect measurements, the rotation vector of the gyroscope $\boldsymbol{\sigma}^{t_2}_{t_1\text{gyro}}$ and the accelerometer $\boldsymbol{\sigma}^{t_2}_{t_1\text{gravity}}$ are equal:

$$\boldsymbol{\sigma}^{t_2}_{t_1\text{gyro}} = \boldsymbol{\sigma}^{t_2}_{t_1\text{gravity}}. \tag{3.36}$$

In order to obtain the rotation vector of the gyroscope $\boldsymbol{\sigma}^{t_2}_{t_1\text{gyro}}$, the angular rates of the gyroscope have to be accumulated. As already described in section 2.4, the accumulation of angular rates is quite complicated. However, if the axis of the rotation does not change between t_1 and t_2, the calculation of the rotation vector simplifies to

$$\boldsymbol{\sigma}^{t_2}_{t_1\text{gyro}} = \int_{t_1}^{t_2} \tilde{\boldsymbol{\omega}}_c dt = \sum_{k=1}^{N} \tilde{\boldsymbol{\omega}}_{c,k} \cdot \Delta T. \tag{3.37}$$

Rewriting this equation using the error-model in equation (3.1) yields to

$$\sigma_{t_1\,\text{gyro}}^{t_2} = \sum_{k=1}^{N} \mathbf{M}_\omega \cdot (\tilde{\boldsymbol{\omega}}_{d,k} - \mathbf{b}_{\omega,d}) \cdot \Delta T \tag{3.38}$$

$$= \mathbf{M}_\omega \Delta T \cdot \sum_{k=1}^{N} (\tilde{\boldsymbol{\omega}}_{d,k} - \mathbf{b}_{\omega,d}) \tag{3.39}$$

$$= \mathbf{M}_\omega \cdot \boldsymbol{\Delta\Phi}, \tag{3.40}$$

with

$$\boldsymbol{\Delta\Phi} = \sum_{k=1}^{N} (\tilde{\boldsymbol{\omega}}_{d,k} - \mathbf{b}_{\omega,d})\, \Delta T. \tag{3.41}$$

The error term $\mathbf{e}_j$ for the j- experiment can be formulated as:

$$\mathbf{e}_j = \mathbf{M}_\omega \cdot \boldsymbol{\Delta\Phi}_j - \sigma_{\text{gravity},j}, \tag{3.42}$$

where $\boldsymbol{\Delta\Phi}_j$ is the accumulated rotation vector of the gyroscope for the j-th experiment and $\sigma_{\text{gravity,j}}$ is the rotation vector calculated from the accelerometer measurements for the j-th experiment. The parameters $S_{xx}, S_{yy}, S_{zz}, S_{xy}, S_{xz}, S_{yx}, S_{yz}, S_{zx}, S_{zy}$ of $\mathbf{M}_\omega$ can be found by solving the linear least square problem minimizing the following cost function:

$$J(S_{xx}, S_{yy}, S_{zz}, S_{xy}, S_{xz}, S_{yx}, S_{yz}, S_{zx}, S_{zy}) = \sum_{j=1}^{m} \mathbf{e}_j^T \cdot \mathbf{e}_j. \tag{3.43}$$

Calculation of the Rotation Vector σ_{gravity}:

For the calculation of the rotation vector the measured accelerations during rotations are assumed to be caused solely by gravity and therefore accelerations due to movements will be neglected. The calculation can be described in two major steps:

- **Axis-Calculation**

  While rotating, the observed acceleration vector describes an arc in the 3-dimensional space (see figure 3.7). Furthermore, all points lie on the same plane, if the axis of rotation does not change. The axis of rotation is equal to the normal of the plane fitted through the arc.

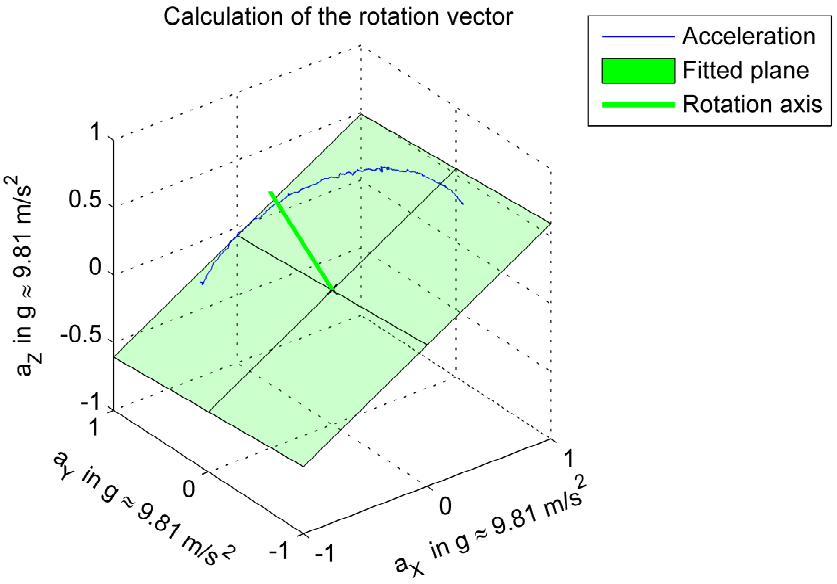


Figure 3.7: Accelerometer measurements describing an arc due to the rotation of the gravity vector.

A point $\mathbf{P}$ on a plane can be described by [29]:

$$\mathbf{P} = \mathbf{p}_0 + c_1\mathbf{v_1} + c_2\mathbf{v_2}. \tag{3.44}$$

The plane can be calculated with the algorithm described in [29]. This algorithm minimizes the orthogonal distance, which is given as:

$$\min\left(\sum_{i=1}^{N}\left((\mathbf{p}_i - \mathbf{p}_0)^T\mathbf{n}\right)^2\right). \tag{3.45}$$

$\mathbf{p}_i$, $\mathbf{p}_0$, $\mathbf{n}$, are a set of N points, a point on the plane and the plane's normal, respectively. $\mathbf{v}_1$ and $\mathbf{v}_2$ are the orthogonal basis of the fitted plane.

- **Angle-Calculation:**

  The angle of rotation is given as the angle of the arc, which is projected onto the fitted plane. In figure 3.8 the result of the circle fit, including the angle calculation, based on the projected points, is demonstrated.

  The projected points $\mathbf{p}_{\text{proj}}$ are calculated by using orthonormal projection according to [29]:

$$\mathbf{p}_{\text{proj}} = (\mathbf{p}_i - \mathbf{p_0}) \cdot \mathbf{v}_1 + (\mathbf{p}_i - \mathbf{p_0}) \cdot \mathbf{v}_2. \tag{3.46}$$

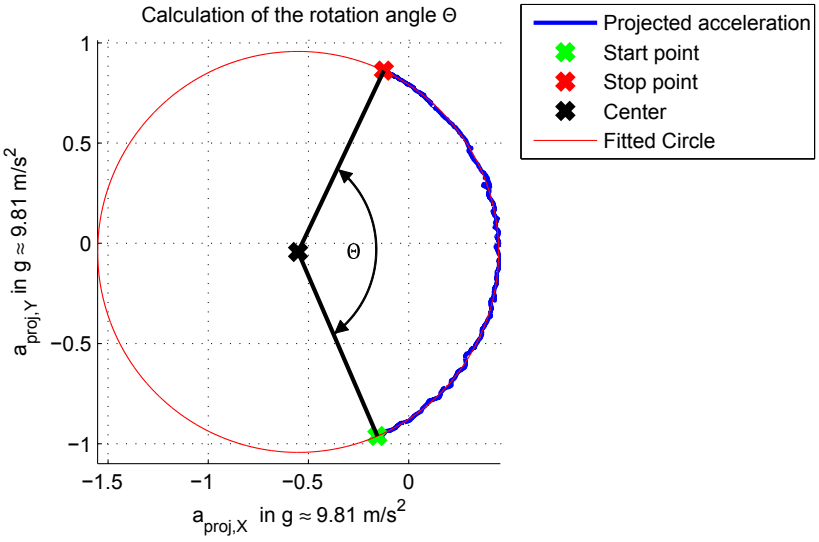


Figure 3.8: Circle fitting to calculate the angle of rotation Θ.

The radius R and center $\mathbf{c} = \begin{bmatrix} x_0 & y_0 \end{bmatrix}^T$ of this circle can be obtained by solving the following overdetermined linear system, which minimizes the orthogonal distance [30]:

$$\begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \\ -2x_m & -2y_m & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ x_0^2 + y_0^2 - R^2 \end{bmatrix} = \begin{bmatrix} -x_1^2 - y_1^2 \\ -x_2^2 - y_2^2 \\ \vdots \\ -x_m^2 - y_m^2 \end{bmatrix}. \tag{3.47}$$

In equation (3.47) x_i and y_i are the components of the i-th projected point.
The rotation angle Θ can then be calculated as:

$$\Theta = \cos\left(\frac{(\mathbf{r}_{\text{start}} - \mathbf{c}) \cdot (\mathbf{r}_{\text{stop}} - \mathbf{c})}{|\,(\mathbf{r}_{\text{start}} - \mathbf{c})\,|\,|\,(\mathbf{r}_{\text{stop}} - \mathbf{c})\,|} \right)^{-1}, \tag{3.48}$$

with $\mathbf{r}_{\text{start}}$ and $\mathbf{r}_{\text{stop}}$ as the starting and stopping point of the rotation.

Finally the rotation vector for the j-rotation is obtained as:

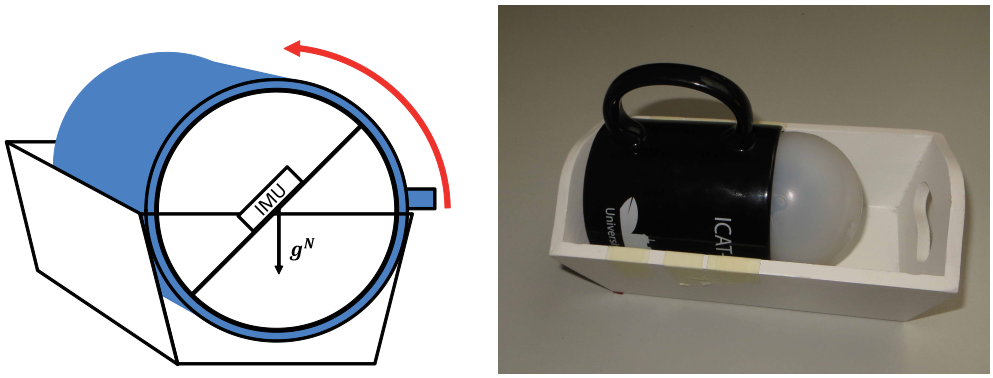
$$\sigma_{\text{gravity},j} = \mathbf{n}_j \Theta_j. \tag{3.49}$$

3.1.3.2. Gyroscope Calibration

For the derivation of the gyroscope calibration, the following assumptions have been made and have to be ensured:

- The axis of rotation does not change.
- The IMU is only exposed to rotations and not to accelerations, such that only pure gravity is measured.

Usually expensive measurement equipment such as a turntable is needed to guarantee these conditions. However such equipment was not available and therefore a simple and inexpensive calibration procedure was invented, which also allows in field-calibrations. Similar to the calibration procedure for the accelerometer, which was described in section 3.1.1.1, a cup and a V-block is needed. For this experiment a simple V-shaped pepper and salt box has been used. The calibration procedure will be explained hereinafter:
The ball containing the rigidly mounted IMU at the center is placed in a cup to ensures that no movements or rotations would perturb the signals during measurements Analog to the accelerometer calibration, the ball containing the IMU was placed in the cup pointing in a different direction for each rotational experiment. This cup was then rotated in a V-block to ensure a constant axis of rotation. The calibration setup is illustrated in figure 3.9.



(a)Theoretical calibration setup. (b)Used calibration setup.

Figure 3.9: Calibration setup for gyroscope calibration.

The following results for the bias $\mathbf{b}_{\omega,d}$ and the scale factor matrix $\mathbf{M}_{\omega}$ were obtained.

$$\mathbf{b}_{\omega,d} = \begin{bmatrix} -0.206 \\ +0.183 \\ -0.070 \end{bmatrix} \frac{\text{deg}}{\text{s}}, \tag{3.50}$$

$$\mathbf{M}_\omega = \begin{bmatrix} +1.0088 & -0.0007 & +0.0050 \\ -0.0021 & +1.0106 & +0.0137 \\ +0.0088 & +0.0170 & +0.9972 \end{bmatrix}. \tag{3.51}$$

The error $||e_k||$ (according to equation (3.42)) for the j-th rotation test is presented in figure 3.10 and shows the improvement due to the calibration. It should be noted, that the bias was already subtracted from the uncalibrated data and that the angle of rotation was about 130 deg.

The MSE is given in table 3.3 and was defined as:

$$MSE(e_k) = \frac{1}{m} \sum_{k=1}^{m} \mathbf{e}_k^T \cdot \mathbf{e}_k. \tag{3.52}$$

Table 3.3: Comparison of MSE between uncalibrated and calibrated gyroscope.

Method	**MSE** (e_k)
uncalibrated	5.01 deg^2
calibrated	0.0688 deg^2

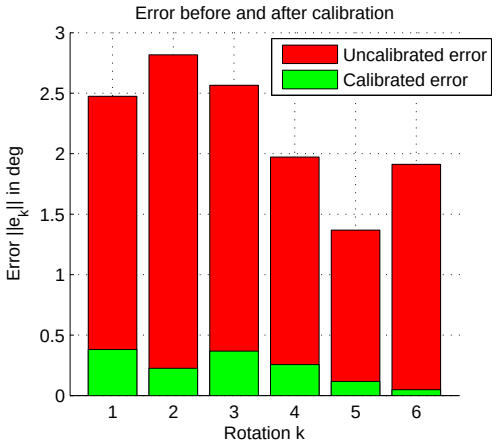


Figure 3.10: Comparison of error $||e_k||$ before and after gyroscope calibration.

3.2. Random-Errors

In addition to deterministic errors, random disturbances also affect the measurements. The nature of random errors cannot be described in a deterministic sense. However, using probability and statistic, their behavior can be described [20].
The following model has frequently been used to describe the measurement $\tilde{\mathbf{x}}$, which is affected by random errors [14], [31]:

$$\tilde{\mathbf{x}} = \tilde{\mathbf{x}}_c + \mathbf{b} + \boldsymbol{\nu}, \tag{3.53}$$

where $\mathbf{b}$ and $\boldsymbol{\nu}$ are the drifting bias and measurement noise, respectively. The term $\tilde{\mathbf{x}}_c$ represents the measurement, which has already been compensated for deterministic errors.
It should be noted, that in the compensation step for deterministic errors, the raw values have been multiplied with the scale factor matrix. This introduces correlations between the axes. However, due to the dominant diagonal behavior of the scale factor matrix, the introduced correlation is neglected.

3.2.1. Measurement Noise

The measurement noise is modeled as zero mean white Gaussian noise $\nu \sim \mathcal{N}(0, \sigma_\nu^2)$ and reflects the uncertainty due to sensor noise itself and unmodeled behavior of the sensors. It should be noted, that unmodeled behavior of sensors such as the nonlinearity, are deterministic errors. However, with the proposed calibration procedure this effect cannot be estimated. A general approach to still consider these effects, is to increase the uncertainty of the measurement by assuming additional noise. Analysis of the data sheet of the IMU [18] revealed, that the nonlinearity strongly affects the sensory output. For that reason the nonlinearity is modeled as additional measurement noise ν_{NL}. The complete measurement noise is given as:

$$\nu = \nu_{\mathrm{NL}} + \nu_{\mathrm{noise}}. \tag{3.54}$$

Modeling the error sources as independent and normally distributed:

$$\sigma_\nu^2 = \sigma_{\mathrm{NL}}^2 + \sigma_{\mathrm{noise}}^2. \tag{3.55}$$

The sensor noise can be estimated using Allan variance analysis (see section 3.2.3). The nonlinearity (NL) will be modeled as normally distributed noise with the following assumption:
If the sensor output is zero, also the nonlinearity will be zero.

Therefore the variance of the NL will be linearly scaled with the magnitude of the measurement $|x|$ and can be expressed as:

$$\sigma_{\mathrm{NL}}^2 = \left(\frac{\mathrm{NL}(\%)}{100} \cdot \mathrm{FSR}\right)^2 \frac{|x|}{0.5 \cdot \mathrm{FSR}} \tag{3.56}$$

$$= \sigma_{\mathrm{NL},0}^2 \cdot \frac{|x|}{0.5 \cdot \mathrm{FSR}}, \tag{3.57}$$

with NL(%) as the nonlinearity, which is usually expressed in % of the Full Scale Range (FSR).

3.2.2. Bias Drift

For modeling the drifting bias the following three approaches have been frequently used [32]:

- Random walk
- 1^{st} order Gauss Markov process
- Autoregressive model

These stochastic processes are well described in [16] and [32]. The autoregressive (AR) model represents the most general approach [33], whereas the random walk and 1^{st} order Gauss Markov process are special cases of the AR model.
For this thesis a random walk (RW) will be used to model the drifting bias. The integration of uncorrelated white noise results in a RW and can be modeled with the following discrete time function [34]:

$$b_k = b_{k-1} + w_{\mathrm{RW}}, \tag{3.58}$$

with b_k as the drifting bias at time step k. w_{RW} represents the noise driving the random walk and is modeled as white Gaussian noise with the variance σ_{RW}^2. This variance can be estimated using Allan variance analysis (see section 3.2.3).

3.2.3. Allan Variance Analysis

In this section the estimation of the random error sources, such as RW and white sensor noise, will be presented. Allan variance analysis is a common method to identify the variances of random errors [33].

The computation of the Allan variance (AVAR) is given as [33]:

The measurement for the AVAR is collected over a long period of time, while the IMU is placed in a stable position. For a chosen cluster time $\tau = n \cdot \Delta T$, the discrete time series with length N is divided into m consecutive clusters of size n. ΔT represents the sampling time. For each cluster a mean is calculated, in order to obtain a series of mean: $[\Theta_1(\tau), \Theta_2(\tau), \cdots, \Theta_m(\tau)]$.

The AVAR for a given τ is then calculated as:

$$\sigma^2_{\text{Allan}}(\tau) = \frac{1}{2(m-1)} \sum_{k=1}^{m-1} (\Theta_{n+1} - \Theta_n)^2 . \tag{3.59}$$

The AVAR is computed for several τ. The square root of AVAR, which is called Allan deviation, is plotted against the cluster time τ in a log-log-plot. This function is called Allan deviation function.

Allan variance analysis uses the idea, that different noise terms appear in different regions of τ and therefore are easy to determine. A typical Allan deviation function for a gyroscope, showing all noise terms specified in [35], is presented in figure 3.11.

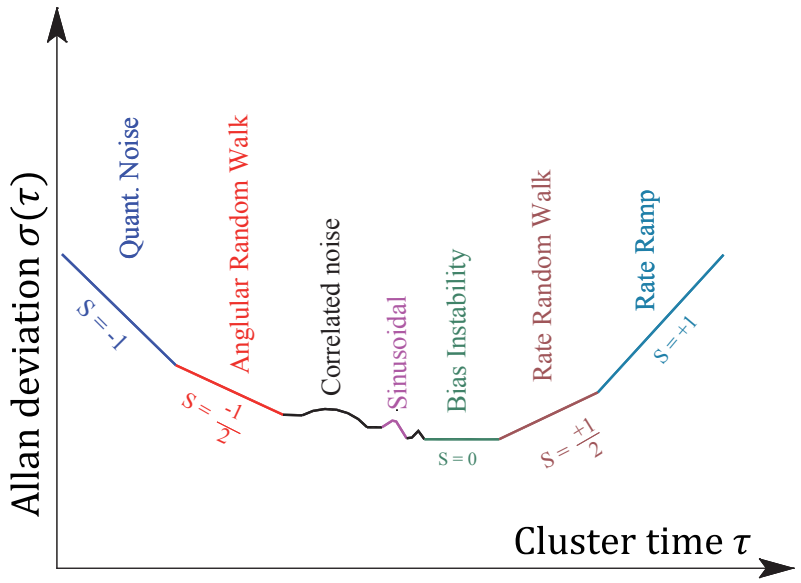


Figure 3.11: Typical Allan deviation plot for different noise sources, adapted from [33].

In figure 3.12 the Allan deviation function for the gyroscope in the MPU9150 for the z-axis is illustrated. Furthermore the approach to estimate the values K_{Allan} and N_{Allan} is demonstrated.

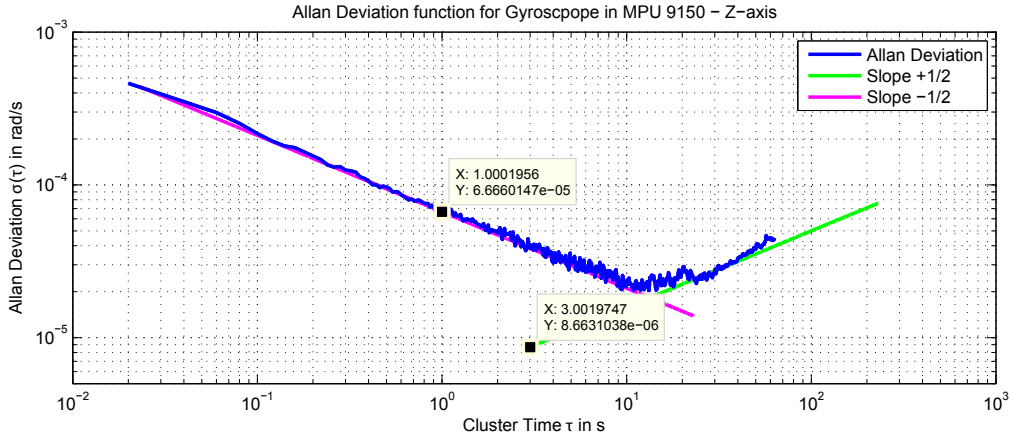


Figure 3.12: Allan deviation plot for the gyroscope in the MPU 9150 for z-axis. The values for N_{Allan} and N_{Allan} can be obtained by reading the value at $\tau = 1$ and $\tau = 3$ for the negative and positive slope, respectively. A value of $N = 6.67 \times 10^{-5} \frac{\text{rad}}{\text{s}\sqrt{\text{Hz}}}$ and $K = 8.66 \times 10^{-6} \frac{\text{rad}\sqrt{\text{Hz}}}{\text{s}}$ was found.

By fitting a line to the slope of $-\frac{1}{2}$ and reading the value at $\tau = 1$, the value N_{Allan} is obtained. N_{Allan} relates to the variance of the sensor noise as follows:

$$\sigma^2_{\text{noise}} = \frac{N^2_{\text{Allan}}}{\Delta T}. \tag{3.60}$$

In the same way, by fitting a line to the slope of $+\frac{1}{2}$ and reading the value at $\tau = 3$, the value K_{Allan} is obtained. Here K_{Allan} relates to the variance of the RW as follows:

$$\sigma^2_{\text{RW}} = \Delta T \cdot K^2_{\text{Allan}}. \tag{3.61}$$

3.2.4. Numerical Values

Numerical values for modeling random errors of the sensor *MPU-9150*, which has been used in the experiments for this thesis, are given in table 3.4, table 3.5 and table 3.6. It should be noted that the values of the magnetometer have been normalized to a magnitude of 1, as a result of the magnetometer calibration. Values for the NL are taken from the data-sheet [18].

Table 3.4: Noise characteristics for the MPU-9150-Gyroscope.

		f_s	NL	FSR	$\sigma^2_{\text{NL},0}$	N_{Allan}	σ^2_{noise}	K_{Allan}	σ^2_{RW}
		Hz	%	$\frac{\text{rad}}{\text{s}}$	$\frac{\text{rad}^2}{\text{s}^2}$	$\frac{\text{rad}}{\text{s}\sqrt{\text{Hz}}}$	$\frac{\text{rad}^2}{\text{s}^2}$	$\frac{\text{rad}\sqrt{\text{Hz}}}{\text{s}}$	$\frac{\text{rad}^2}{\text{s}^2}$
Gyroscope	x	500	0.2	34.91	4.90×10^{-3}	6.25×10^{-5}	1.95×10^{-6}	6.44×10^{-5}	8.29×10^{-12}
	y	500	0.2	34.91	4.90×10^{-3}	7.69×10^{-5}	2.96×10^{-6}	1.58×10^{-5}	5.00×10^{-13}
	z	500	0.2	34.91	4.90×10^{-3}	6.67×10^{-5}	2.22×10^{-6}	8.66×10^{-6}	1.50×10^{-13}

Table 3.5: Noise characteristics for the MPU-9150-Accelerometer.

		f_s	NL	FSR	$\sigma^2_{\text{NL},0}$	N_{Allan}	σ^2_{noise}	K_{Allan}	σ^2_{RW}
		Hz	%	$\frac{\text{m}}{\text{s}^2}$	$\frac{\text{m}^2}{\text{s}^4}$	$\frac{\text{m}}{\text{s}^2\sqrt{\text{Hz}}}$	$\frac{\text{m}^2}{\text{s}^4}$	$\frac{\text{m}\sqrt{\text{Hz}}}{\text{s}^2}$	$\frac{\text{m}^2}{\text{s}^4}$
Accelerometer	x	500	0.5	39.24	3.85×10^{-2}	1.53×10^{-3}	1.17×10^{-3}	4.92×10^{-4}	4.85×10^{-10}
	y	500	0.5	39.24	3.85×10^{-2}	1.31×10^{-3}	8.64×10^{-3}	1.41×10^{-4}	3.99×10^{-11}
	z	500	0.5	39.24	3.85×10^{-2}	1.96×10^{-3}	1.92×10^{-3}	3.06×10^{-4}	1.87×10^{-10}

Table 3.6: Noise characteristics for the MPU-9150-Magnetometer.

		f_s	N_{Allan}	σ^2_{noise}	K_{Allan}	σ^2_{RW}
		Hz	*a.u.*	*a.u.*	*a.u.*	*a.u.*
Magnetometer	x	125	2.12×10^{-3}	5.62×10^{-4}	1.28×10^{-3}	1.31×10^{-7}
	y	125	2.12×10^{-3}	5.62×10^{-4}	5.66×10^{-4}	2.56×10^{-9}
	z	125125	2.12×10^{-3}	5.62×10^{-4}	7.40×10^{-4}	4.38×10^{-9}

4. Monocular Position Tracking

In this chapter the scheme to estimate the 3-dimensional position of a spherical object, using a monocular camera will be described. In the first part the general idea and concept will be presented. Furthermore the challenges and factors, which corrupt and degrade the results, are pointed out. The second part will provide a detailed procedure to solve these problems.

The reader is reminded that the main focus of this thesis was on the development of the sensor fusion framework. The monocular position tracking system represents only a side path and therefore this content is presented in a more relaxed way.

4.1. Introduction

The concept for the monocular position tracking using spheres is presented in figure 4.1 and can be summarized as follows:

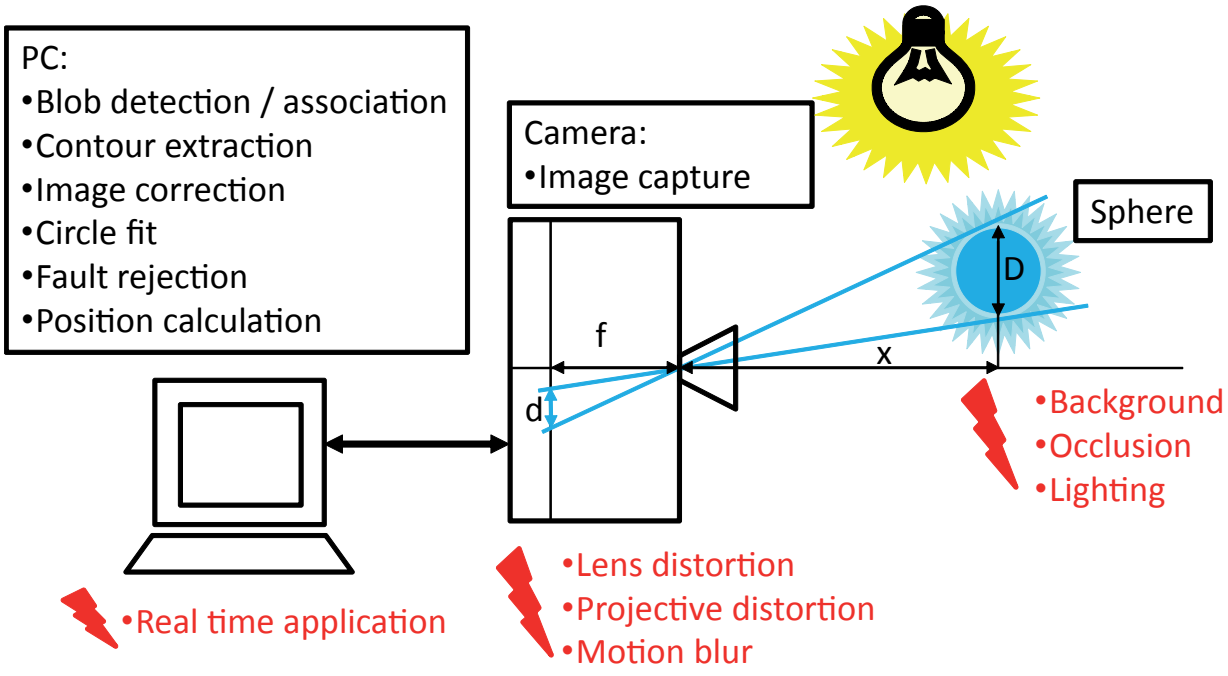


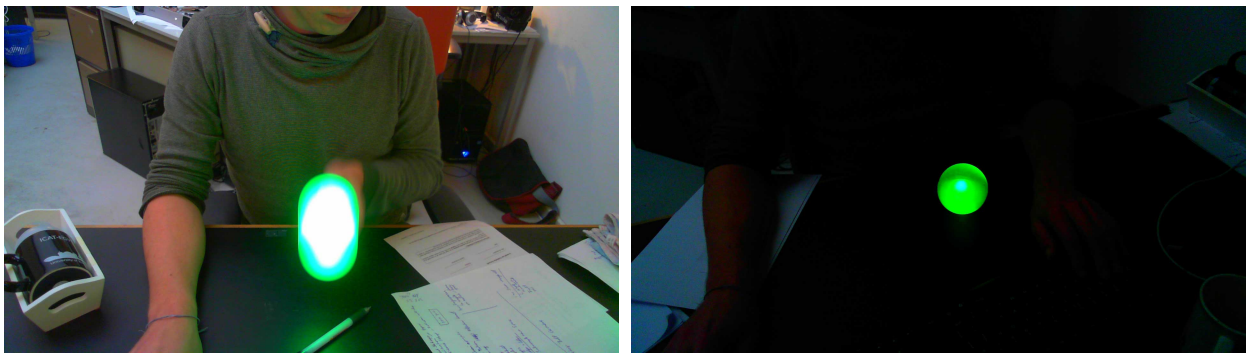
Figure 4.1: Main steps for monocular position tracking of a sphere.

A camera captures the actively illuminated sphere and the sphere's outline is projected on the image plane as an ellipse. If the parameters of projected ellipse, the properties of camera and the diameter of the sphere are known, the 3D position can be calculated using trigonometry.
However, the identification of the ellipse and its properties are aggravated due to occlusions of the sphere, or by objects with similar color or shape. Also reflections of the sphere itself could create substantial perturbations. Furthermore, the color of the sphere and its representation in the image is influenced by illumination of the surrounding scene and the illumination of the sphere itself.
Moreover, a realistic camera contains lens flaws, which distort the image. In addition the sensitivity of the sensor and the shutter speed are limited. These factors lead to motion blur in scenes of fast moving objects.
The challenge of the procedure is to identify the correct blob of the projected sphere, in order to extract its contour. After correcting lens distortion and projective distortions, a circle fit to the contour is performed. If the calculated parameters satisfy fault rejection criteria, the 3D position will be calculated. In addition, special care has to be taken to minimize the computational burden, so real time application is possible.

4.2. Procedure

4.2.1. Image Capture

The sphere is actively illuminated using LEDs, which are mounted inside the sphere. The color of the LEDs in the sphere can be chosen, in order to ease the correct identification of the sphere and to allow tracking of multiple spheres. Because of the active illumination, the exposure time can be reduced. This yields to a significant reduction of motion blur and to a higher contrast between foreground and background. Also, the influence of the surrounding light is reduced. In figure 4.2 motion blur and the effect of reducing exposure time is demonstrated.



(a) (b)

Figure 4.2: (a) Motion blur for high exposure time, (b) Adjusted exposure time.

4.2.1.1. Automatic Lighting Adjustment

As already mentioned, the lighting conditions strongly influence the image. Therefore correct camera settings are crucial. To ease the adjustment of the camera, a simple automatic lighting adjustment algorithm has been developed.

The main parameters affecting the lighting condition for a low cost video camera are exposure and brightness. The exposure setting directly changes camera properties and shows the strongest effect. The brightness setting only changes video filter properties and will be used for fine tuning. In order to adjust the lighting, the following assumption is made: There is only one illuminated sphere with a diameter of 80 mm with a distance of 1 m to the camera. Manual testing showed, that a good lighting condition has been found, if about 3% of all pixels are located in the last 10 bins of the histogram of any color. The algorithm consists of the following steps:

1. Calculate the histogram for each color.
2. Calculate the percent of pixels in the last $N = 10$ bins for each color and set p_{last} to the maximum of all three colors.
3. Update the brightness B as follows:

$$B = \begin{cases} B + 5 & \text{if } p_{\text{last}} \leq 2 \\ B - 5 & \text{if } p_{\text{last}} > 4 \\ B & \text{else.} \end{cases} \tag{4.1}$$

4. If the brightness exceeds a lower $B_{\text{min}} = 30$ or upper $B_{\text{max}} = 250$ threshold, the brightness

B and exposure E are again updated as follows:

$$\begin{aligned} &\text{if } B > B_{\max}: \\ &\qquad B = B_{\min} \\ &\qquad E = E + 1, \end{aligned} \tag{4.2}$$

$$\begin{aligned} &\text{if } B < B_{\min}: \\ &\qquad B = B_{\max} \\ &\qquad E = E - 1. \end{aligned} \tag{4.3}$$

5. The iteration is stopped if the brightness does not change anymore.

It should be noted that the values were manually tuned and are specific for the camera model. The lighting adjustment was usually performed once before the tracking. A typical image after lighting adjustment is given in figure 4.2.

4.2.2. Kalman Filter to Decrease Computational Burden

The tracked object only appears in a small part of the whole image. In order to reduce the amount of processing pixels, a search region based on the predicted estimations of a Kalman filter will be used. A detailed discussion on Kalman filters can be found in chapter 5. A constant velocity model represents the dynamic of the object and is given as the following discrete time state space model according to [22]:

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{v} \end{bmatrix}_{k+1} = \begin{bmatrix} \mathbf{1} & \Delta\mathbf{T} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \end{bmatrix}_{k} + \begin{bmatrix} \Delta\mathbf{T}^2 \\ \Delta\mathbf{T} \end{bmatrix} \cdot \mathbf{n_a}, \tag{4.4}$$

$$\mathbf{y}_{k+1} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \end{bmatrix}_{k+1} + \mathbf{v}_p, \tag{4.5}$$

with $\mathbf{p}$ and $\mathbf{v}$ as the 3D position and velocity vector, respectively. The unobserved acceleration is modeled as zero mean white Gaussian process noise $\mathbf{n}_a$ with covariance $\mathbf{Q}_a$. The measurement $\mathbf{y}_{k+1}$ is disturbed by zero mean white Gaussian measurement noise $\mathbf{v}_p$ with covariance $\mathbf{R}_p$. Since the purpose of this KF is rather to provide an appropriate search region, than an optimal estimation of the position, the covariance matrices $\mathbf{Q}_a, \mathbf{R}_p$ are manually tuned.
The search region presented in pixels u, v is given as:

$$\begin{bmatrix} u_{\min}, & u_{\max} \\ v_{\min}, & v_{\max} \end{bmatrix} = \begin{bmatrix} u_{\text{pred}} + 3 \cdot \Delta u - (r + 3 \cdot \Delta r), & u_{\text{pred}} - 3 \cdot \Delta u + (r + 3 \cdot \Delta r) \\ v_{\text{pred}} + 3 \cdot \Delta v - (r + 3 \cdot \Delta r), & v_{\text{pred}} - 3 \cdot \Delta u + (r + 3 \cdot \Delta r) \end{bmatrix}. \tag{4.6}$$

The projection of the objects center $[u_{\text{pred}}, v_{\text{pred}}]^T$ and the radius r is given as:

$$\begin{bmatrix} u_{\text{pred}} \\ v_{\text{pred}} \\ 1 \end{bmatrix} = \frac{\mathbf{M}_{\text{cam}} \cdot \mathbf{p}}{z}, \tag{4.7}$$

$$r = \frac{f \cdot R}{z}, \tag{4.8}$$

with $\mathbf{M}_{\text{cam}}$, f, z and R as the intrinsic camera matrix, focal length, z position and the radius of the object, respectively. The deviations $\Delta u, \Delta v, \Delta r$ are given as:

$$\begin{bmatrix} \Delta u \\ \Delta v \\ \Delta r \end{bmatrix} = \sqrt{\text{diag}(\mathbf{J} \cdot \mathbf{p} \cdot \mathbf{J}^T)}, \tag{4.9}$$

with $\mathbf{J}$ as the Jacobian of equation (4.7) and equation (4.8). In figure 4.3 a typical search region, which is marked as red box, is presented.

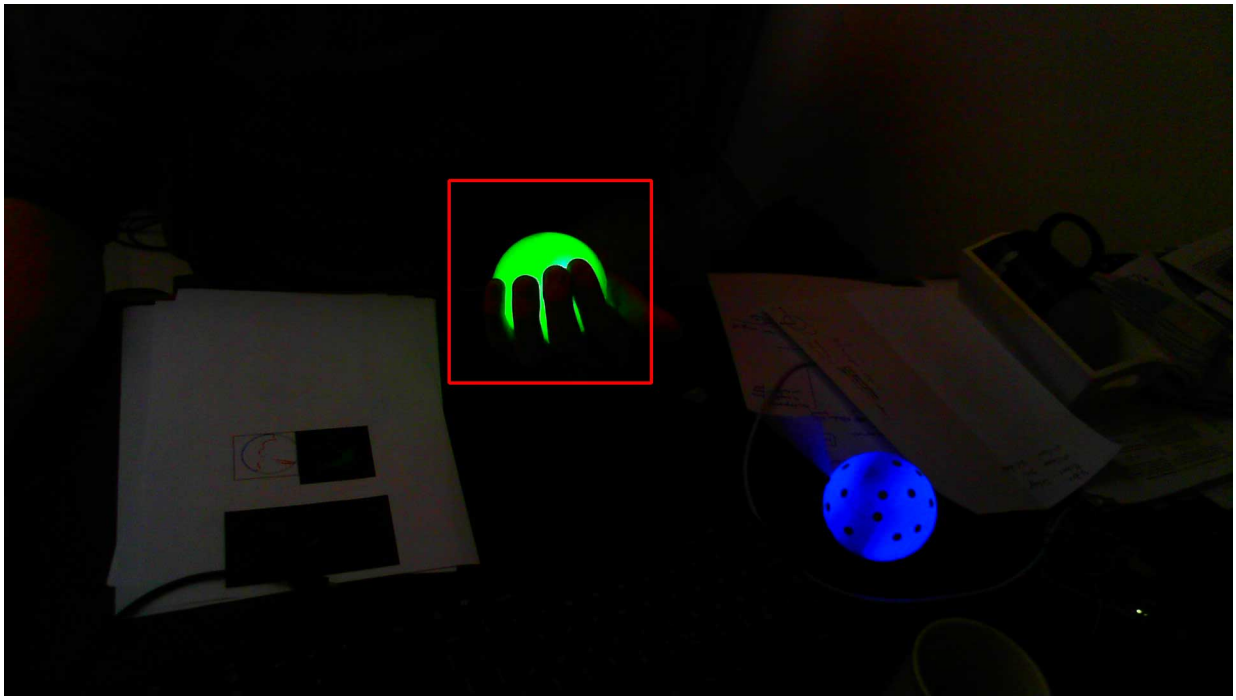


Figure 4.3: Predicted search region based on the Kalman filter for monocular position tracking. The search region is presented as red box, whereas the object to be tracked is green.

4.2.3. Blob Detection / Association

The next step is to detect the correct blob of the sphere's projection within the search region. In the field of image processing a blob is referred as a region of an image with similar properties (E.g. the projection of the ball). The following points explain the blob detection and association to find the correct blob:

- **Subsample**

  When starting tracking and after periods of lost objects only little or no information of the balls location is known and therefore the search region can become very large. In the worst case it can be as large as the complete image. This results in a very large number of pixels to be processed resulting in drastic increase of the computational burden. In order to limit the pixel number, the search region will be subsampled to a predefined image size. When the lost object is found, the search region is small enough and no subsampling is needed. Hence, this step enables faster detections of lost objects without affecting the accuracy of tracking.

- **Color Segmentation**

  Since the color of the object is known, the associated blob can be detected using color segmentation. For that reason the image is transformed from the Red/Green/Blue-space (RGB) to the Hue/Saturation/Value-space (HSV). The color representation in HSV-space is known to be robust to lighting effects and presents a very intuitive interpretation of the color given in the hue-channel. A good introduction to color-spaces is given in [36] and the HSV-transformation is given by [37].

  Blob-candidates can be found by thresholding each channel and then logically combining them. A candidate pixel is found if following three criteria are met:

$$I_{\text{Hue,min}} > I_{\text{Hue}}(u,v) > I_{\text{Hue,max}}, \tag{4.10}$$

$$I_{\text{Sat,min}} > I_{\text{Sat}}(u,v) > I_{\text{Sat,max}}, \tag{4.11}$$

$$I_{\text{Val,min}} > I_{\text{Val}}(u,v) > I_{\text{Val,max}}, \tag{4.12}$$

  with $I_{\text{Hue,min}}$,$I_{\text{Sat,min}}$, $I_{\text{Val,min}}$, $I_{\text{Hue,max}}$, $I_{\text{Sat,max}}$, $I_{\text{Val,max}}$ as the lower and upper thresholds for Hue, Saturation and Value, respectively. $I_{\text{Hue}}(u,v)$, $I_{\text{Sat}}(u,v)$, $I_{\text{Val}}(u,v)$ are the values for Hue, Saturation and Value for the pixel u,v respectively. The thresholds for hue are chosen according to the sphere's color. The thresholds for value and saturation are less

important; they are used to filter out background.

Typical values for a green sphere are:

$$\begin{bmatrix} I_{\text{Hue,min}}, & I_{\text{Hue,max}} \\ I_{\text{Sat,min}}, & I_{\text{Sat,max}} \\ I_{\text{Val,min}}, & I_{\text{Val,max}} \end{bmatrix} = \begin{bmatrix} 40, & 85 \\ 35, & 255 \\ 90, & 255 \end{bmatrix}. \tag{4.13}$$

- **Image Cleanup**

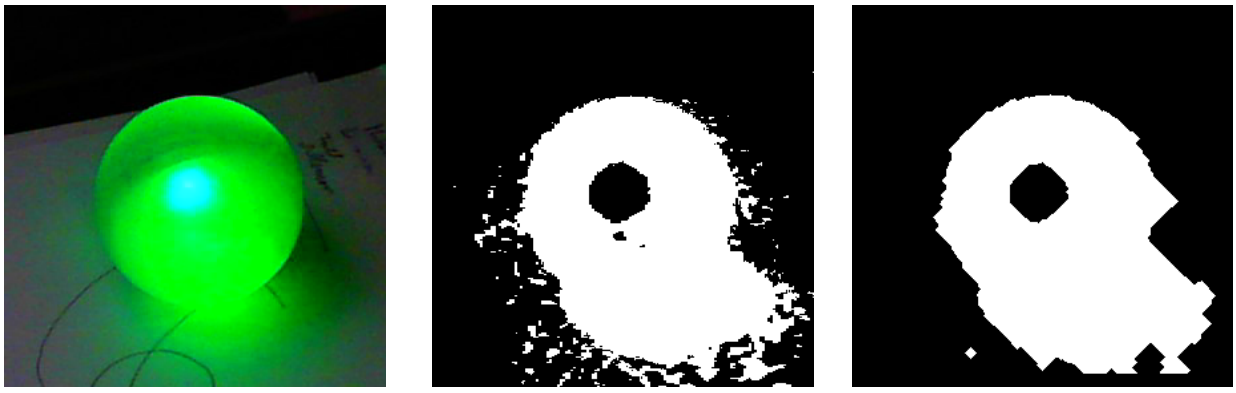
  Experiments revealed that reflections on the near environment, such as white paper and table tops, create numerous blobs that degrade the result. Furthermore, occlusions could separate areas of the true blob. In order to remove these small blobs and combine separated blobs, morphological operations are performed according to following equation:

$$\mathbf{I}_{\text{Clean}} = ((\mathbf{I} \ominus_{k_1} SE) \oplus_{k_2} SE) \ominus_{k_3} SE. \tag{4.14}$$

  $\ominus_k$ and $\oplus_k$ represent the k times iteratively performed erosion and dilation, respectively. A disk of size three was used as structure element (SE). Good performance was found with $k_1 = 3$, $k_2 = 14$, $k_3 = 10$.

  Since the dilated image is also used as a binary mask to perform contour extraction, the cleaned up blob has an increased size with $k_2 - (k_1 + k_3) = 1$.

  In figure 4.4 the result of color segmentation and the morphological clean up is illustrated. It demonstrates the successful segmentation and clean up of the blob, for a strongly perturbated image due to reflections.



(a) (b) (c)

Figure 4.4: Blob detection using color segmentation and morphological clean up:
(a) Original sub image, (b) Segmented image, (c) Morphological cleaned up image.

- **Blob Association**

  The blob with the biggest area will be assumed to be the correct one.

4.2.4. Contour Extraction

The contour extraction of the blob involves the following steps:

- **Cut out Blob Area**

  To reduce the computational burden and to trim away edges from the background, the area of the blob is cut out.
- **Convert to Gray Scale Image**

  The area of the blob represented as RGB image is converted to a gray scale image, by using the following transformation:

$$\text{Gray} = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B. \tag{4.15}$$

- **Edge Detection**

  Candidate points for the boundary are provided by the Canny edge detection. Canny edge detection will not be covered here; the interested reader is referred to [38].
  For the Gaussian filtering and Sobel gradient calculation a kernel size of three was taken. The thresholds for the hysteresis have to be chosen individually for each color, because in the color to gray transformation, the colors are weighted differently. A good ratio between the upper and lower threshold is given in between $\frac{T_{\text{upper}}}{T_{\text{lower}}} = \frac{1}{2} \ldots \frac{1}{3}$ [38]. For a green object the following thresholds showed good results: $T_{\text{upper}} = 200$, $T_{\text{lower}} = 100$.
- **Combine Edges and Blob-mask**

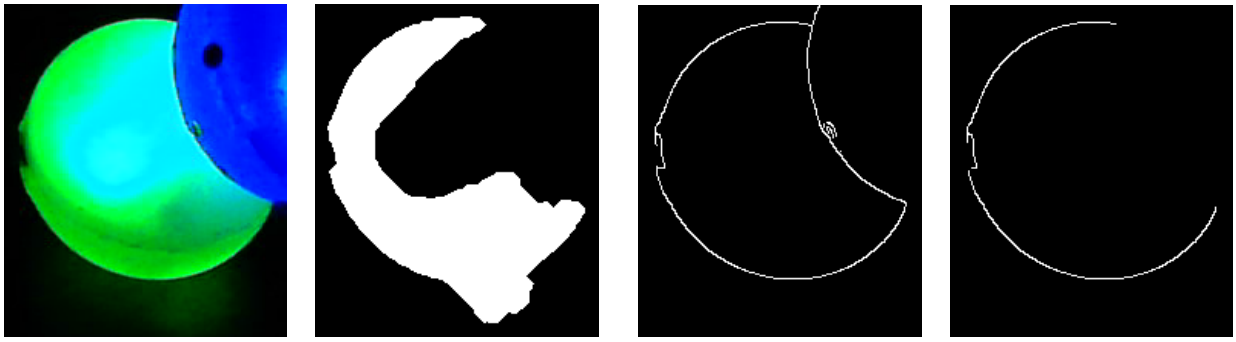
  In order to remove wrong colored edges, which are caused by background or foreground objects, the color segmentation mask is applied on the detected edges. As already mentioned in section 4.2.3 under *Image Cleanup*, the dilation operations have been performed once more, than the erosions operations. Resulting in an increased size of the mask, in order to contain the detected edges.

It should be noted that the less wrong edges are found, the faster and less error prone the following processing steps will be.

An exemplary result of contour extraction is presented in figure 4.5 for an image perturbated by reflections and in figure 4.6 for an image perturbated by a second sphere. Both show the successful contour extraction even under difficult conditions.

(a) (b) (c) (d)

Figure 4.5: Results of contour extraction for an image affected by reflections:
(a) Sub cut area of the original image, (b) Color segmented image, (c) Detected edges. (d) Final contour: Combination of color segmented and detected edges.



(a) (b) (c) (d)

Figure 4.6: Results of contour extraction for an image including a second sphere:
(a) Sub cut area of the original image, (b) Color segmented image, (c) Detected edges. (d) Final contour: Combination of color segmentation and detected edges.

4.2.5. Image Correction

The image correction consists of the following two steps. The first corrects lens flaws and the second transforms the projected ellipse to a circle to compensate projective distortions.

4.2.5.1. Compensation of Lens Flaws

The projection of a point $[X, Y, Z]^T$ from the 3D space onto a 2D plane can be described by using the pinhole camera model according to [27]:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{bmatrix} \tag{4.16}$$

$$= \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix}. \tag{4.17}$$

f_x, f_y, c_x and c_y are focal lengths and principal points in the x and y-direction, respectively and are expressed in pixels. In figure 4.7 a graphical demonstration of the image formation is illustrated.

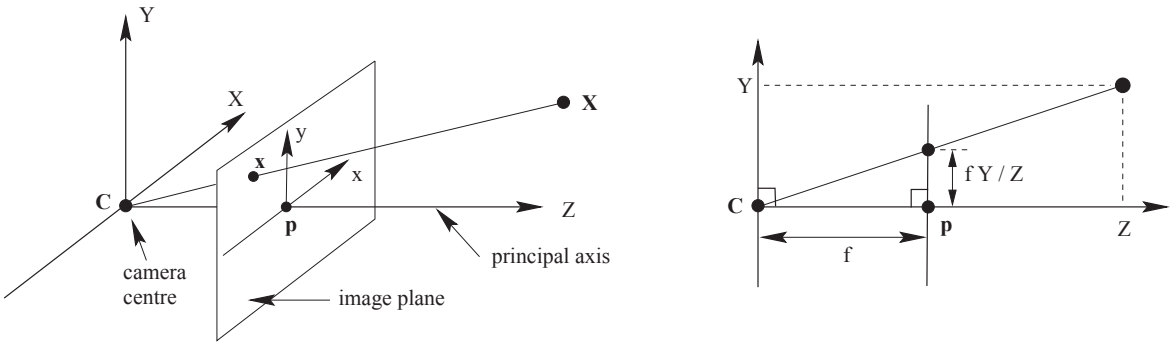


Figure 4.7: Pinhole camera model with C as the camera center and p as the principal point. The image plane is placed in front of the camera [27].

The equation above solely describes the projection of a perfect pinhole camera. However, realistic cameras contain tangential and radial distortions, which are caused by non-collinear mounting of multiple lenses and by spherical aberration, respectively.

For the compensation, the following nonlinear mapping according to [39] will be used:

$$X'_{\text{corr}} = X' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 X' Y' + p_2 (r^2 + 2X'^2), \tag{4.18}$$

$$Y'_{\text{corr}} = Y' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1 (r^2 + 2Y'^2) + 2p_2 X' Y', \tag{4.19}$$

$$r^2 = X'^2 + Y'^2, \tag{4.20}$$

with X'_{corr} and Y'_{corr} as the corrected points. The coefficients for the radial distortion are given by k_1, k_2, k_3, k_4, k_5 and k_6, whereas the coefficients for the tangential distortion are given as p_1 and p_2. Usually the correction is performed for the whole image using a precomputed Look up Table LUT. However, using a LUT results in discretization of the corrected coordinates. Hence the number of contour points is very small, each point is corrected by solving equation (4.18), equation (4.20) and equation (4.20).
The open source library OpenCV [39] has been used to estimate the camera matrix, to estimate the distortion coefficient, as well as to compensate lens flaws.

4.2.5.2. Compensation of Projective Distortion

The projection of a sphere onto the image plane can be seen as a cone-plane intersection. In general this intersection creates an ellipse for a right cone. However the intersection will describe a circle, if the cutting plane is perpendicular to the symmetry axis of the cone. This is demonstrated in figure 4.8.

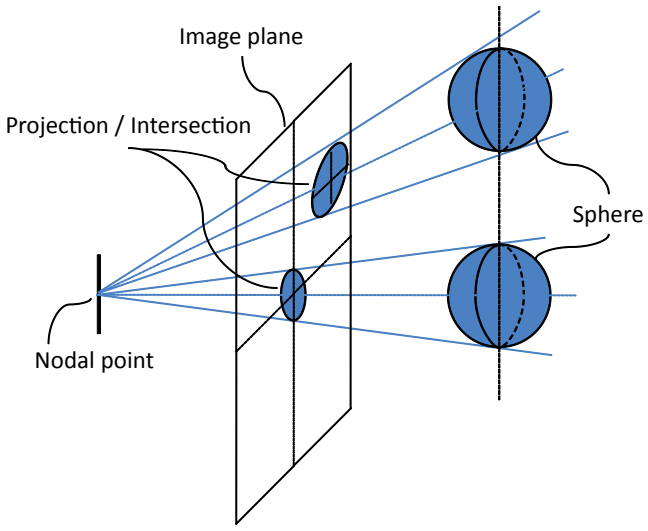


Figure 4.8: Image projection modeled as cone plane intersection.

The idea is to reproject the contour points on a new plane, which is perpendicular to the symmetry axis of the cone. This reprojected outline of the sphere describes a circle, whose parameters can be obtained using a circle fit method.
The desired plane is found as the intersection of the symmetry axis of the cone itself with the tangential plane of the sphere, which is described by the focal length. This is demonstrated in figure 4.9.

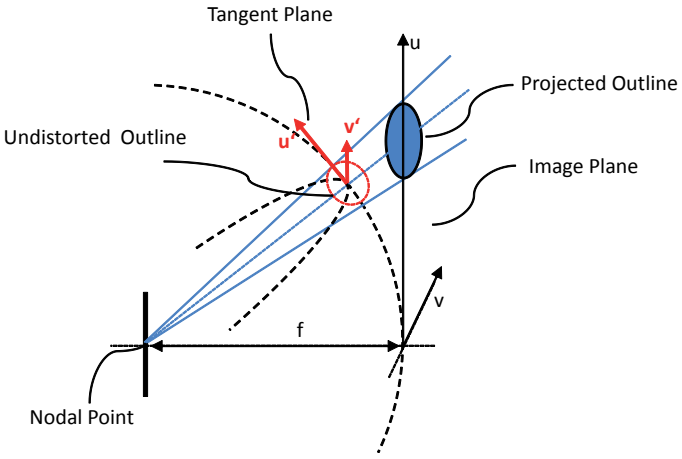


Figure 4.9: Projective undistortion as reprojection on a sphere's tangent plane.

The computation of the compensation of the projective distortion contains the following steps:

- **Normalize image coordinates**

  In order to get the contour points expressed in units of the 3D coordinate system, the contour points are normalized as:

$$\mathbf{p}_{\text{norm}} = \begin{bmatrix} u_{\text{norm}} \\ v_{\text{norm}} \\ 1 \end{bmatrix} = \mathbf{M}_{\text{Cam}}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{4.21}$$

- **Find Symmetry Axis**

  The four points describing the bounding box of the contour are projected on the unit sphere as:

$$\mathbf{p}_{\text{Unit}} = \frac{\mathbf{p}_{\text{norm}}}{\|\mathbf{p}_{\text{norm}}\|}. \tag{4.22}$$

  An estimation of the symmetry axis $\mathbf{r}_{\text{axis}}$ is given by taking the unit vector of the vectorial summation of these four projected points.

$$\mathbf{r}_{\text{axis}} = \frac{1}{4} \sum_{i=1}^{4} \mathbf{p}_{\text{Unit,i}}. \tag{4.23}$$

- **Calculate Coordinate System**

  With the given symmetry axis a new coordinate system on the tangential plane of the unit sphere can be introduced using spherical coordinates. The relation between the two coordinated systems is shown in figure 4.10.

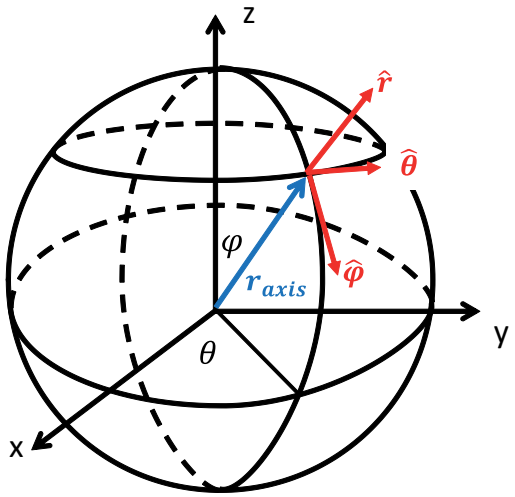


Figure 4.10: Spherical coordinate system for projective undistortion.

The unit vectors of the spherical coordinate system are given as [40]:

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \\ 0 \end{bmatrix}, \tag{4.24}$$

$$\hat{\boldsymbol{\phi}} = \begin{bmatrix} \cos(\theta) \cdot \cos(\phi) \\ \sin(\theta) \cdot \cos(\phi) \\ -\sin(\phi) \end{bmatrix}, \tag{4.25}$$

$$\hat{\mathbf{r}} = \mathbf{r}_{\text{axis}} \tag{4.26}$$

with

$$\theta = \tan^{-1}\left(\frac{\mathbf{r}_{\text{axis,x}}}{\mathbf{r}_{\text{axis,y}}}\right), \tag{4.27}$$

$$\phi = \cos^{-1}\left(\frac{\mathbf{r}_{\text{axis,z}}}{\sqrt{\mathbf{r}_{\text{axis,x}}^2 + \mathbf{r}_{\text{axis,y}}^2 + \mathbf{r}_{\text{axis,z}}^2}}\right). \tag{4.28}$$

- **Reprojection**

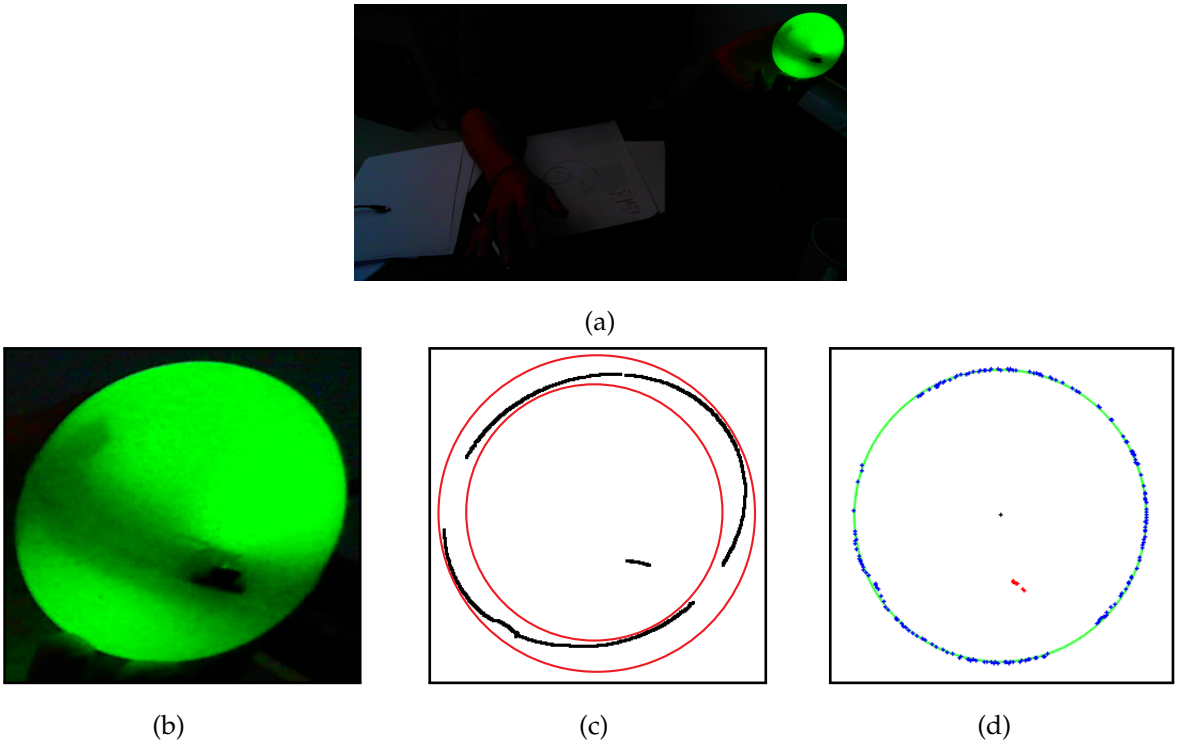
  A projected point $[u', v']$ is given as the intersection between the line through the original point $\mathbf{p}_{norm}$ and the tangent plane and can be written as follows:

$$\mathbf{p}_{\text{norm}} \cdot \frac{1}{\lambda} = \hat{\mathbf{r}} + \hat{\boldsymbol{\theta}} \cdot u' + \hat{\boldsymbol{\phi}} \cdot v', \tag{4.29}$$

$$\mathbf{p}_{\text{norm}} = \begin{bmatrix} \hat{\boldsymbol{\theta}} & \hat{\boldsymbol{\phi}} & \hat{\mathbf{r}} \end{bmatrix} \begin{bmatrix} \lambda u' \\ \lambda v' \\ \lambda \end{bmatrix}. \tag{4.30}$$

  The projected points are obtained by solving section 4.2.5.2, which represents an efficient calculation, since the matrix inverse has to be calculated only once for all points.

In figure 4.11 the image of a sphere, which is located close to the border, is presented and shows strong projective distortion. To highlight the effect of distortion, the contour is overlaid with its inner and outer circle. Furthermore, the undistorted contour including a successful circle fit is shown.



(a)

(b) (c) (d)

Figure 4.11: Demonstration of projective distortion and its correction:
(a) Original image, sphere located at the edge; (b) Sub-cut image; (c) Extracted contour (black) including inner and outer circle (red); (d) Undistorted contour (blue) and successful circle fit (green).

4.2.6. Circle Fit

For the obtained contour a simple circle fit cannot be applied. The contour not only contains elements of the real circle. It also contains outliers, which would strongly degrade the result if neglected. Such outliers are caused by:

- Reflections
- Occlusion due to fingers
- Wrong detected edges from the foreground or background

In order to deal with outliers, a robust circle fit method has to be used. Two popular methods are given with the Hough Transform (HT) and the **Ran**dom **Sa**mple **C**onsensus (RANSAC) algorithm. The HT allows detection of multiple object, however the computational complexity and storage is higher compared to RANSAC. Furthermore, RANSAC provides higher precession [41].
Since only one circle has to be fitted and due to the above mentioned advantages of the RANSAC compared to HT, the RANSAC method will be used.
The circle fit consists of two parts. The first identifies the correct contour points and the second refines the circle fit using best fit method.

4.2.6.1. RANSAC for Circular Objects

The random sample consensus approach was introduced by Fischler [42] and contains the following steps [27]:

1. **Random Sampling**
   Randomly sample a minimal set of data points from the contour points. For fitting a circle, a minimum set of three points is required.
2. **Model fit**
   The model fit is based on the fact, that perpendiculars of the bisection between two points on the circle pass through the center of the circle (see figure 4.12). The parameters of the circle can be calculated given the samples set of three points $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ according to [43]:

$$\mathbf{c}_{u'} = (D \cdot E - B \cdot F)/G, \tag{4.31}$$

$$\mathbf{c}_{v'} = (A \cdot F - C \cdot E)/G, \tag{4.32}$$

$$r = \sqrt{(\mathbf{p}_{1,u'} - \mathbf{c}_{u'})^2 + (\mathbf{p}_{1,v'} - \mathbf{c}_{v'})^2} \tag{4.33}$$

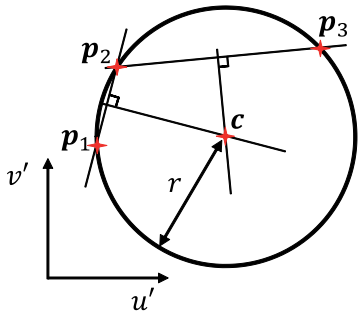


Figure 4.12: Circle fit using 3 points.

with:

$$A = \mathbf{p}_{2,u'} - \mathbf{p}_{1,u'}, \tag{4.34}$$

$$B = \mathbf{p}_{2,v'} - \mathbf{p}_{1,v'}, \tag{4.35}$$

$$C = \mathbf{p}_{3,u'} - \mathbf{p}_{1,u'}, \tag{4.36}$$

$$D = \mathbf{p}_{3,v'} - \mathbf{p}_{1,v'}, \tag{4.37}$$

$$E = A \cdot (\mathbf{p}_{1,u'} + \mathbf{p}_{2,u'}) + B \cdot (\mathbf{p}_{1,v'} + \mathbf{p}_{2,v'}), \tag{4.38}$$

$$F = C \cdot (\mathbf{p}_{1,u'} + \mathbf{p}_{3,u'}) + D \cdot (\mathbf{p}_{1,v'} + \mathbf{p}_{3,v'}), \tag{4.39}$$

$$G = 2 \cdot (A \cdot (\mathbf{p}_{3,v'} - \mathbf{p}_{2,v'}) - B \cdot (\mathbf{p}_{3,u'} - \mathbf{p}_{2,u'})). \tag{4.40}$$

The indices u' and v' stand for the projected image coordinates.

3. **Determine Inlier / Outlier**

   All points are tested if they are within a distance threshold ϵ of the model, given the calculated parameters $(\mathbf{c}_{u'}, \mathbf{c}_{v'}, r)$. A point $\mathbf{p}$ is counted as an inlier, if the following criterion is met:

   $$\epsilon^2 \geq |r^2 - ((\mathbf{c}_{u'} - \mathbf{p}_{u'})^2 + (\mathbf{c}_{v'} - \mathbf{p}_{v'})^2)|. \tag{4.41}$$

   If the number of inliers is bigger than the previous set S_i, the new points are taken as new set of inliers S_i. For this thesis a threshold of $\epsilon = 3$ pixels was chosen.

4. **Adjust Number of Iterations**

   Assuming only one outlier free sample is necessary to separate inliers from outliers, a number of N_p samples can be given to guarantee with a chosen probability p that a correct fit is obtained.

   The probability, that N_p samples are contaminated with outliers, is given as: $(1 - w^s)^{N_p}$.

With w as the probability of drawing an inlier and s as the sample size. Therefore the probability p of drawing at least once a sample, which is outlier free, is given as:

$$p = 1 - (1 - w^s)^{N_p}. \tag{4.42}$$

The number of samples N_p can then be calculated as:

$$N_p = \frac{log(1-p)}{log(1-w^s)}. \tag{4.43}$$

If a new set of inliers is found, the number of iterations N_p is updated as defined in the equation above with w as the actual ratio of inliers. The ratio of inliers is initialized with $w = 0.5$ and a probability of $p = 99.9999\,\%$ was chosen.

5. **Repeat**

   The steps 1 - 4 are repeated until the following criteria are met:

   - The actual number of iteration N exceeds the minimum number of required iterations N_p to obtain an outlier free sample: $N > N_p$.
   - The iteration limit is reached $N > N_{\max}$.

6. **Check Health of Fit**

   In order to reject bad sets of inliers, the health of the fit is evaluated. The following criteria have to be fulfilled, in order to consider a correct fit:

   - The iteration limit is not reached $N < N_{\max}$.
   - A minimum number of inliers are found.

     Note: Here a minimum size of eight points was used.
   - A minimum ratio of inliers is reached $w > w_{\min}$.

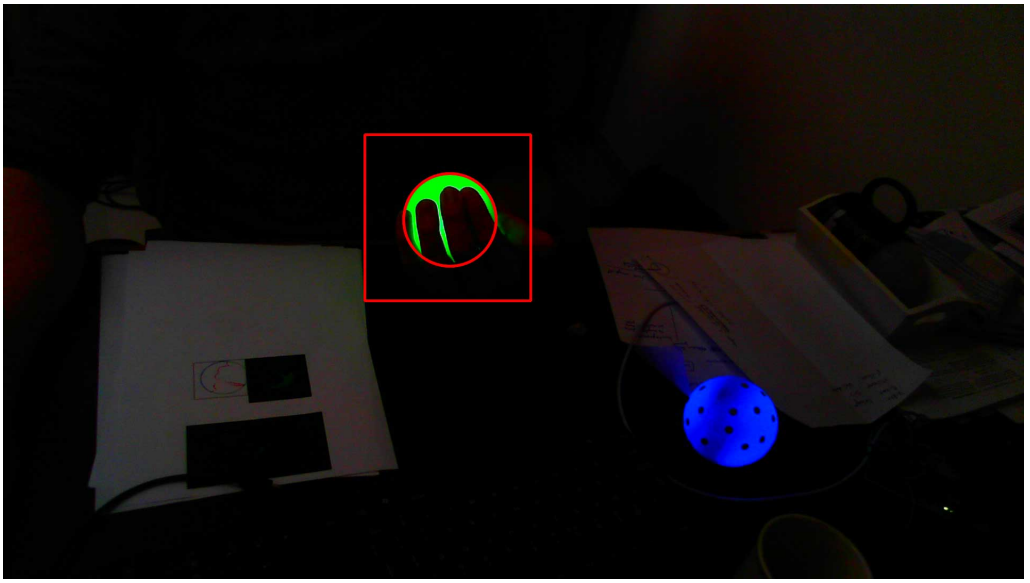
     Note: Here a minimum ratio of $w_{\min} = 25\,\%$ was used.

7. **Circle Refinement**

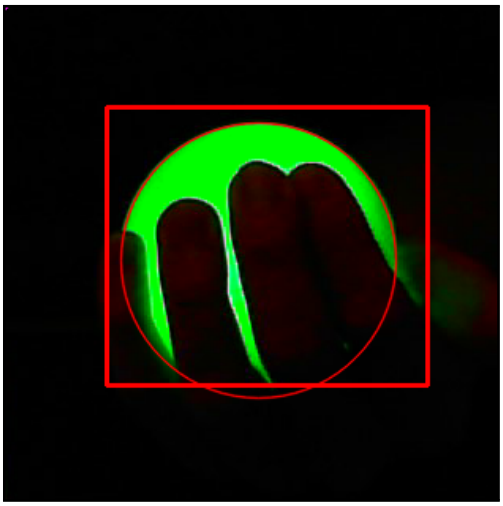
   The circle fit method explained in section 3.1.3.1 and defined by equation (3.47), was used to refine the fit.

It should be noted that the contour points have been subsampled, in order to reduce the number of calculations in step 3.

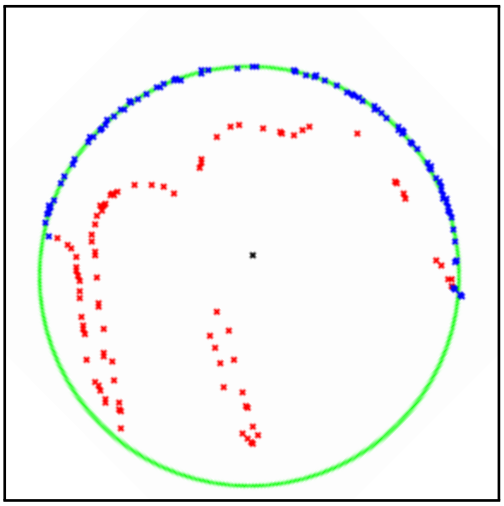
In figure 4.13 the result of the RANSAC circle fit is shown. The figure clearly demonstrates a successful fit even under strong occlusions such as fingers.



(a)



(b)



(c)

Figure 4.13: Demonstration of RANSAC circle fit:
(a) Original image of a strongly occluded image showing circle fit (red); (b) Sub-cut image showing circle fit (red); (c) RANSAC result: Inlier (blue), Outlier (red), fitted circle (green).

4.2.7. Fault Rejection

Before the position of the sphere is calculated, the fitted circle is again checked for its correctness. This check mostly removes pseudo circles, which could be created due to partly occlusion, as shown in figure 4.14.

In order to reject those wrong pseudo circles, only circles satisfying the following criteria are

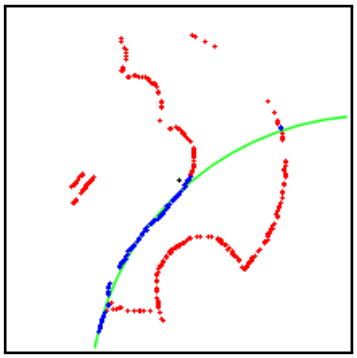


Figure 4.14: Wrong circle detection of RANSAC due to pseudo circles.
Inlier (blue), Outlier (red), fitted circle (green).

accepted:

$$\frac{FG_{\text{Circle}}}{A_{\text{Circle}}} > 0.3, \tag{4.44}$$

$$\frac{FG_{\text{Circle}}}{FG_{\text{Total}}} > 0.3, \tag{4.45}$$

with FG_{Circle} as the amount foreground pixels within the fitted circle, FG_{Total} as the total amount of foreground pixels and A_{Circle} as the area of the fitted circle, expressed in pixels.

4.2.8. Position Calculation

Using the parameters of the circle fit $\mathbf{c}_{u'}, \mathbf{c}_{v'}, r$, the sphere's position can be calculated using trigonometry according to figure 4.15.

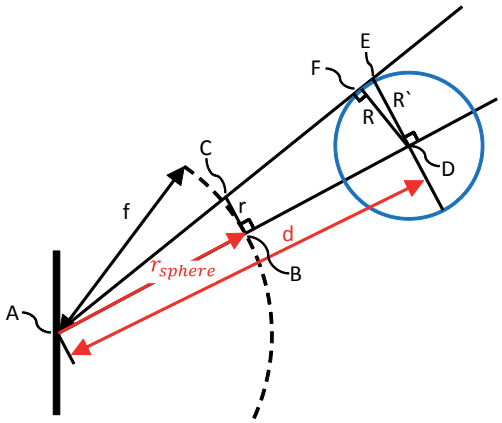


Figure 4.15: Position calculation using similar triangles.

The position calculation contains the following three steps:

- **Calculate Distance**

  The distance calculation is based on the triangles in figure 4.15. Using the similarity between the triangle ABC and ADE we get:

  $$\frac{f}{r} = \frac{R'}{d}, \tag{4.46}$$

  and with the triangle ABC and DFE:

  $$\frac{r}{f} = \frac{\sqrt{R'^2 - R^2}}{R}. \tag{4.47}$$

  Solving for the distance d will result in:

  $$d = R\sqrt{1 + \frac{f^2}{r^2}}, \tag{4.48}$$

  with R as the known radius of the sphere and the normalized focal length $f = 1\,\text{m}$.

- **Calculate Direction**

  The direction of the sphere $\mathbf{n}_{\text{sphere}}$ is given as the unit vector of the estimated circle center and is expressed in the camera coordinate system. The direction can be calculated as follows:

  $$\mathbf{r}_{\text{sphere}} = \mathbf{r}_{\text{axis}} + \mathbf{c}_{u'} \cdot \hat{\boldsymbol{\phi}} + \mathbf{c}_{v'} \cdot \hat{\boldsymbol{\theta}}, \tag{4.49}$$

  $$\mathbf{n}_{\text{sphere}} = \frac{\mathbf{r}_{\text{sphere}}}{\|\mathbf{r}_{\text{sphere}}\|}. \tag{4.50}$$

- **Calculate Position**

  If the direction and the distance of the sphere is known, the position can be calculated as follows:

  $$\mathbf{P}_{\text{sphere}} = \mathbf{n}_{\text{sphere}} \cdot d. \tag{4.51}$$

5. Sensor Fusion for Tracking

This chapter starts with the introduction to sensor fusion by providing an illustrative example of sensor fusion. Further the advantages of sensor fusion are pointed out. Subsequently a brief overview of possible fusion methods is given. After explaining the selected fusion strategy in detail, the complete system model for the optical-inertial sensor fusion based on chapter 2/3/4 will be explained. Finally several adaptations to obtain a reasonable functioning fusion framework are presented.

5.1. Introduction

The aim of tracking is to provide position and orientation information of a moving object over time. Also in several applications time derivatives, such as velocity and angular rotation, are of interest.

Using state estimation techniques, the position and the orientation can be determined given indirect, inaccurate or uncertain observations. The following example should give insight in the functional principle of the state estimation process with respect to tracking:

In order to provide a descriptive example it will be assumed the person's brain in this example acts in a similar way as a state estimator. Imagine, a person remotely controls a toy plane using throttle and steering angle. In order to properly maneuver the plane, the person has to continuously determine the plane's position, velocity and orientation. These quantities of interest will be called states.

By using the knowledge of the inputs of the remote control and the actual position, the person predicts the future states of the plane. Known inputs are steering angle and throttle. The model describes the movement of the plane and how it behaves according to given input.

In general, the plane does not completely behave the way the person predicted it. Reasons for the discrepancy can either be an inappropriate model of the plane's behavior, or unknown perturbations (e.g. wind dragging the plane away).

Furthermore the person's eyes can be seen as a measurement system, which provides position measurements. Unfortunately bad weather conditions and fog impairs his sight and the person can only roughly guess the plane's position.

By combining the information of the predicted position and the observed position, an improved estimation of the airplane's states can be given. Moreover consider the plane flies behind a thick/dense cloud and the person sight of the plane is further degraded. In that case the person increases the trust for predicted position, since hardly information by the person's eyes is available. The information from the prediction and the measurements is weighted based on the trust for each.

In a similar way, as the person describes the behavior of the plane's movement, the evolution of states in time can be modeled as a dynamic system according to stochastic equations [1]. The following discrete-time nonlinear dynamic state-space model (DSSM) will be used to model the system and measurement behavior:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{n}_k) \tag{5.1}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) \tag{5.2}$$

The nonlinear state transition function $\mathbf{f}(\cdot)$ represents the dynamic of the discrete time state vector of the plane (or of any other tracked object). In some cases direct input to the model is known (e.g. throttle and steering angle). This known input is called exogenous input $\mathbf{u}_k$.
Furthermore, hardly all inputs for such a model can be known or the model does not totally reflect the real world behavior. To consider these effects, another input called process noise $\mathbf{n}_k$, is assumed. In general only statistical assumptions of the process noise can be made.
The states $\mathbf{x}_k$ of the system represent position, orientation or any other quantities of interest.
In order to incorporate measurements, for example as provided by the person's eyes, the nonlinear measurement function $\mathbf{h}(\cdot)$ is introduced. It describes the relation between states and measurements $\mathbf{y}_k$. To consider unknown errors or inaccurate measurement models, the measurement noise $\mathbf{v}_k$ is introduced. Again only statistical assumptions of the measurement noise can be made.

A state estimator performs similar steps as the person in this example. By using the actual states $\hat{\mathbf{x}}_{k-1}$, known exogenous input $\mathbf{u}_k$ and assumptions of the unknown inputs $\mathbf{n}_k$, future states $\hat{\mathbf{x}}_{k|k-1}$ will be predicted based on the model of the system $\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{n}_k)$.
In the next step measurements $\hat{\mathbf{y}}_{k|k-1}$ are predicted using the predicted states $\hat{\mathbf{x}}_{k|k-1}$, assumptions of the measurement noise $\mathbf{v}_k$ and a model for the measurement device itself $\mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)$. By combining the information from the predicted measurements $\hat{\mathbf{y}}_{k|k-1}$ and the real measurements

$\mathbf{y}_k$, an optimal estimate $\hat{\mathbf{x}}_k$ of the real state can be given.
A typical structure of state estimation is demonstrated in figure 5.1.

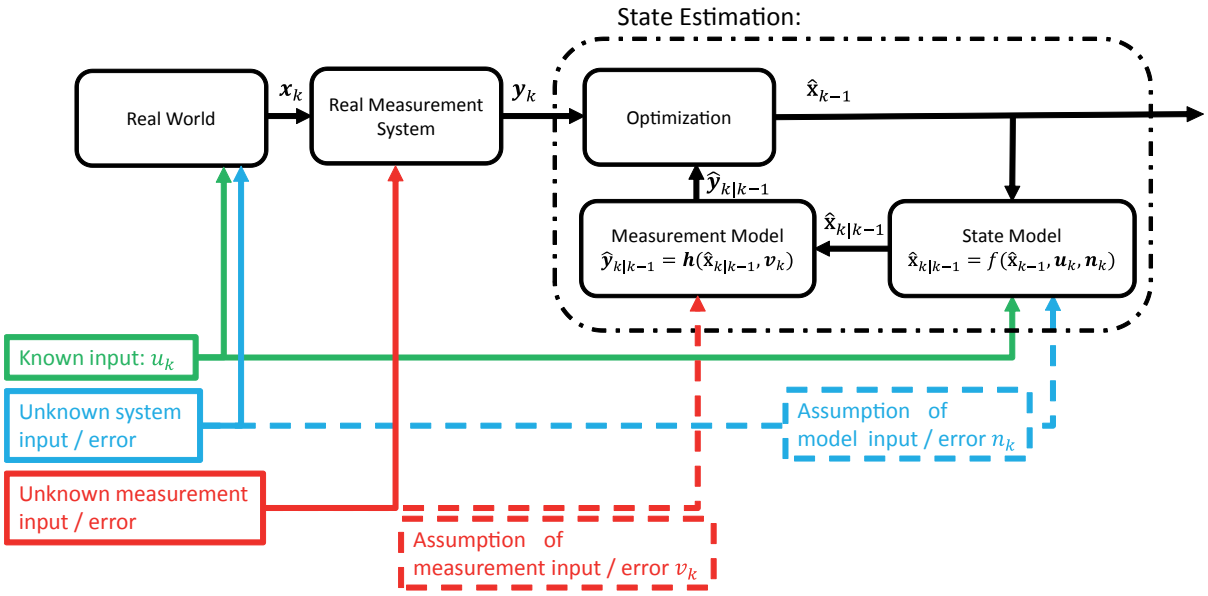


Figure 5.1: Typical approach of state estimation.

Furthermore, the estimation can be improved when information of multiple sensors is fused. This is demonstrated with the following example:
If a gust of wind drags the plane away and this wind is not noticed by the person, the estimation degrades and could be wrong. However, if the person would have felt the strength and direction of the wind on his skin, it could have considered its influence and therefore given a better estimate.

Sensor fusion can be defined as combining data, such that the resulting information is better than it would be possible, if these sources were used individually [44].
The improvements of sensor fusion can be categorized as follows [45]:

- **Robustness / Reliability**

  A system, which relies only on one sensor, is fragile. If the information is provided by an erroneous sensor, the measurement is just as erroneous. The system reacts based on the error affected information, fails and therefore it is not robust. On the other hand a system, which fuses redundant information of multiple sensors, will yield higher fault tolerance and reliability.

- **Accuracy / Certainty**

  The combination of different sensors can provide a more accurate estimation and resolve ambiguities. Consider the following example. We want to measure the frequency of a signal using two different sensors: The first has a very low accuracy, but is unambiguous over the whole measuring range. A second sensor achieves a very high accuracy, but due to aliasing it contains ambiguities in the sense of periodic output. By combining the information of both, the ambiguity of the second sensor can be solved and an accurate result can be achieved. Additionally a third sensor could improve the certainty, by confirming the actual measurement result.

- **Extended Coverage**

  By combining sensors with different physical quantities or sampling rates, the spatial and temporal coverage can be extended. Furthermore, the combination of complementary sensors allows the calculation of information, which could not be calculated if the sensors would be used individually. An example would be combining angle and distance information to determine the position [46].

5.2. Methods for Sensor Fusion

Methods for sensor fusion can be categorized as follows [47]:

- Gaussian approximation methods (KF, EKF, UKF)
- Sampling based methods (Particle filter)
- Artificial Intelligent methods (ANFIS)

The first two methods are Bayesian inference methods and are based on the calculation of the posterior probability distribution, which is defined as follows [48]:

$$p(\mathbf{x}|\mathbf{y}_1,\ldots,\mathbf{y}_n) = \frac{p(\mathbf{y}_1,\ldots,\mathbf{y}_n|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y}_1,\ldots,\mathbf{y}_n)}. \tag{5.3}$$

The posterior probability distribution is the distribution over the states $\mathbf{x}$, which is conditioned on the observed measurements. It represents the distribution when the information of all measurements, as well as the information of the prior distribution $p(\mathbf{x})$ over the states $\mathbf{x}$ is used. The prior distribution over the states $p(\mathbf{x})$ represents the information of the states before using information of any measurement. The term $p(\mathbf{y}_1,\ldots,\mathbf{y}_n|\mathbf{x})$ is called the likelihood function and is the joint probability of the measurements $\mathbf{y}_1,\ldots,\mathbf{y}_n$ given the states $\mathbf{x}$. The last term $p(\mathbf{y}_1,\ldots,\mathbf{y}_n)$ describes only a normalization term and is often neglected if the measurements are conditional independent [48].
Bayesian inference methods provide optimal solution to any model including nonlinear and non-Gaussian stochastic systems [48]. Practically only finite approximations of a Bayesian estimator can be implemented and a tradeoff between computational cost and accuracy has to be made. The following section gives a brief overview of common methods for sensor fusion with respect to tracking.

5.2.1. Kalman Filter

The Kalman Filter (KF) [49] is a stochastic sequential state-estimator, which falls under the category of Gaussian approximation methods. As the name already implies, Gaussian approximation methods approximate Random Variables (RV) as Gaussian (first and second order moments). Furthermore, a linear transition function $\mathbf{f}(\cdot)$, as well as a linear measurement function $\mathbf{h}(\cdot)$ are assumed. The process noise $\mathbf{n}_k$ and the measurement noise $\mathbf{v}_k$ are modeled as zero mean white noise with a multivariate normal distribution.
When using these assumptions, a closed form solution for propagating a RV can be given [31].

The iterative estimation consists of two steps. The prediction-step propagates the available estimation of the states and the according covariance through the model, in order to obtain *a priori* estimates.
In the correction-step the information from new measurements is used to correct the *a priori* estimates, which yields the *a posteriori* estimation of the states. The KF produces an optimal solution with respect to (MMSE), the maximum likelihood (ML) sense and the maximum a posteriori (MAP) sense [50]. However, in practical applications linear DSSM are rare and the real distribution of the RV is only approximated as Gaussian.

5.2.2. Extended Kalman Filter

For nonlinear estimation problems, the Extended Kalman filter (EKF) has become the most used nonlinear state-estimator in the last 30 years [31]. It is an adaption of the standard KF, in order to process nonlinear DSSM. The nonlinear transition function $\mathbf{f}(\cdot)$, as well as the nonlinear measurement function $\mathbf{h}(\cdot)$, are approximated by a first-order truncated Taylor series around the current estimated state [31]. The RVS are then propagated through that linearization DSSM . The EKF is computational efficient and can be applied to nonlinear DSSM. The EKF works fine for most applications [48].
Unfortunately, first-order Taylor series have to be calculated. Results using numerical derivatives are usually noisy and can become inaccurate. On the other hand the derivation of analytic derivatives can be difficult for certain DSSM. In addition the linearization introduces errors, which could lead to inconsistent estimations or even to divergence of the filter [31].

5.2.3. Unscented Kalman Filter

The Unscented Kalman Filter (UKF) was introduced by Julier and Uhlmann [51] to overcome the linearization problems of the EKF. It is based on the scaled unscented transformation (SUT) [52], in order to propagate the RV through the DSSM. It is based on the concept that it is easier to approximate a probability distribution function rather than to approximate the nonlinear transition or measurement function. Carefully chosen samples, so called sigma points, are selected around the current estimate. These sigma points are then propagated through the nonlinear state transition function $\mathbf{f}(\cdot)$ and the nonlinear measurement function $\mathbf{h}(\cdot)$. With the

propagated sigma points the first two statistical moments of the *a priori* states and measurements can be calculated. This step substitutes the KF prediction-step [31], [53].
An advantage of the UKF compared to the EKF is that no derivatives of nonlinear functions have to be calculated. Therefore it can be applied for non-smooth systems. Additionally, the UKF outperforms the EKF in terms of accuracy and consistency [31]. The computational cost of the UKF is slightly higher than the EKF using analytic derivatives. However, the computational cost are comparable to the EKF using numerical derivatives [31].

5.2.4. Particle Filter

The use of Particle filters is often assumed to be a brute force approach [54]. They often work well in situations where KF/ EKF / UKF show difficulties. This is the case for highly nonlinear systems and for RV with multi modal probability functions.
The Particle filter is a numerical implementation of a Bayesian estimator and its approach can be summarized as follows [54]:

- Generate N particles by sampling from their PDF $p(\mathbf{x})$.
- Perform the prediction step, in order to receive the *a priori* particles, using the nonlinear transition function $\mathbf{f}(\cdot)$ and the PDF of the process-noise $p(\mathbf{n})$.
- Perform the correction step by calculating the conditional relative likelihood for the particles given the measurements.
- Perform the resampling step by generating the *a posteriori* particles on basis of the conditional relative likelihood.
- Continue with the prediction step.

Advantages of the Particle filter are that it can be applied for strong nonlinear DSSM and for non-Gaussian and multi-modal distributions [31]. Unfortunately there are also downsides to this filter. In order to receive good accuracy, a huge amount of particles has to be used. This results in a much higher computational cost, than with KF, EKF or UKF.

5.2.5. Adaptive Neuro Fuzzy Inference System

An Adaptive Neuro Fuzzy Inference System (ANFIS) is the combination of an Artificial Neural Network (ANN) with a Fuzzy Interference System (FIS). The system provides the ability to include expert-knowledge (If-Then-statements) by FIS, as well as the ability of learning and adaption of parameters by ANN [55].

This approach is very different from the above mentioned methods. No physical model of the dynamic system or the measurement system is needed. The input/output relation is obtained by training the network on datasets.

Advantages of ANFIS are that there is no need for any physical model of the system and no need to provide statistical information of the sensory input. However, also no information of statistics of the estimation is provided. ANFIS showed a superior behavior in long measurement outages compared to Bayesian estimators [56].

Disadvantages of ANIFS are that it is based on training using large amount of data. The trained network can be critically sensitive to the training data and could behave unexpected in untrained situations [47].

For further information the reader is referred to [55]–[57].

5.2.6. Selection of the Sensor Fusion Method

As already mentioned, the sensor fusion framework should be applied in the 3D interaction device. With respect to this application the following criteria have been identified and should be considered when choosing an appropriate method for the sensor fusion framework.

- The sensor fusion framework should be able to track position and orientation of a moving object.
- The IMU, which will be used for the 3D interaction device, provides sensory data from a gyroscope and an accelerometer with a sampling rates up to 1 kHz. The fusion framework should be able to process this amount of data in real-time.
- The sensor fusion framework should operate on a consumer grade PC.
- Adding data from additional sensors should be easy.

Since the navigation equation, which calculates position and orientation, represents a nonlinear transition function, the sensor fusion framework should be able to process nonlinear measurement functions. Hence a standard KF cannot be used for the framework.
Even though a Particle filter can provides superior results within the explained Bayesian estimators, running a Particle filter with an update rate up to 1 kHz in real-time cannot be guaranteed on a consumer grade PC. For that reason the Particle filter is neither an option.
Although ANFIS demonstrated good performances for inertial navigation in long outages, adding a new sensor to the framework would require retraining and therefore ANFIS will not be used for the framework.
Because the UKF outperforms the EKF [31], [54] and no derivatives have to be calculated, the UKF has been chosen for the sensor fusion framework.

5.3. Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is a modified version of the classical KF and the EKF to overcome the problems with the propagation of the uncertainty through nonlinear DSSM. Instead of approximating the nonlinear functions of the DSSM as in EKF, the distribution of the states is approximated by deterministically chosen samples. The mean and covariance of the states are represented by these samples, which are also called sigma points. By propagating the sigma points through a nonlinear system, the *a posteriori* mean and covariance are obtained, which are accurate up to 2^{nd} order nonlinearity. Errors only occur in 3^{rd} and higher orders [31].

5.3.1. Scaled Unscented Transformation

The Scaled Unscented Transformation (SUT) represents the core of the UKF and describes the calculation of the expected mean and covariance of a random variable that undergoes a nonlinear transformation. We consider the following RV $\mathbf{x}$ with dimension L and propagate it through a nonlinear function:

$$\mathbf{y} = \mathbf{g}(\mathbf{x}). \tag{5.4}$$

In a Gaussian sense the RV $\mathbf{x}$ is fully described by its mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P_x}$.
The steps of the SUT are explained according to [58]:

- **Calculation of the Sigma Points**

  $2L + 1$ sigma points are deterministically drawn according to the following sampling scheme:

$$\begin{aligned}
&\mathcal{X}_0 = \bar{\mathbf{x}} && && w_0^{(m)} = \frac{\lambda}{L+\lambda},\\
&\mathcal{X}_i = \bar{\mathbf{x}} + \left(\sqrt{(L+\lambda)\mathbf{P_x}}\right)_i && i = 1,\dots,L && w_0^{(c)} = \frac{\lambda}{L+\lambda} + (1-\alpha^2+\beta),\\
&\mathcal{X}_i = \bar{\mathbf{x}} - \left(\sqrt{(L+\lambda)\mathbf{P_x}}\right)_i && i = L+1,\dots,2L && w_i^{(m)} = w_i^{(c)} = \frac{1}{2(L+\lambda)} \qquad i = 1,\dots,2L.
\end{aligned} \tag{5.5}$$

  The sampling scheme satisfies, that the set of samples $\mathbf{S}_i = \{w_i, \mathcal{X}_i\}$ captures the true mean and covariance of the RV. Each sigma point $\mathcal{X}_i$ is associated with its according weight w_i. The term $\left(\sqrt{(L+\lambda)\mathbf{P_x}}\right)_i$ represents the i-th column of the matrix square root of the weighted and scaled covariance matrix $((L+\lambda)\mathbf{P_x})$:

$$\left(\sqrt{(L+\lambda)\mathbf{P_x}}\right)\left(\sqrt{(L+\lambda)\mathbf{P_x}}\right)^T = \left((L+\lambda)\mathbf{P_x}\right). \tag{5.6}$$

An efficient method calculating the matrix square root is given by the Cholesky factorization method. The expression λ is calculated as follows:

$$\lambda = \alpha^2(L + \kappa) - L. \tag{5.7}$$

α, β, κ are called scaling parameters. α influences the spread of the sigma points around its mean $\bar{\mathbf{x}}$ and should be set to a small positive value ($\alpha = 10^{-3}$), in order to avoid non-local effects in strong nonlinear functions. The secondary scaling parameter κ should be chosen $\kappa \geq 0$ to ensure a positive semidefinite covariance matrix. It is mostly set to $\kappa = 0$.
In order to incorporate prior knowledge of the distribution function of the RV, the scaling parameter β can be used to capture higher order moments of the distribution function. For a Gaussian distribution, $\beta = 2$ represents the optimal choice [31], [58]. In figure 5.2 and figure 5.3 a set of selected sigma points for a 2-dimensional Gaussian distribution is shown.

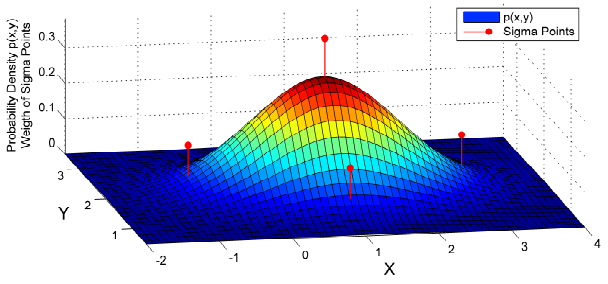


Figure 5.2: 2-Dimensional Gaussian RV and its according scaled sigma points. The height of sigma points represent their weight. Chosen parameter: $\alpha = 1, \beta = 1.5, \kappa = 1$. Adapted from [31].

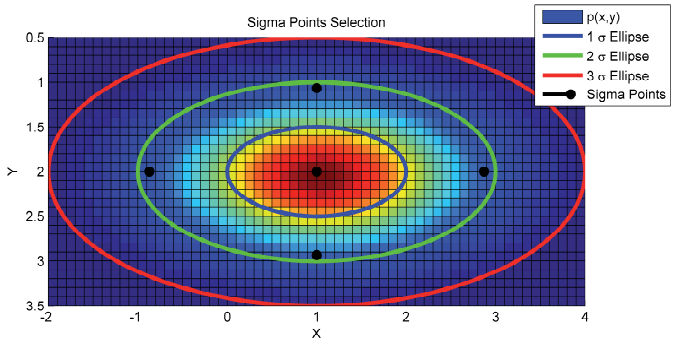


Figure 5.3: Top view: 2-Dimensional Gaussian RV and its according scaled sigma points.

- **Propagation of Sigma Points**

  Each column $\mathcal{X}_i$ of the sigma points is propagated through the true nonlinear function.

$$\mathcal{Y}_i = \mathbf{g}(\mathcal{X}_i) \qquad i = 0, \ldots, 2L. \tag{5.8}$$

- **Calculation of the posterior statistical moments**

  For the calculation of the first two statistical moments of the propagated RV, the weighted mean and covariance of the sigma points is calculated:

$$\bar{y} \approx \sum_{i=0}^{1L} w_i^{(m)} \mathcal{Y}_i, \tag{5.9}$$

$$P_y \approx \sum_{i=0}^{1L} w_i^{(c)} (\mathcal{Y}_i - \bar{y})(\mathcal{Y}_i - \bar{y})^T, \tag{5.10}$$

$$P_{xy} \approx \sum_{i=0}^{1L} w_i^{(c)} (\mathcal{X}_i - \bar{x})(\mathcal{Y}_i - \bar{y})^T. \tag{5.11}$$

5.3.2. Example of the Scaled Unscented Transformation

The following example shows the superiority of SUT in propagating a RV compared to propagating a RV using 1^{st} order linearization (used in EKF).

In figure 5.4 a tilted car in a 2-dimensional space is shown. The acceleration a_m^B in the body-frame $X_B Y_B$ and also the orientation θ of the car can be measured. We would like to calculate the linear acceleration a_{lin}^G in the global-frame $X_G Y_G$, which is given as:

$$\begin{aligned} \mathbf{a}_{\text{lin}}^{\mathbf{G}} = \begin{pmatrix} a_{\text{lin,x}}^G \\ a_{\text{lin,y}}^G \end{pmatrix} &= \mathbf{g}(\mathbf{x}) \\ &= \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \mathbf{a}_m^B + \mathbf{g}^G. \end{aligned} \tag{5.12}$$

The measurement $\mathbf{x} = \begin{pmatrix} a_m^B, x \\ a_m^B, y \\ \theta \end{pmatrix}$ is modeled as a Gaussian RV with mean $\bar{\mathbf{x}} = \begin{pmatrix} 5.1 \\ 8.5 \\ 30 \end{pmatrix}$ and covariance $\mathbf{P}_x = \begin{pmatrix} 10^{-4} & 0 & 0 \\ 0 & 10^{-4} & 0 \\ 0 & 0 & 25 \end{pmatrix}$.

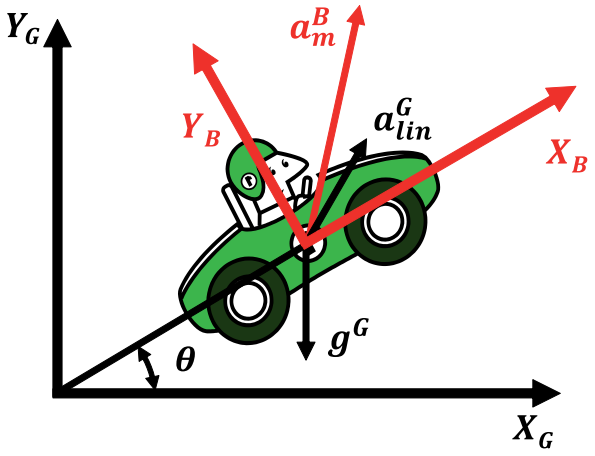


Figure 5.4: Example of the Scaled Unscented Transformation: Calculation of the linear acceleration $\mathbf{a}_{\text{lin}}^{\mathbf{G}}$ in the global-frame $X_G Y_G$; given the measured acceleration $\mathbf{a}_m^B$ in the body-frame $X_B Y_B$ and the angle θ.

Figure 5.5 shows the results for the true propagation of the RV $\mathbf{a}_{\text{lin}}^{\mathbf{G}}$ (blue), the propagation using SUT (red) and the propagation using 1^{st} order-linearization (green). The ellipses and bold crosses represent the covariance and mean, respectively.

The ground truth (small blue crosses) is carried out by propagating 10 000 samples through the nonlinear function. The samples are drawn from their distribution.
The propagated mean and covariance using 1^{st} order-linearization are calculated as:

$$\bar{\mathbf{a}}_{\text{lin}}^{\mathbf{G},1^{st}O.} = \mathbf{g}(\bar{\mathbf{a}}_{\mathbf{m}}^{\mathbf{B}}, \bar{\theta}), \tag{5.13}$$

$$\mathbf{P}_{a_{\text{lin}}^{G}}^{1^{st}O.} = (\nabla \mathbf{g}(x))\mathbf{P}_x(\nabla \mathbf{g}(x))^T. \tag{5.14}$$

The result for the 1^{st} order-linearization shows a biased and inconsistent result, whereas the result for SUT presents an accurate calculation of the true mean and covariance.

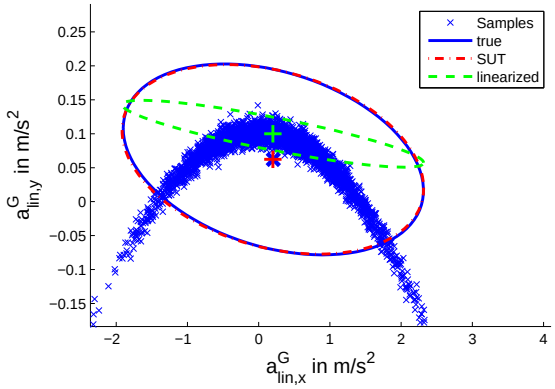


Figure 5.5: Comparison: Propagation of random variables and its first two statistical moments for $\mathbf{a}_{\text{lin}}^{\mathbf{G}}$; Ground Truth (blue); SUT (red); 1^{st} order-linearization (green). The ellipses and bold crosses represent the covariance and mean, respectively.

5.3.3. Implementation of the UKF

Van Der Merwe proposed two different algorithms for the UKF in [31]. One assumes simply additive noise, whereas the other also can handle non-additive noise. In order to ensure the full potential of the sensor fusion framework, the non-additive noise version will be implemented. The following explanation of the UKF is according to [31].
For the implementation the following nonlinear discrete-time dynamic state-space model (DSSM) will be used:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{n}_k), \tag{5.15}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k), \tag{5.16}$$

where:

$\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{n}_k)$	is the discrete-time transfer function.
$\mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)$	is the discrete-time measurement function.
$\mathbf{x}_k$	is the state vector at time step k.
$\mathbf{y}_k$	is the measurement vector at time step k.
$\mathbf{u}_k$	is the known exogenous input at time step k.
$\mathbf{n}_k$	is the process noise at time step k.
$\mathbf{v}_k$	is the measurement noise at time step k.

The process noise $\mathbf{n}_k$ and the measurement noise $\mathbf{v}_k$ are independent zero-mean Gaussian noises with covariance matrix $\mathbf{R}_n$ and $\mathbf{R}_v$, respectively. In order to incorporate non-additive noise, sigma points for these noises have to be deterministically sampled. For that reason the original state vector $\mathbf{x}_{k-1}$ is expanded with the process noise $\mathbf{v}_k$ and measurement noise $\mathbf{n}_k$. The augmented state vector $\mathbf{x}^a_{k-1}$ is given as:

$$\mathbf{x}^a_{k-1} = \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{v}_k \\ \mathbf{n}_k \end{bmatrix}. \tag{5.17}$$

In a similar way, the augmented covariance matrix $\mathbf{P_x}^a$ is merged using their according covariance matrices:

$$\mathbf{P_x}^a = \begin{bmatrix} \mathbf{P_x} & 0 & 0 \\ 0 & \mathbf{R_v} & 0 \\ 0 & 0 & \mathbf{R_n} \end{bmatrix}. \tag{5.18}$$

$\mathbf{P_x}, \mathbf{R_v}, \mathbf{R_n}$ represent the covariance matrix of the original state, process noise and measurement noise, respectively. The procedure of the UKF is illustrated by the flowchart in figure 5.6 and can be described by the following steps:

- **Initialization:**

  The UKF is initialized with the expected initial states and the covariance matrix for this estimation:

$$\begin{aligned} &\hat{\mathbf{x}}_0 = E[\mathbf{x}_0], && \mathbf{P}_{\mathbf{x}_0} = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T], \\ &\hat{\mathbf{x}}_0^a = E[\mathbf{x}_0^a] = E[\mathbf{x}_0 \quad \mathbf{0} \quad \mathbf{0}]^T, && \mathbf{P}_{\mathbf{x}_0}{}^a = E[(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)^T] = \begin{bmatrix} \mathbf{P}_{\mathbf{x}_0} & 0 & 0 \\ 0 & \mathbf{R_v} & 0 \\ 0 & 0 & \mathbf{R_n} \end{bmatrix}. \end{aligned} \tag{5.19}$$

- **For $k = 1, \ldots, \infty$:**

  1. **Calculation of the sigma points:**

     The augmented sigma points $\mathcal{X}_{k-1}^a$are calculated according to the SUT in section 5.3.1 with the augmented state vector $\mathbf{x}^a$:

$$\mathcal{X}_{k-1}^a = \begin{bmatrix} \hat{\mathbf{x}}_{k-1}^a & \hat{\mathbf{x}}_{k-1}^a + \left(\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{x}_{k-1}}^a}\right) & \hat{\mathbf{x}}_{k-1}^a - \left(\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{x}_{k-1}}^a}\right) \end{bmatrix}. \tag{5.20}$$

     Where $L = L_x + L_v + L_n$ is the dimension of the augmented state variable and L_x, L_v, L_n represents the dimension of the original state, process noise and measurement noise variable, respectively. The resulting sigma points can be seen as the concatenation of the individual sigma points: $\mathcal{X}_{k-1}^a = \begin{bmatrix} (\mathcal{X}_{k-1}^x)^T & (\mathcal{X}_k^v)^T & (\mathcal{X}_k^n)^T \end{bmatrix}^T$. The scaling parameter λ can be calculated by equation (5.7).

  2. **Prediction:**

     The sigma points $\mathcal{X}_{k-1}^x$ are predicted to the next time step using the discrete-time transition function $\mathbf{f}(\cdot)$, the known exogenous input $\mathbf{u}_k$ and the process noise $\mathbf{v}_k$. These predicted sigma points are also called *a priori* sigma points. The prediction is given as:

$$\mathcal{X}_{k|k-1}^x = \mathbf{f}(\mathcal{X}_{k-1}^x, \mathcal{X}_k^v, \mathbf{u}_k). \tag{5.21}$$

     The *a priori* state estimation $\hat{\mathbf{x}}_{k|k-1}$ and its covariance matrix $\mathbf{P}_{\mathbf{x}_{k|k-1}}$ are obtained by

calculating the weighted mean and covariance of the *a priori* sigma points $\mathcal{X}^x_{k|k-1}$ as:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{X}^x_{i,k|k-1}, \tag{5.22}$$

$$\mathbf{P}_{\mathbf{x}_{k|k-1}} = \sum_{i=0}^{2L} w_i^{(c)} (\mathcal{X}^x_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\mathcal{X}^x_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})^T. \tag{5.23}$$

The factors $w_i^{(m)}$ and $w_i^{(c)}$ represent the weights for the mean and covariance calculation, respectively and can be obtained by equation (5.5).

The *a priori* sigma points $\mathcal{X}^x_{k|k-1}$ are propagated through the measurement function $\mathbf{h}(\cdot)$ in order to obtain *a priori* measurements $\hat{\mathbf{y}}_{k|k-1}$ including their covariance matrix $\mathbf{P}_{\tilde{\mathbf{y}}_k}$ as:

$$\mathcal{Y}_{k|k-1} = \mathbf{h}(\mathcal{X}^x_{k|k-1}\mathcal{X}^n_k), \tag{5.24}$$

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{Y}_{i,k|k-1}, \tag{5.25}$$

$$\mathbf{P}_{\tilde{\mathbf{y}}_{k|k-1}} = \sum_{i=0}^{2L} w_i^{(c)} (\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T. \tag{5.26}$$

The cross-covariance matrix between *a priori* measurements and *a priori* states is calculated as:

$$\mathbf{P}_{\mathbf{x}_k,\mathbf{y}_k} = \sum_{i=0}^{2L} w_i^{(c)} (\mathcal{X}^x_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T. \tag{5.27}$$

3. **Correction:**

The Kalman gain $\mathbf{K}_k$, which minimizes the covariance of the estimation, is calculated as follows:

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k\mathbf{y}_k} \mathbf{P}^{-1}_{\tilde{\mathbf{y}}_{k|k-1}}. \tag{5.28}$$

The difference between the *a priori* measurement $\hat{\mathbf{y}}_{k|k-1}$ and the actual measurement $\mathbf{y}_k$, also called innovation, is weighted with the Kalman gain $\mathbf{K}_k$ to obtain the correction term $\Delta\mathbf{x}_k$:

$$\Delta\mathbf{x}_k = \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}). \tag{5.29}$$

The *a posteriori* estimation $\hat{\mathbf{x}}_k$ is calculated by correcting the *a priori* estimation $\hat{\mathbf{x}}_k$ by the correction term $\Delta\mathbf{x}_k$:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \Delta\mathbf{x}_k. \tag{5.30}$$

Finally the *a posteriori* covariance matrix $\mathbf{P}_{\mathbf{x}_k}$ is updated as:

$$\mathbf{P}_{\mathbf{x}_k} = \mathbf{P}_{\mathbf{x}_{k|k-1}} - \mathbf{K}_k \mathbf{P}_{\tilde{\mathbf{y}}_{k|k-1}} \mathbf{K}_k^T. \tag{5.31}$$

4. The estimation process starts again with step 1 *'Sigma points calculation'*.

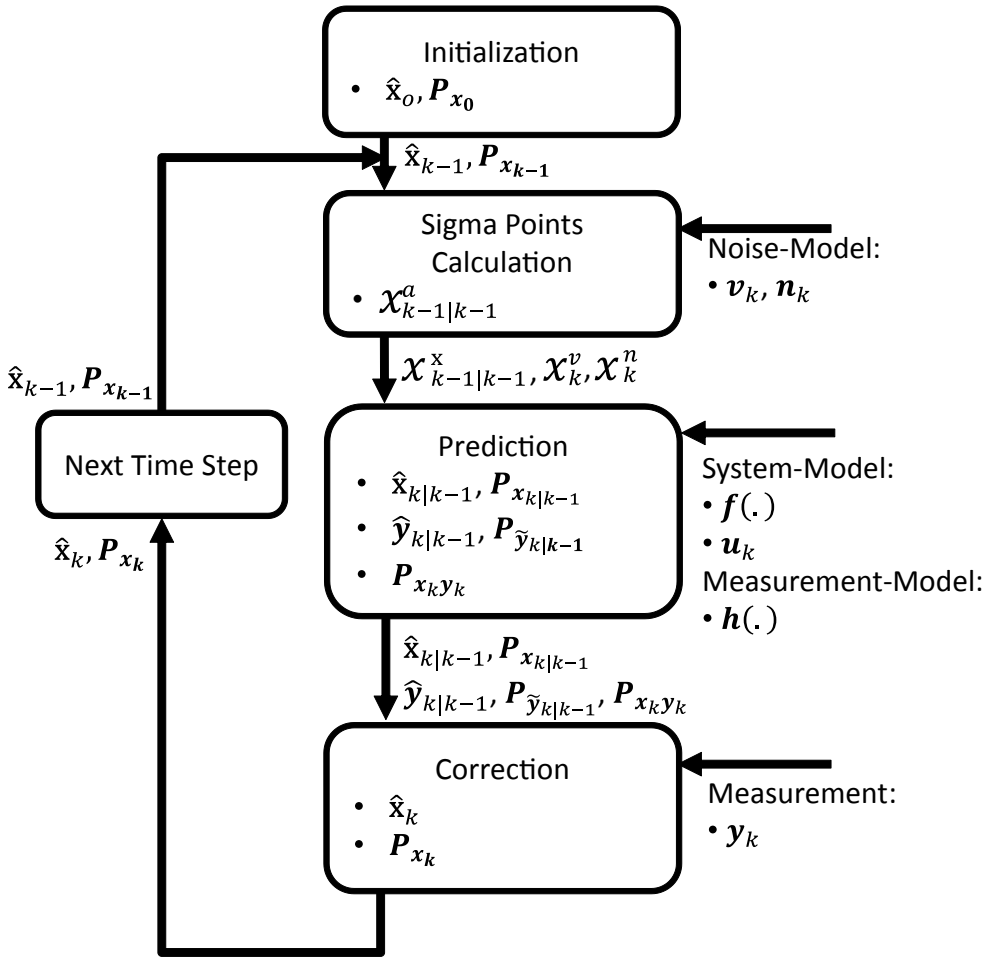


Figure 5.6: Flowchart of the main steps of Unscented Kalman Filter for state estimation.

5.4. Complete Model Description for 6 DOF Tracking

In this section the complete model for 6 DOF tracking will be presented. As already explained in section 2.4, the orientation will be expressed in terms of quaternions. The position and orientation will be expressed with respect to an arbitrary coordinate system chosen by the user (user-frame). The estimation is based on the gyroscope and accelerometer measurements, which are aided with the data from an optical measurement system and the magnetometer.

5.4.1. Process Model

In this section the strapdown navigation equations, presented in section 2.4 and the error models of the inertial sensors, presented in chapter 3, are combined to obtain the complete process model.

The process model is given with the following discrete-time transfer function:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{n}_k), \tag{5.32}$$

with $\mathbf{x}_k$, $\mathbf{u}_k$ and $\mathbf{n}_k$ as the state vector, the known exogenous input and process noise at time step k, respectively.

5.4.1.1. System States

The system states are defined as follows:

$$\mathbf{x}_k = \left[\left(\mathbf{q}_U^{B,k}\right)^T \left(\mathbf{p}_{B,k}^U\right)^T \left(\mathbf{v}_{UB,k}^U\right)^T \left(\mathbf{b}_{\omega_{\mathrm{RW}},k}\right)^T \left(\mathbf{b}_{a_{\mathrm{RW}},k}\right)^T \left(\mathbf{b}_{m_{\mathrm{RW}},k}\right)^T \left(\mathbf{g}_k^U\right)^T \left(\mathbf{m}_k^U\right)^T \left(\mathbf{r}_{\mathrm{lever},k}^B\right)^T \right]. \tag{5.33}$$

The vectors $\mathbf{q}_U^{B,k}$, $\mathbf{p}_{B,k}^U$ and $\mathbf{v}_{UB,k}^U$ are the states of interest. They describe the rotation from the user-frame to the body-frame, the position of the body-frame with respect to the user-frame and the velocity of the body-frame with respect to the user-frame measured in the user-frame at time step k, respectively.

The remaining states are auxiliary variables, in order to model the error characteristics of the sensors. $\mathbf{b}_{\omega_{\mathrm{RW}},k}$, $\mathbf{b}_{a_{\mathrm{RW}},k}$ and $\mathbf{b}_{m_{\mathrm{RW}},k}$ represent the RW biases for the gyroscope, accelerometer and magnetometer, respectively.

Furthermore, the true values of the reference vectors of the gravity-field $\mathbf{g}_k^U$ and the magnetic field $\mathbf{m}_k^U$ with respect to the user-frame, are not exactly known at the beginning of the estimation and are therefore added as states. However, when the accuracy of these parameters reaches a

predefined threshold, they could be removed as states and could be treated as constants. Their treatment as states makes it possible to refine their values online to improve the estimation results.
In addition, the magnetic field is affected by soft and hard iron distortions in its near environment such as tables, computers and other metallic objects (see section 3.1.2). In order to account for these effects, the magnetic field is modeled as RW.
The last entry of the state vector $\mathbf{r}^B_{\text{lever},k}$ represents the lever arm and is used to account for displacement of the sensor-frame with respect to the body-frame. This displacement is caused by constructive restrictions.

5.4.1.2. Exogenous Input

Two choices exist to incorporate the information of the gyroscope and the accelerometer measurements [22]. Either their information is used as output of the system and the relation to the states is given by the measurement function, or their information is used as exogenous input to the system and relation to the states is incorporated in the transfer-function. The first choice is preferred if the dynamic of the movement and rotation is precisely known and does not change over time. However, if this is not the case, the dynamic of the movement has to be estimated online. This can be difficult if the dynamic changes quickly, such as in user interaction.
In the latter method the estimation of the dynamic of the movement can be omitted and only the error dynamic of the sensors has to be modeled. Furthermore, no states for the angular rate and acceleration are needed. This results in a lower number of states needed in the filter and thus makes the algorithm stabler to numerical errors.
Due to the above mentioned reasons, the accelerometer and gyroscope measurements are used as exogenous input in the process model. The exogenous input vector is given as:

$$\mathbf{u}_k = \left[\boldsymbol{\omega}_k^T, \mathbf{a}_k^T, \Delta T_k\right], \tag{5.34}$$

with $\boldsymbol{\omega}_k$ as the angular rate and $\mathbf{a}_k$ as the acceleration at time step k measured by the IMU. ΔT_k represents the sampling time for time step k.
It should be noted that both measurements are still affected by deterministic and random errors.

5.4.1.3. Process Noise Input

The process noise models the random errors of the states and the exogenous input. It is given as:

$$\mathbf{n}_k = \left[(\mathbf{n}_{b,\omega_{\mathrm{RW}}})^T, (\mathbf{n}_{b,a_{\mathrm{RW}}})^T, (\mathbf{n}_{b,m_{\mathrm{RW}}})^T, (\mathbf{n}_{m_{\mathrm{RW}}})^T, (\mathbf{n}_{a,v,k})^T, (\mathbf{n}_{\omega,v,k})^T \right], \tag{5.35}$$

with $\mathbf{n}_{b,\omega_{\mathrm{RW}}}$, $\mathbf{n}_{b,a_{\mathrm{RW}}}$ and $\mathbf{n}_{b,m_{\mathrm{RW}}}$ being the zero mean white Gaussian noises driving the bias random walk of the gyroscope, accelerometer and magnetometer given their variance $\sigma^2_{b,\omega_{\mathrm{RW}}}$, $\sigma^2_{b,\mathrm{RW}}$, $\sigma^2_{b,m_{\mathrm{RW}}}$, respectively. $\mathbf{n}_{m_{RW}}$ is the zero mean white Gaussian noise driving the RW of the reference vector of the magnetic field with its variance $\sigma^2_{m_{\mathrm{RW}}}$. The variable $\sigma^2_{a,v}$ and $\sigma^2_{\omega,v}$ are the variances of the measurement noises for the accelerometer $\mathbf{n}_{a,v,k}$ and the gyroscope $\mathbf{n}_{\omega,v,k}$, respectively. They are composed of a constant part and a magnitude depending part, as defined in section 3.2.1. For the sake of completeness their relation is presented again:

$$\sigma^2_{\mathrm{Dev},v,k} = \sigma^2_{\mathrm{Dev,noise}} + \sigma^2_{\mathrm{Dev,NL},0} \cdot \frac{|\mathrm{Dev}_k|}{0.5 \cdot \mathrm{FSR}}. \tag{5.36}$$

The variable 'Dev' is in this case a placeholder for the measurement device, like 'a' for the accelerometer and 'ω' for the gyroscope.
It will be assumed that random errors are not correlated to each other. Therefore only the variance for each channel is given. This results in a diagonal covariance matrix with the given variances as follows:

$$\mathbf{\Sigma}^2 = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}. \tag{5.37}$$

The covariance for the whole process noise vector is summarized as:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{\Sigma}^2_{b,\omega_{\mathrm{RW}}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}^2_{b,a_{\mathrm{RW}}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}^2_{b,m_{\mathrm{RW}}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}^2_{m_{\mathrm{RW}}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}^2_{a,v,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}^2_{\omega,v,k} \end{bmatrix}. \tag{5.38}$$

5.4.1.4. State Transition Function

The transfer function for the process model is given as:

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{q}_U^{B,k+1} \\ \mathbf{p}_{B,k+1}^U \\ \mathbf{v}_{UB,k+1}^U \\ \mathbf{b}_{\omega_{\mathrm{RW}},k+1} \\ \mathbf{b}_{a_{\mathrm{RW}},k+1} \\ \mathbf{b}_{m_{\mathrm{RW}},k+1} \\ \mathbf{g}_{k+1}^U \\ \mathbf{m}_{k+1}^U \\ \mathbf{r}_{\mathrm{lever},k+1}^B \end{bmatrix} = \begin{bmatrix} \mathbf{f}_q(\mathbf{q}_U^{B,k}, \mathbf{b}_{\omega_{\mathrm{RW}},k}, \boldsymbol{\omega}_k, \Delta T_k, \mathbf{n}_{\omega,v,k}) & \\ \mathbf{f}_p(\mathbf{q}_U^{B,k+1}, \mathbf{p}_{B,k}^U, \mathbf{v}_{UB,k}^U, \mathbf{b}_{a_{\mathrm{RW}},k+1}, \mathbf{g}_k^U, \mathbf{a}_k, \Delta T_k, \mathbf{n}_{a,v,k}) & \\ \mathbf{f}_v(\mathbf{q}_U^{B,k+1}, \mathbf{v}_{UB,k}^U, \mathbf{b}_{a_{\mathrm{RW}},k+1}, \mathbf{g}_k^U, \mathbf{a}_k, \Delta T_k, \mathbf{n}_{a,v,k}) & \\ \mathbf{b}_{\omega_{\mathrm{RW}},k} & + \mathbf{n}_{b,\omega_{\mathrm{RW}}} \\ \mathbf{b}_{a_{\mathrm{RW}},k} & + \mathbf{n}_{b,a_{\mathrm{RW}}} \\ \mathbf{b}_{m_{\mathrm{RW}},k} & + \mathbf{n}_{b,m_{\mathrm{RW}}} \\ \mathbf{g}_k^U & \\ \mathbf{m}_k^U & + \mathbf{n}_{m_{\mathrm{RW}}} \\ \mathbf{r}_{\mathrm{lever},k}^B & \end{bmatrix}. \tag{5.39}$$

The transfer function is composed of two parts. The first part includes $\mathbf{f}_q$, $\mathbf{f}_p$ and $\mathbf{f}_v$ and represents the dynamic of a rigid body. The second part consists of auxiliary variables to improve the estimation results.

The dynamic of a rigid body has been derived in section 2.4, but only perfect measurements were assumed. In this section the navigation equations are combined with the error models from chapter 3, like deterministic and random errors.

Quaternion-Update

To ease the readability, the calculation of the quaternion update function $\mathbf{f}_q(\cdot)$ is split up in several smaller parts.

The orientation is estimated in the user-frame, whereas the navigation equations from section 2.4 are derived in the navigation-frame. Therefore the equation has to be rotated from the navigation-frame to the user-frame. Taking the quaternion update from equation (2.29) and performing the rotation $\mathbf{q}_U^N$ on both sides, yields:

$$\mathbf{q}_N^{B,k+1} \bullet \mathbf{q}_U^N = \mathbf{q}_{B,k}^{B,k+1} \bullet \mathbf{q}_N^{B,k} \bullet \mathbf{q}_U^N, \tag{5.40}$$

$$\mathbf{q}_U^{B,k+1} = \mathbf{q}_{B,k}^{B,k+1} \bullet \mathbf{q}_U^{B,k}. \tag{5.41}$$

The quaternion $\mathbf{q}_{B,k}^{B,k+1}$ can be expressed in terms of a rotation vector $\Delta\boldsymbol{\sigma}_{B_k}^{B_{k+1}}$ as:

$$\mathbf{q}_U^{B,k+1} = \mathbf{f}_q(\cdot) = \begin{bmatrix} cos\left(\frac{\left\|\Delta\boldsymbol{\sigma}_{B_k}^{B_{k+1}}\right\|}{2}\right) \\ \frac{\boldsymbol{\sigma}_{B_k}^{B_{k+1}}}{\left\|\Delta\boldsymbol{\sigma}_{B_k}^{B_{k+1}}\right\|} \cdot sin\left(\frac{\left\|\Delta\boldsymbol{\sigma}_{B_k}^{B_{k+1}}\right\|}{2}\right) \end{bmatrix} \bullet \mathbf{q}_U^{B,k}. \tag{5.42}$$

The rotation vector $\Delta\boldsymbol{\sigma}_{B,k}^{B,k+1}$, which rotates the body-frame from time step k to time step k+1, is calculated from the corrected gyroscope measurement $\boldsymbol{\omega}_{\text{corr},k}$ and the sampling time ΔT_k as:

$$\Delta\boldsymbol{\sigma}_{B,k}^{B,k+1} = (\boldsymbol{\omega}_{\text{corr},k} \cdot \Delta T_k) + \frac{1}{12}\left[(\boldsymbol{\omega}_{\text{corr},k-1} \cdot \Delta T_{k-1}) \times (\boldsymbol{\omega}_{\text{corr},k} \cdot \Delta T_k)\right]. \tag{5.43}$$

The gyroscope measurement $\boldsymbol{\omega}_{\text{corr},k}$, which is corrected for deterministic and random errors, is obtained as follows:

$$\boldsymbol{\omega}_{\text{corr},k} = [\mathbf{M}_\omega \cdot (\boldsymbol{\omega}_k - \mathbf{b}_{\omega,d})] - \mathbf{b}_{\omega_{\text{RW}},k} - \mathbf{n}_{\omega,v,k}. \tag{5.44}$$

The scale factor matrix $\mathbf{M}_\omega$ and the deterministic bias $\mathbf{b}_{\omega,d}$ are obtained from the calibration procedure.

Velocity-Update

Analog to the quaternion update, the calculation of the velocity update function $\mathbf{f}_v(\cdot)$ is split up in smaller parts to ease the readability. It is given as follows:

$$\mathbf{v}_{UB,k+1}^U = \mathbf{f}_v(\cdot) = \mathbf{v}_{UB,k}^U + \Delta\mathbf{v}_{UB,a,k}^U + \Delta\mathbf{v}_{UB,g,k}^U. \tag{5.45}$$

In section 2.4 the velocity update due to the specific force was derived in the navigation-frame and therefore has to be adapted for the user-frame. Taking equation (2.36) and performing the rotation from the navigation-frame to the user-frame yields to:

$$\Delta\mathbf{v}_{UB,a,k}^U = \mathbf{R}_N^U\mathbf{R}_{B,k+1}^N\left(\Delta\mathbf{v}_{IB,k}^B + \frac{1}{2}\Delta\boldsymbol{\theta}_k \times \Delta\mathbf{v}_{IB,k}^B + \frac{1}{12}\left(\Delta\boldsymbol{\theta}_k \times \Delta\mathbf{v}_{IB,k-1}^B + \Delta\mathbf{v}_{IB,k}^B \times \Delta\boldsymbol{\theta}_{k-1}\right)\right), \tag{5.46}$$

$$= \mathbf{R}_{B,k+1}^U\left(\Delta\mathbf{v}_{IB,k}^B + \frac{1}{2}\Delta\boldsymbol{\theta}_k \times \Delta\mathbf{v}_{IB,k}^B + \frac{1}{12}\left(\Delta\boldsymbol{\theta}_k \times \Delta\mathbf{v}_{IB,k-1}^B + \Delta\mathbf{v}_{IB,k}^B \times \Delta\boldsymbol{\theta}_{k-1}\right)\right). \tag{5.47}$$

The rotation matrix $\mathbf{R}_{B,k+1}^U$ can either be constructed by using the quaternion $\mathbf{q}_U^{B,k+1}$ or the rotation is performed using quaternions directly (equation (2.18)).

The velocity increment $\Delta\mathbf{v}_{IB,k}^B$ and the angle increment $\Delta\boldsymbol{\theta}_k$ can be calculated from the corrected measurements as:

$$\Delta\mathbf{v}_{IB,k}^B = \mathbf{a}_{\text{corr},k} \cdot \Delta T_k, \tag{5.48}$$

$$\Delta\boldsymbol{\theta}_k = \boldsymbol{\omega}_{\text{corr},k} \cdot \Delta T_k. \tag{5.49}$$

The accelerometer measurement $\boldsymbol{a}_{\text{corr},k}$, which is corrected for deterministic and random errors, is obtained as follows:

$$\boldsymbol{a}_{\text{corr},k} = [\mathbf{M}_a \cdot (\boldsymbol{a}_k - \mathbf{b}_{a,d})] - \mathbf{b}_{a_{\text{RW}},k} - \mathbf{n}_{a,v,k}. \tag{5.50}$$

The scale factor matrix $\mathbf{M}_a$ and the deterministic bias $\mathbf{b}_{a,d}$ are obtained from the calibration procedure. By using the estimate of the gravity field $\mathbf{g}_k^U$, the velocity increment due to gravity is obtained as:

$$\Delta\mathbf{v}_{UB,g,k}^U = \mathbf{g}_k^U \cdot \Delta T_k. \tag{5.51}$$

Position-Update

The position update function $\mathbf{f}_p(\cdot)$ is given as:

$$\mathbf{p}_{B,k+1}^U = \mathbf{f}_p(\cdot) = \mathbf{p}_{B,k}^U + \mathbf{v}_{UB,k}^U \cdot \Delta T_k + \left[\Delta\mathbf{v}_{UB,a,k}^U + \Delta\mathbf{v}_{UB,g,k}^U\right] \cdot \frac{\Delta T_k}{2}. \tag{5.52}$$

5.4.2. Measurement Models

In this section the measurement models of the magnetometer and the monocular position tracking system will be presented. It should be noted that measurements from any other device can be used, if their measurement functions and covariance matrices are known. Furthermore, the representation of the measurement result has to be mathematically closed for addition and subtraction. It should be noted here, that quaternions are not closed for addition and subtraction. This issue will be further discussed in section 5.5.2.

5.4.2.1. Magnetometer

The discrete time measurement function $\mathbf{h}_m(\mathbf{x}_k, \mathbf{v}_k)$, which can be used to obtain an estimate $\hat{\mathbf{m}}_k^B$ for the real magnetometer measurement $\mathbf{m}_k^B$, is given as:

$$\hat{\mathbf{m}}_k^B = \mathbf{h}_m(\mathbf{x}_k, \mathbf{v}_{m,k}) = \mathbf{R}_U^{B,k} \cdot \mathbf{m}_k^U + \mathbf{b}_{m_{\text{RW}},k} + \mathbf{v}_{m,k}. \tag{5.53}$$

The rotation matrix $\mathbf{R}_U^{B,k}$ can be either constructed by using the quaternion $\mathbf{q}_U^{B,k}$ or the rotation can be performed using quaternions directly (equation (2.18)). The variable $\mathbf{m}_k^U$ represents the estimation of the magnetic field for time step k, whereas $\mathbf{b}_{m_{\text{RW}},k}$ is the bias random walk. The measurement noise $\mathbf{v}_{m,k}$ is modeled as a zero mean white Gaussian noise with variance $\sigma_{m,\text{noise}}^2$. It should be noted that the measured magneto measurement $\mathbf{m}_k^B$ will be compensated for

deterministic errors, before feeding it to the UKF. The compensation was derived in section 3.1.2 and will be presented here again for completeness:

$$\mathbf{m}^B_{\text{corr},k} = \mathbf{M}_m \cdot \left(\boldsymbol{m}^B_k - \mathbf{b}_{m,d} \right). \tag{5.54}$$

The scale factor matrix $\mathbf{M}_m$ and the deterministic bias $\mathbf{b}_{m,d}$ are obtained in the calibration procedure.

5.4.2.2. Position Information (Monocular Position Tracking)

The position measurements, which are obtained by the monocular position tracking system, are given with respect to the center of the object. However, due to constructive restrictions, the IMU cannot be located at the center of the object.
The influence of this displacement $\mathbf{r}_{\text{lever}}$, also called lever arm effect, can be modeled in two ways:
Either the body-frame is originated at the center of the object and the lever arm effect is included by modifying the navigation equations. Or the body-frame is located at the origin of the IMU. In this case the lever arm will be included in the measurement equation of the monocular position tracking system. For the first approach the additional acceleration $\mathbf{a}_{lever}$ due to the lever arm effect, is given as [59]:

$$\mathbf{a}_{\text{lever}} = \dot{\boldsymbol{\omega}} \times \mathbf{r}_{\text{lever}} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{\text{lever}}). \tag{5.55}$$

For the compensation time derivatives of the angular rotations $\boldsymbol{\omega}$ are required. However, the measurements are affected by noise and numerical derivatives could yield to inaccurate results. For the second approach the calculated position has to be shifted by the lever arm $\Delta\mathbf{p}_{\text{lever},k}$. To do so, the lever arm has to be rotated from the body-frame to the user-frame, by using the current estimation of the orientation $\mathbf{R}^U_{B,k}$. The lever arm expressed in the user-frame is given as:

$$\Delta\mathbf{p}_{\text{lever},k} = \mathbf{R}^U_{B,k} \cdot \mathbf{r}_{\text{lever}}. \tag{5.56}$$

Since only one rotation is required, the second approach was chosen.
The measurement function to calculate the objects position $\hat{\mathbf{p}}^U_{pm,k}$ is given as:

$$\hat{\mathbf{p}}^U_{pm,k} = \mathbf{h}_p(\mathbf{x}_k, \mathbf{v}_{p,k}) \tag{5.57}$$

$$= \mathbf{p}^U_{B,k} + \Delta\mathbf{p}_{\text{lever},k} + \mathbf{v}_{p,k}. \tag{5.58}$$

The measurement noise $\mathbf{v}_{p,k}$ is modeled as a zero mean white Gaussian noise with covariance matrix $\mathbf{R}_{p,U}$.

It should be noted that measurements from the monocular position tracking system are provided in camera coordinates $\mathbf{p}_{pm,k}^{\text{Cam}}$. Therefore these measurements have to be transformed to the user-frame, before feeding them to the UKF. This transformation is given as:

$$\mathbf{p}_{pm,k}^{U} = \mathbf{R}_{\text{Cam}}^{U} \cdot (\mathbf{p}_{pm,k}^{\text{Cam}} - \mathbf{p}_{U}^{\text{Cam}}). \tag{5.59}$$

Usually, the covariance matrix for the measurement noise $\mathbf{v}_{p,k}$ is estimated in the camera-frame $\mathbf{R}_{p,\text{Cam}}$ and thus also has to be transformed into the user-frame:

$$\mathbf{R}_{p,U} = \mathbf{R}_{\text{Cam}}^{U} \cdot \mathbf{R}_{p,\text{Cam}} \cdot \left(\mathbf{R}_{\text{Cam}}^{U}\right)^{T}. \tag{5.60}$$

The offset $\mathbf{p}_{U}^{\text{Cam}}$ and the rotation matrix $\mathbf{R}_{\text{Cam}}^{U}$ are estimated in the initialization phase of the filter. This will be further explained in section 6.2.1.

5.5. Adaptations

5.5.1. Method to Deal with Asynchronous Data at different Sampling Rates

In general the obtained data, such as measurements are not synchronized and do not have the same sampling rate. The data from IMU, such as accelerometer and gyroscope measurements, are available with a much higher sampling rate, than measurements for position or other aided sensors. In some cases the gyroscope can even run at a higher rate than the accelerometer. The MPU-9150 could provide gyroscope measurements with a rate up to 8 kHz and a rate up to 1 kHz for the accelerometer.

To challenge this problem, the UKF was set to run with the frequency of the sensor with the highest sampling rate. By assuming piece-wise constant measurements, the exogenous inputs are hold constant until a newer measurement is available.

Depending on the actual number of available measurements the measurement vector $\mathbf{y}_k$ and the measurement noise $\mathbf{R}_k$ are stacked at each iteration as follows:

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_{1,k} \\ \mathbf{y}_{2,k} \\ \vdots \\ \mathbf{y}_{m,k} \end{bmatrix}, \tag{5.61}$$

$$\mathbf{R}_k = \begin{bmatrix} \mathbf{R}_{1,k} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{2,k} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{R}_{m,k} \end{bmatrix}, \tag{5.62}$$

with $\mathbf{y}_{m,k}$ as the available measurement at time step k with its covariance matrix $\mathbf{R}_{k,m}$. In a similar way the measurement function $\mathbf{h}_k$ is stacked as follows:

$$\mathbf{h}_k = \begin{bmatrix} \mathbf{h}_{1,k} \\ \mathbf{h}_{2,k} \\ \vdots \\ \mathbf{h}_{m,k} \end{bmatrix}, \tag{5.63}$$

with $\mathbf{h}_{m,k}$ as the measurement function for the measurement $\mathbf{y}_{m,k}$, respectively.

5.5.2. Modification of the UKF for the Use of Quaternions

The UKF-framework presented in section 5.3.3 cannot be used straight forward, since quaternions are not closed for addition and subtractions [19]. In order to use quaternions, the following adaptations are introduced. It is based on the Quaternion Unscented Kalman Filter in [60]. The basic idea consists of switching between quaternions and rotation vectors in different parts of the filter framework. Quaternions are used to present actual states of the orientation, whereas rotation vectors are used to operate the covariance of the orientation. The augmented state vector $\mathbf{x}_k^a$ can be split up in a quaternion part $\mathbf{x}_k^{\text{quat}}$ and a remaining part $\mathbf{x}_k^{\text{rest}}$ as:

$$\mathbf{x}_k^a = \begin{bmatrix} \mathbf{x}_k^{\text{quat}} \\ \mathbf{x}_k^{\text{rest}} \end{bmatrix}. \tag{5.64}$$

5.5.2.1. Calculation of Sigma Points

The sigma points can be interpreted as perturbed state vectors. The perturbation $\Delta\mathbf{x}_k^{\text{quat}}$ of a quaternion can be expressed as the following rotation:

$$\tilde{\mathbf{x}}_k^{\text{quat}} = \Delta\mathbf{x}_k^{\text{quat}} \bullet \mathbf{x}_k^{\text{quat}}, \tag{5.65}$$

with $\tilde{\mathbf{x}}_k^{\text{quat}}$ as the perturbed quaternion and $\mathbf{x}_k^{\text{quat}}$ as the original quaternion. By using equation (5.20) for the remaining part and equation (5.65) for the quaternion part, the calculation of the complete sigma points is given as follows:

$$\mathcal{X}_{k-1}^a = \begin{bmatrix} \mathcal{X}_{k-1}^{\text{quat}} \\ \mathcal{X}_{k-1}^{\text{rest}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_{k-1}^{\text{quat}} & \Delta\mathcal{X}_{k-1}^a \bullet \hat{\mathbf{x}}_{k-1}^{\text{quat}} & (\Delta\mathcal{X}_{k-1}^a)^{-1} \bullet \hat{\mathbf{x}}_{k-1}^{\text{quat}} \\ \hat{\mathbf{x}}_{k-1}^{\text{rest}} & \hat{\mathbf{x}}_{k-1}^{\text{rest}} + \Delta\mathcal{X}_{k-1}^{\text{rest}} & \hat{\mathbf{x}}_{k-1}^{\text{rest}} - \Delta\mathcal{X}_{k-1}^{\text{rest}} \end{bmatrix}. \tag{5.66}$$

The perturbation vector for the orientation $\Delta\mathcal{X}_{k-1}^{\sigma}$, in terms of rotation vectors and remaining states $\Delta\mathcal{X}_{k-1}^{\text{rest}}$, is given by the augmented covariance matrix as:

$$\begin{bmatrix} \Delta\mathcal{X}_{k-1}^{\sigma} \\ \Delta\mathcal{X}_{k-1}^{\text{rest}} \end{bmatrix} = \left(\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{x}_{k-1}}^a} \right). \tag{5.67}$$

By taking the relation between rotation vectors and quaternions, the perturbation vector for quaternions $\Delta\mathcal{X}_{k-1}^{\text{quat}}$ is obtained as:

$$\Delta\mathcal{X}_{k-1}^{\text{quat}} = \begin{bmatrix} cos\left(\dfrac{\left\| \Delta\mathcal{X}_{k-1}^{\sigma} \right\|}{2} \right) \\ \dfrac{\Delta\mathcal{X}_{k-1}^{\sigma}}{\left\| \Delta\mathcal{X}_{k-1}^{\sigma} \right\|} \cdot sin\left(\frac{\left\| \Delta\mathcal{X}_{k-1}^{\sigma} \right\|}{2} \right) \end{bmatrix}. \tag{5.68}$$

5.5.2.2. Prediction

The *a priori* sigma points are simply obtained by using the process model transfer function:

$$\mathcal{X}^x_{k|k-1} = \mathbf{f}(\mathcal{X}^x_{k-1}, \mathcal{X}^v_k, \mathbf{u}_k), \tag{5.69}$$

where for the calculation of the *a priori* mean and its covariance matrix some additional steps have to be considered.

Calculation of the *a priori* Mean

The *a priori* mean is concatenated as:

$$\hat{\mathbf{x}}_{k|k-1} = \begin{bmatrix} (\hat{\mathbf{x}}^{\text{quat}}_{k|k-1}) \\ (\hat{\mathbf{x}}^{\text{rest}}_{k|k-1}) \end{bmatrix}, \tag{5.70}$$

where the weighted mean of the remaining states, is calculated as follows:

$$(\hat{\mathbf{x}}^{\text{rest}}_{k|k-1}) = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{X}^{\text{rest}}_{i,k|k-1}. \tag{5.71}$$

For the calculation of the *a priori* mean quaternion $(\hat{\mathbf{x}}^{\text{quat}}_{k|k-1})$, the intrinsic gradient descent algorithm according to [61] will be used.

The algorithm starts by choosing a start value for the *a priori* mean quaternion. A good guess is usually given with the predicted *a posteriori* state of the last time step $k-1$. This value is stored in the first element of the *a priori* sigma points $(\Delta\mathcal{X}^{\text{quat}}_{k|k-1})$.

In the next step, the rotation, between the *a priori* mean $(\hat{\mathbf{x}}^{\text{quat}}_{k|k-1})_t$ and the sigma points $(\mathcal{X}^{\text{quat}}_{k|k-1})_{t,i}$, is calculated. In terms of delta quaternions $(\Delta\mathcal{X}^{\text{quat}}_{k|k-1})_{t,i}$, this rotation is given as:

$$(\Delta\mathcal{X}^{\text{quat}}_{k|k-1})_{t,i} = (\mathcal{X}^{\text{quat}}_{k|k-1})_{t,i} \bullet \text{inv}\left((\hat{\mathbf{x}}^{\text{quat}}_{k|k-1})_t\right). \tag{5.72}$$

The variable t stands for the t-th iteration step and i stands for the i-th column of the set of sigma points.

Subsequently, the *a priori* mean quaternion is iteratively updated by using the barycentric mean $\bar{e}_t$ of the delta quaternions as follows:

$$(\hat{\mathbf{x}}^{\text{quat}}_{k|k-1})_{t+1} = \bar{e}_t \bullet (\hat{\mathbf{x}}^{\text{quat}}_{k|k-1})_t, \tag{5.73}$$

with the barycentric mean $\bar{e}_t$ given by:

$$\bar{e}_t = \frac{1}{2L+1} \sum_{i=0}^{2L} \left[(\Delta\mathcal{X}^{\text{quat}}_{k|k-1})_{t,i}\right]. \tag{5.74}$$

The optimization is stopped, when the average angle of rotation θ reaches a defined criterion. θ can be calculated from first element of the mean delta quaternion as:

$$\theta = 2 \cdot \arccos(\bar{e}_1). \tag{5.75}$$

It should be noted that the algorithm usually converges within 5 iterations to a value $\theta \leq 1 \times 10^{-20}$ rad.

Calculation of the *a priori* Covariance Matrix

The *a priori* perturbation vectors for the quaternions $(\Delta\mathcal{X}^{\text{quat}}_{k|k-1})$ are already calculated in the last iteration step in the calculation of the mean quaternion. Their corresponding rotation vectors can be obtained as:

$$(\Delta\mathcal{X}^{\sigma}_{k|k-1}) = \left[arcos\left((\Delta\mathcal{X}^{\text{quat}}_{k|k-1})_s\right) \cdot \left((\Delta\mathcal{X}^{\text{quat}}_{k|k-1})_v\right)\right], \tag{5.76}$$

with $(\Delta\mathcal{X}^{\text{quat}}_{k|k-1})_s$ and $(\Delta\mathcal{X}^{\text{quat}}_{k|k-1})_v$ as the scalar and vector part of the quaternion, respectively. The weighted covariance matrix is obtained from the concatenated perturbation vector as:

$$\mathbf{P}_{\mathbf{x}_{k|k-1}} = \sum_{i=0}^{2L} w_i^{(c)} \begin{bmatrix} (\Delta\mathcal{X}^{\sigma}_{k|k-1}) \\ (\Delta\mathcal{X}^{\text{rest}}_{k|k-1}) \end{bmatrix}_i \cdot \begin{bmatrix} (\Delta\mathcal{X}^{\sigma}_{k|k-1}) \\ (\Delta\mathcal{X}^{\text{rest}}_{k|k-1}) \end{bmatrix}_i^T, \tag{5.77}$$

with

$$(\Delta\mathcal{X}^{\text{rest}}_{k|k-1}) = \mathcal{X}^{\text{rest}}_{k|k-1} - (\hat{\mathbf{x}}^{\text{rest}}_{k|k-1}). \tag{5.78}$$

5.5.2.3. Correction

The *a priori* sigma points $\mathcal{Y}_{k|k-1}$ of the expected measurements, its according mean $\hat{\mathbf{y}}_{k|k-1}$ and covariance matrix $\mathbf{P}_{\tilde{\mathbf{y}}_k}$ are calculated using the equation from the standard UKF:

$$\mathcal{Y}_{k|k-1} = \mathbf{h}(\mathcal{X}^x_{k|k-1}, \mathcal{X}^n_{k-1}), \tag{5.79}$$

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{Y}_{i,k|k-1}, \tag{5.80}$$

$$\mathbf{P}_{\tilde{\mathbf{y}}_{k|k-1}} = \sum_{i=0}^{2L} w_i^{(c)} (\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T. \tag{5.81}$$

However, the cross-covariance matrix between measurements and states is calculated as:

$$\mathbf{P}_{\mathbf{x}_k,\mathbf{y}_k} = \sum_{i=0}^{2L} w_i^{(c)} \begin{bmatrix} (\Delta\mathcal{X}^{\sigma}_{k|k-1}) \\ (\Delta\mathcal{X}^{\text{rest}}_{k|k-1}) \end{bmatrix}_i \cdot \left[\Delta\mathcal{Y}_{k|k-1}\right]_i^T, \tag{5.82}$$

with

$$\Delta\mathcal{Y}_{k|k-1} = (\mathcal{Y}_{k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T. \tag{5.83}$$

The Kalman gain $\mathbf{K_k}$, the correction term $\Delta\mathbf{x}_k$, as well as the *a posteriori* covariance matrix $\mathbf{P}_{\mathbf{x}_k}$, are again calculated using the standard UKF filter equations:

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k\mathbf{y}_k}\mathbf{P}^{-1}_{\tilde{\mathbf{y}}_{k|k-1}}, \tag{5.84}$$

$$\Delta\mathbf{x}_k = \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}), \tag{5.85}$$

$$\mathbf{P}_{\mathbf{x}_k} = \mathbf{P}_{\mathbf{x}_{k|k-1}} - \mathbf{K}_k\mathbf{P}_{\tilde{\mathbf{y}}_{k|k-1}}\mathbf{K}_k^T. \tag{5.86}$$

For the calculation of the *a posterior* estimation, the quaternion part and the remaining part has to be treated differently. The correction term $\Delta\mathbf{x}_k$ was calculated with the cross-covariance matrix $\mathbf{P}_{\mathbf{x}_k,\mathbf{y}_k}$ based on the rotation vectors. For that reason the orientational part of $\Delta\mathbf{x}_k$ is presented as a rotation vector:

$$\Delta\mathbf{x}_k = \begin{bmatrix} \Delta\mathbf{x}_k^{\sigma} \\ \Delta\mathbf{x}_k^{\text{rest}} \end{bmatrix}. \tag{5.87}$$

The delta quaternion, which performs the correction, can be calculated as:

$$\Delta\mathbf{x}_k^{\text{quat}} = \begin{bmatrix} cos\left(\dfrac{\left\|\Delta\mathbf{x}_k^{\sigma}\right\|}{2}\right) \\ \dfrac{\Delta\mathbf{x}_k^{\sigma}}{\left\|\Delta\mathbf{x}_k^{\sigma}\right\|} \cdot sin\left(\dfrac{\left\|\Delta\mathbf{x}_k^{\sigma}\right\|}{2}\right) \end{bmatrix}. \tag{5.88}$$

Finally the *a posterior* estimation can be given as:

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\mathbf{x}}_k^{\text{quat}} \\ \hat{\mathbf{x}}_k^{\text{rest}} \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{x}_k^{\text{quat}} \bullet (\hat{\mathbf{x}}_{k|k-1}^{\text{quat}}) \\ (\hat{\mathbf{x}}_{k|k-1}^{\text{rest}}) + \Delta\mathbf{x}_k^{\text{rest}} \end{bmatrix}. \tag{5.89}$$

5.5.3. Timing Delays

Additionally to asynchronous data and different sampling rates, the sensors have different delays. On the first glance someone might assume that these effects can be neglected. However, some pretests showed significant performance losses for uncompensated estimations.
Basically the information is obtained from two different sources. The first part is retrieved from the IMU, such as gyroscope, accelerometer and magnetometer measurements. For those measurements, it can be assumed that they have the same delay, which is due to data transmission.
The second source is the position information from the monocular position tracking system, which is mainly delayed by two processes. In the first, the camera has to capture the image and transfer a big amount of data to the computer. In the second, position information is extracted from the images.
It is assumed that the position information system has a significant larger delay than the data received from the IMU.

5.5.3.1. Compensation

In order to compensate the timing delays a simple approach has been chosen. The measurements from the IMU are delayed in order to be timely aligned with the position information. In addition, a constant time delay t_{delay} between the inertial sensors and the position information is assumed. The compensation can be achieved by using a data queue with the following size:

$$n_{\text{queue}} = \text{round}\left(t_{\text{delay}} \cdot f_{\text{Filter}}\right), \tag{5.90}$$

with n_{queue} as the size of the queue and f_{Filter} as the frequency of the UKF.

5.5.3.2. Estimation of Delays

Data Transmission Delay

The approach to estimate the data transmission delay is demonstrated in figure 5.7. A μC sets the LED-intensity according to a known sequence. Every time a new value for the LEDs is set, the information is also sent to the computer using the same connection as the IMU. At the same time the optical system captures the time varying intensity of the light. By taking the average of all pixel intensities, a reference sequence of the intensity for the optical system is obtained. Using correlation analysis, the data transmission delay can be calculated.

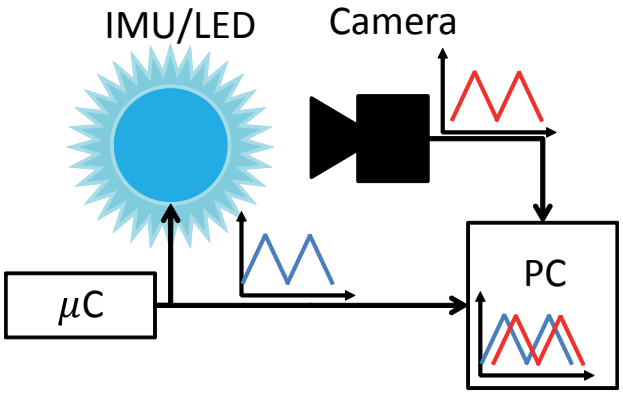


Figure 5.7: Experimental setup to measure time delay.

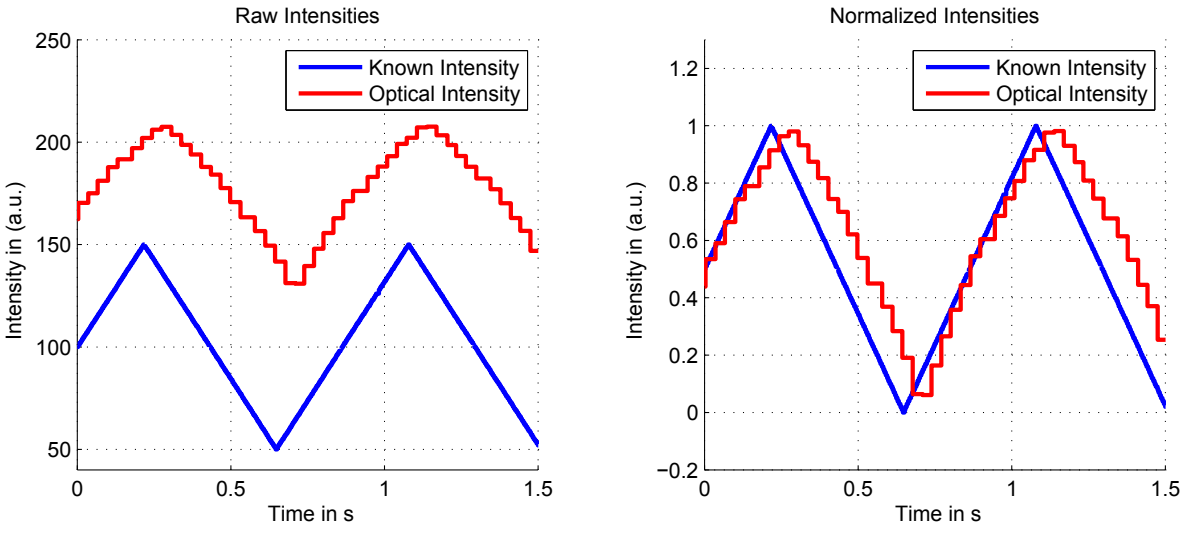
In the following, the steps for the correlation analysis will be presented:

1. **Normalization**

   The first step is to remove the scale-errors and offset for both sequences. The normalized sequence $\mathbf{x}_{\text{norm}}$ is given as:

$$\mathbf{x}_{\text{norm}} = \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}. \tag{5.91}$$

   In figure 5.8 the sequence before and after normalization is presented.



(a) (b)

Figure 5.8: (a) Raw intensities, (b) Normalized intensities;
The set/known LED-intensity in blue and the measured optical intensity in red.

2. **Interpolation**

   The two sequences are presented in different time-bases and therefore have to be interpolated to a common time-base ΔT_{base}. It should be noted, that the smaller the time-base, the higher the resolution of the estimated delay. A time base of $\Delta T_{\text{base}} = 50\,\mu\text{s}$ has been chosen for the interpolation.

3. **Coarse alignment**

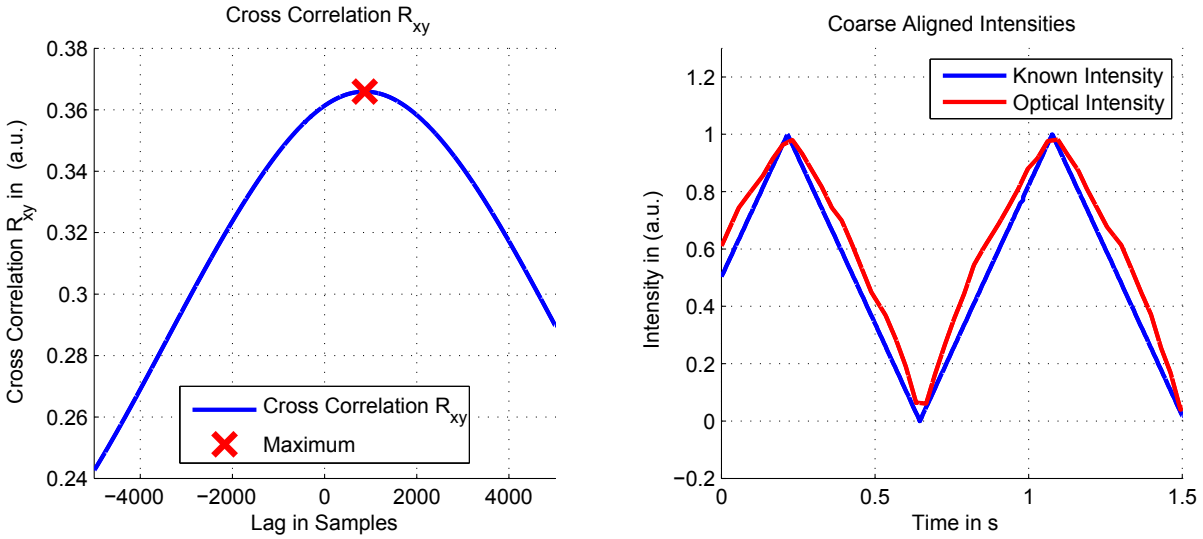
   For the alignment, the cross-correlation function between both sequences is calculated according to:

$$R_{xy}(l) = \frac{1}{M} \sum_{m=1}^{M} x(m) \cdot y(m+l). \tag{5.92}$$

   The time shift n_{delay}, which is expressed in multiples of the time-base ΔT_{base}, is given at the maximum of the cross-correlation function. Then the delay in seconds t_{delay} is calculated according to:

$$t_{\text{delay,transmission}} = n_{\text{delay}} \cdot \Delta T_{\text{base}}. \tag{5.93}$$

   A zoomed part of cross-correlation function and the coarse aligned intensities are shown in figure 5.9.



(a) (b)

Figure 5.9: (a) Cross correlation function $R_{x,y}$ with a time base of $\Delta T_{\text{base}} = 50\,\mu\text{s}$, (b) Coarse aligned intensities; known intensity (blue) and optical intensity (red).

4. **Compensation for Nonlinearities**

   Figure 5.9b shows that the intensity of the optical system follows a nonlinear function. This is even more evident when the intensity of the optical system is plotted against the known intensity (see figure 5.10).

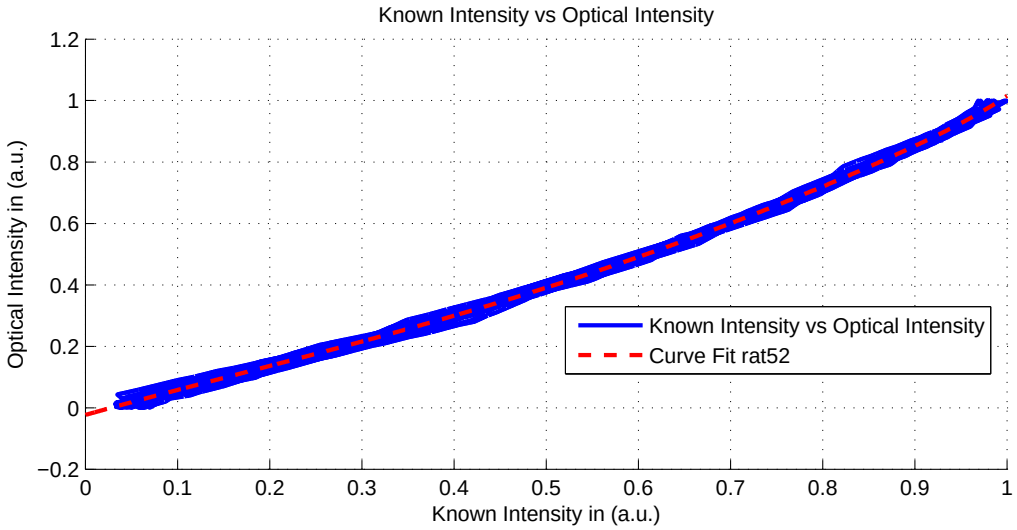


Figure 5.10: Nonlinearity of the optical intensity.

   In order to compensate the distortion, a rational function is fitted using a numerator degree of 5 and denominator degree of 2. The compensation is performed using a Look up Table (LUT) based on the fit.

5. **Fine Alignment**

   The steps for fine alignment are identical to the coarse alignment, with the exception that the compensated intensities of the optical system are used for the correlation function. The result of the compensated and fine aligned intensity sequences are shown in figure 5.11. The following delay of the data transmission of the optical system with respect to the IMU was found:

$$t_{\text{delay,transmission}} = 43.65\,\text{ms}. \tag{5.94}$$

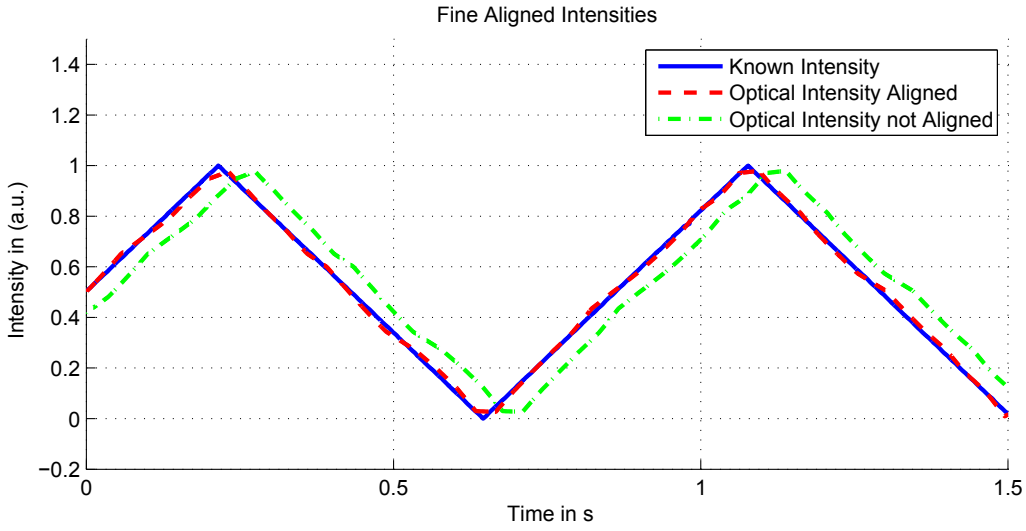


Figure 5.11: Compensated and aligned intensity sequences; known intensity (blue), optical intensity aligned (red), optical intensity not aligned (green).

Data Processing Delay

The estimation of the data processing delay is fairly simply compared to data transmission delay. Only the receiving time of the images, as well as the finishing time of the image processing task has to be logged and their difference has to be calculated. The following average processing time was found:

$$t_{\text{delay,processing}} = 3.16\,\text{ms}. \tag{5.95}$$

Total Delay

The total delay is then obtained by the summation of the individual delays:

$$t_{\text{delay}} = t_{\text{delay,processing}} + t_{\text{delay,transmission}} \tag{5.96}$$

$$= 3.16\,\text{ms} + 43.65\,\text{ms} = 46.81\,\text{ms}. \tag{5.97}$$

5.5.4. Outlier Rejection

In real technical implementations sensory data can be corrupted by anomalies, such as transmission errors or sensor failures. These sensor errors have to be detected before they are fed to the UKF, in order to inhibit corruption of the estimation. The UKF is designed according to the statistics of the sensor without sensor failures and could therefore strongly degrade the accuracy of the estimated or even lead to diverge the filter [16].
To avoid degradation due to failures, detected outliers are removed from the measurement vector. The outlier detection uses the Mahalanobis distance (MD) and is based on the method presented in [16]. The MD for a measurement vector $\mathbf{y}_k$ at time step k, is given as:

$$MD = (\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})^T \mathbf{P}^{-1}_{\tilde{\mathbf{y}}_{k|k-1}} (\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}). \tag{5.98}$$

If the UKF is perfectly modeled, the MD follows a χ^2_d distribution with a degree of freedom (DOF) $= d$. The DOF is given by the number of independent standard normal random variables, which is the length of the measurement vector.
A measurement is rejected, if the MD exceeds a chosen threshold $\chi^2_{d,\max}$:

$$MD > \chi^2_{d,\max}. \tag{5.99}$$

The threshold $\chi^2_{d,\max}$ can be calculated according to a chosen probability region. For a DOF $d = 3$ and a p-value of 99.9 the threshold is given as:

$$\chi^2_{3,\max}(99.9) = 16.27. \tag{5.100}$$

In technical implementations the UKF cannot be modeled perfectly. Therefore a practical value for the threshold should be chosen according to operational values of χ^2 [16].

5.5.5. Adaptive Measurement Noise Scaling

The following example shows how the noise characteristics of the sensors could change over time. Consider the following case: The dynamic of a moving object suddenly changes and the effect of motion blur increases. This results in a decrease of the accuracy of the monocular position tracking system. If the corresponding noise terms are not adjusted correctly, the estimations of UKF will degrade or even diverge.

For that reason an adaptive measurement noise scaling algorithm has been implemented. It is based on the Mahalanobis distance (MD), which was introduced in section 5.5.4. As already mentioned the MD follows a χ^2 distribution, if the UKF is correctly modeled [16].

When the covariance matrix of the measurements is chosen to low, the mean of the MD would be higher than the expected mean of the χ^2. On the other hand, if the measurement noise is chosen to high, the mean MD would be too low.

Therefore the covariance matrix of the measurements should be scaled, in order to match the actual MD with the expected mean of the χ^2 distribution. The mean of a χ^2 with DOF d is given as:

$$\bar{\chi}_d^2 = d. \tag{5.101}$$

An adaption algorithm is formed as:

$$c_{k+1} = c_k \cdot (1 + G \cdot (MD_k - d)), \tag{5.102}$$

$$\mathbf{R}_k = c_k \cdot \mathbf{R}_0, \tag{5.103}$$

with c_k as the covariance scaling factor, MD_k the Mahalanobis distance and $\mathbf{R}_k$ as the adapted covariance matrix at time step k. The initial covariance is represented by $\mathbf{R}_0$ and can be determined in static conditions. The gain G controls the speed of convergence.

6. Application: Low Cost 6 DOF Interaction Device

The following chapter describes the application of the framework in a low cost 6 DOF interaction device. After a brief description of the hardware components, the data and program flow is presented. Furthermore the steps and parameters for the system setup are provided.

6.1. System Description

The goal was to develop a low cost 6 DOF interaction device, which provides 3D position and orientation information in real time. The system consists of three main parts:

- Optical System (Webcam ≈ 60 €).
- Hand device containing a PCB equipped with sensors (Prototype ≈ 100 €).
- A Computer performing the sensor fusion and distributes position and orientation information.

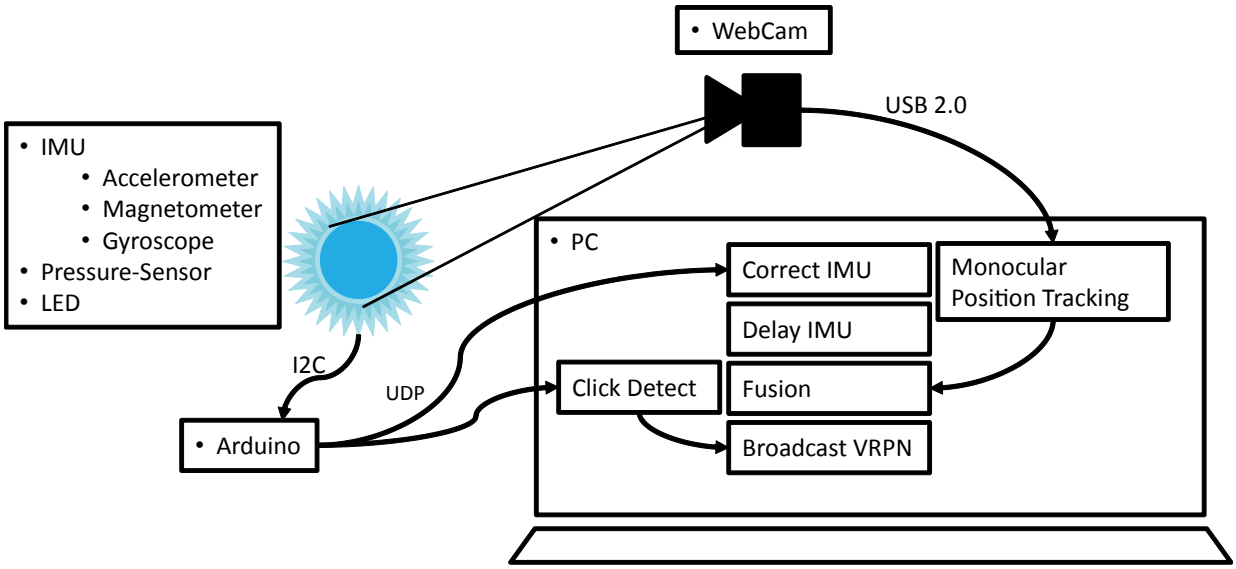


Figure 6.1: System description for a low cost 3D interaction device.

Optical System:

The optical system is mounted on top of a screen and is pointing towards the user. The webcam 'LifeCam Studio' from Microsoft is used and provides full HD images with a resolution of 1920x1080 pixels at a frame rate of 30 fps. The webcam is connected to the computer via USB 2.0.

Hand device:

The hand-device is a transparent silicon ball with a diameter of $d = 80\,\text{mm}$ (see figure 6.2). A PCB is mounted inside the ball and contains the following parts:

- Inertial measurement unit MPU9150:
  - Gyroscope; set to a sampling rate of $f = 500\,\text{Hz}$.
  - Accelerometer; set to a sampling rate of $f = 500\,\text{Hz}$.
  - Magnetometer; set to a sampling rate of $f = 125\,\text{Hz}$.
- Pressure sensor MPL3115A2.

  Note: An algorithm to extract clicks, based on the pressure was implemented but will not be covered in this thesis.
- Battery to power the device.
- LED to actively illuminate ball.
- Bluetooth Chip including a μC.

Note: The original setup intended to use Bluetooth to transmit the data from the hand device to the computer. However, due to difficulties with the Bluetooth device and to accelerate the development process, an Arduino UNO μC is used instead of the Bluetooth chip. The Arduino UNO collects the data from the sensors on the PCB by using I2C and sends the measurements via an User Datagram Protocol (UDP) to the computer.

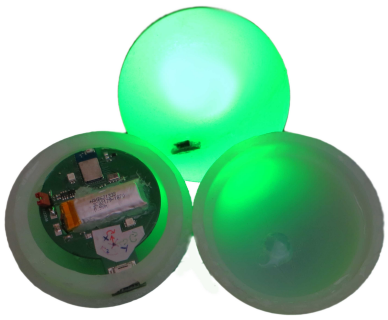


Figure 6.2: Hand device for 3D interaction.

Computer:

The computer subsequently performs the correction of deterministic errors and the delay compensation of the IMU-data, while simultaneously performing the monocular position tracking task. The sensor fusion algorithm runs with the rate of the IMU $f = 500\,\text{Hz}$ and is aided with position information from the monocular position tracking task. Position, orientation as well as click information are provided to the user via a Virtual Reality Peripheral Network (VRPN).

6.2. System Setup

6.2.1. Definition of the User Coordinate System

Since the user can define its own coordinate system the transformation from the arbitrary chosen user coordinates to the camera coordinates has to be found. Furthermore, the vectors of the gravity and magnetic field have to be determined in that coordinate system. The transformation to the user coordinate system can be found with the following procedure. Collect the positions in four different locations as illustrated in figure 6.3. The first 3 points define the X-Y-plane of the user coordinated system, where the direction of the X-coordinate is given by the vector pointing from point $\mathbf{P}_1^{\text{Cam}}$ to $\mathbf{P}_2^{\text{Cam}}$. The last point $\mathbf{P}_4^{\text{Cam}}$ defines the origin of the user coordinate system. The reader is reminded that the index 'Cam' means that the points are measured with respect to the camera coordinate system.

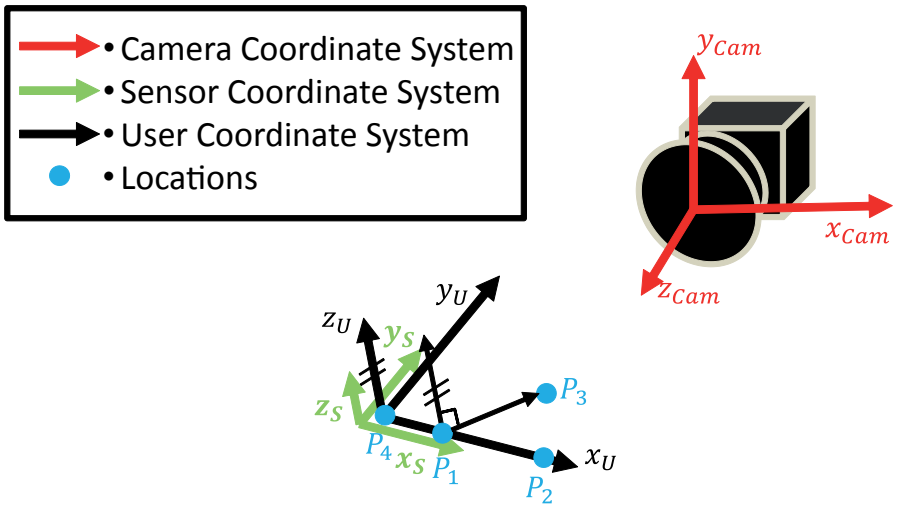


Figure 6.3: Definition of the user coordinate system. The coordinates of the camera-frame, user-frame and sensor-frame are presented in red, black and green, respectively. The points defining the user-frame are blue. Note: The axes from the user-frame are aligned with the axes from the sensor-frame.

Using the properties of the rotation matrix (see equation (2.5)), the rotation $\mathbf{R}_{\text{Cam}}^{\text{U}}$ and translation $\mathbf{p}_{\text{U}}^{\text{Cam}}$ from the camera-frame to the user-frame are given as:

$$\mathbf{R}_{\text{U}}^{\text{Cam}} = \begin{bmatrix} (\mathbf{x}_{\text{U}}^{\text{Cam}})^T \\ (\mathbf{y}_{\text{U}}^{\text{Cam}})^T \\ (\mathbf{z}_{\text{U}}^{\text{Cam}})^T \end{bmatrix}, \tag{6.1}$$

$$\mathbf{p}_{\text{Cam}}^{\text{U}} = \mathbf{P}_4^{\text{Cam}}, \tag{6.2}$$

with:

$$\mathbf{x}_{\text{U}}^{\text{Cam}} = \frac{\mathbf{P}_2^{\text{Cam}} - \mathbf{P}_1^{\text{Cam}}}{\|\mathbf{P}_2^{\text{Cam}} - \mathbf{P}_1^{\text{Cam}}\|}, \tag{6.3}$$

$$\mathbf{y}_{\text{U}}^{\text{Cam}} = \frac{\mathbf{z}_U^{\text{Cam}} \times \mathbf{x}_U}{\|\mathbf{z}_U^{\text{Cam}} \times \mathbf{x}_U^{\text{Cam}}\|}, \tag{6.4}$$

$$\mathbf{z}_{\text{U}}^{\text{Cam}} = \frac{\mathbf{x}_U^{\text{Cam}} \times \dfrac{\mathbf{P}_3^{\text{Cam}} - \mathbf{P}_1^{\text{Cam}}}{\|\mathbf{P}_3^{\text{Cam}} - \mathbf{P}_1\|}}{\|\mathbf{x}_U \times \dfrac{\mathbf{P}_3^{\text{Cam}} - \mathbf{P}_1^{\text{Cam}}}{\|\mathbf{P}_3^{\text{Cam}} - \mathbf{P}_1^{\text{Cam}}\|}\|}. \tag{6.5}$$

6.2.2. Definition of the Reference Vectors

For the estimation of the reference vectors of earth magnetic field $\mathbf{m}^U$ and gravity $\mathbf{g}^U$, the ball is placed in a stable position, such that the sensor axes are aligned with the axes from the chosen user coordinate system. This is illustrated in figure 6.3 with the sensor axes in green and the user axes in black.

The reference vectors can be calculated by averaging the magnetometer and accelerometer measurements for some seconds and are given as example for the magnetometer as follows:

$$\mathbf{m}^U = \frac{1}{N} \sum_{i=1}^{N} \mathbf{m}_k. \tag{6.6}$$

$\mathbf{m}^U$ represents the obtained reference vector in the user coordinate system and $\mathbf{m}_k$ represents the measurement at time step k. Since the coordinate system of the sensors and the user-coordinate system cannot be aligned perfectly, the vectors represent only a rough estimation.

Note: The proposed steps are usually done once for a chosen setup. They do not have to be repeated until the position or orientation of the camera is changed, or a new user coordinate system is defined.

6.2.3. Initial States and Initial Covariance

For the definition of the initial covariance matrix of the initial states, it will be assumed that the states are not correlated to each other and have the same variance in each direction such that $\sigma^2 = \sigma_x^2 = \sigma_y^2 = \sigma_z^2$.

- **Orientation:**

  The initial orientation can be roughly estimated by using the first magnetometer and accelerometer measurements.

  With the properties of the rotation matrix (represented in equation (2.5)) and the definition of the NED-frame ($\mathbf{g}^U$ points down and $\mathbf{m}^U$ lies in the north-down plane; see figure 6.4), the rotation from the user-frame to the body-frame can be given as:

$$\mathbf{R}_U^{NED} = \begin{bmatrix} \text{north} \\ \text{east} \\ \text{down} \end{bmatrix} = \begin{bmatrix} \left(\frac{\left(\frac{\mathbf{g}^U \times \mathbf{m}^U}{\|\mathbf{g}^U \times \mathbf{m}^U\|} \right) \times \mathbf{g}^U}{\left\| \left(\frac{\mathbf{g}^U \times \mathbf{m}^U}{\|\mathbf{g}^U \times \mathbf{m}^U\|} \right) \times \mathbf{g}^U \right\|} \right)^T \\ \left(\frac{\mathbf{g}^U \times \mathbf{m}^U}{\|\mathbf{g}^U \times \mathbf{m}^U\|} \right)^T \\ \left(\frac{\mathbf{g}^U}{\|\mathbf{g}^U\|} \right)^T \end{bmatrix}, \tag{6.7}$$

  where $\mathbf{g}^U$ and $\mathbf{m}^U$ are the reference vectors determined in section 6.2.2.

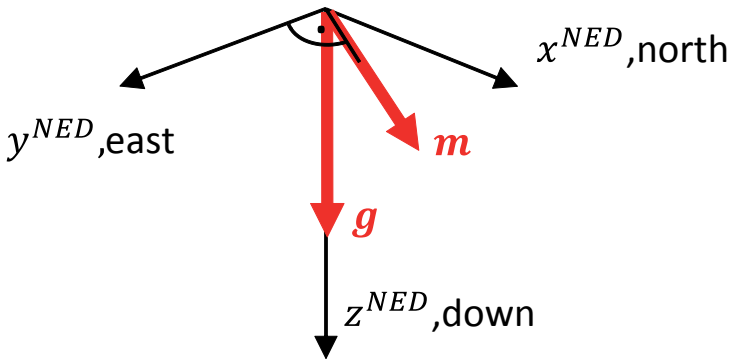


Figure 6.4: Initial estimation of the orientation.

In the same way the orientation of the body-frame with respect to the NED-frame can be calculated with the actual magnetometer **m** and accelerometer **a** measurements as follows:

$$\mathbf{R}_B^{NED} = \begin{bmatrix} \left(\frac{\left(\frac{\mathbf{a} \times \mathbf{m}}{\|\mathbf{a} \times \mathbf{m}\|} \right) \times \mathbf{a}}{\left\| \left(\frac{\mathbf{a} \times \mathbf{m}}{\|\mathbf{a} \times \mathbf{m}\|} \right) \times \mathbf{a} \right\|} \right)^T \\ \left(\frac{\mathbf{a} \times \mathbf{m}}{\|\mathbf{a} \times \mathbf{m}\|} \right)^T \\ \left(\frac{\mathbf{a}}{\|\mathbf{a}\|} \right)^T \end{bmatrix}. \tag{6.8}$$

The initial orientation can be given by combining both rotations:

$$\mathbf{R}_U^B = \left(\mathbf{R}_B^{NED} \right)^T \cdot \mathbf{R}_U^{NED}. \tag{6.9}$$

Since the orientation is just a rough estimation, a large initial variance of $\sigma^2 = (\frac{15\pi}{180} \text{ rad})^2$ was chosen.

Note: This procedure represents a runtime initialization and has to be performed every time the tracking is started.

- **Position:**

  The initial position is set to the first received position. In order to represent the large uncertainty, a variance of $\sigma^2 = (15 \times 10^{-3} \text{ m})^2$ was chosen.

- **Velocity:**

  Since no information of the velocity is known beforehand, the initial value was set to zero and the following initial variance $\sigma^2 = (60 \times 10^{-3} \frac{\text{m}}{\text{s}})^2$ was chosen.

- **Biases:**

  All biases are initially set to zero and the following variances are chosen; magnetometer: $\sigma_m^2 = (0.03\ a.u.)^2$, accelerometer: $\sigma_a^2 = (1 \times 10^{-3} \frac{\text{m}}{\text{s}^2})^2$, gyroscope: $\sigma_\omega^2 = (1 \times 10^{-4} \frac{\text{rad}}{\text{s}})^2$.

- **Reference Vectors:**

  Reference vectors of the magnetic field and gravity are initialized with the values from section 6.2.1.

- **Lever Arm:**

  The lever arm is initially set according to the design drawings and with an initial variance of $\sigma^2 = (0.5 \times 10^{-3}\ \mathrm{m})^2$.

6.2.4. Process Noise / Measurement Noise

The process noise will be set according to the values obtained in section 3.2.4. The variance of the noise, which drives the RW of the magnetic field, was set to $\sigma^2_{m_{RW}} = (1 \times 10^{-3}\ a.u.)^2$.
The measurement noise for the magnetometer is set according to the values in section 3.2.4. The following covariance matrix for the optical measurement system was found:

$$\mathbf{R}_{p,Cam} = \begin{bmatrix} (0.2 \times 10^{-3})^2 & 0 & 0 \\ 0 & (0.2 \times 10^{-3})^2 & 0 \\ 0 & 0 & (1.8 \times 10^{-3})^2 \end{bmatrix} \mathrm{m}^2. \tag{6.10}$$

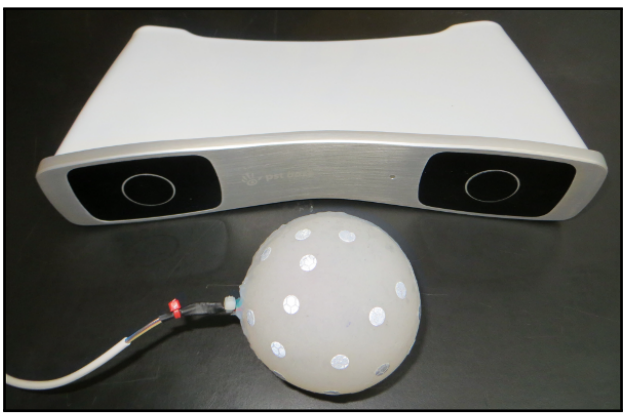
It should be noted that this covariance matrix $\mathbf{R}_{p,Cam}$ is expressed in camera-coordinates.
The given covariance matrices for the measurements represent the initial covariance matrices and will be scaled by the adaptive measurement noise scaling algorithm pretested in section 5.5.5.

7. Demonstration and Results

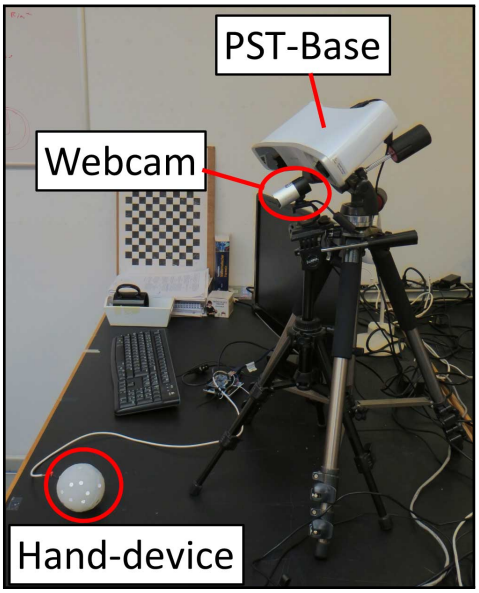
In this chapter the system described in chapter 6 will be demonstrated and its performance will be evaluated. After presenting the testing setup and the testing procedure, the results for the 6 DOF interaction device are finally presented.

7.1. Testing Setup

For the validation the ground truth was obtained using the optical tracking system PST Base from PS-Tech. The PST-Base provides position information with an accuracy of $RMSE < 0.5\,\text{mm}$ and orientation with $RMSE < 1\,\text{deg}$ with an update rate of 120 Hz [62]. In order to track the hand device, several retro-reflective markers were attached to the hand-device (see figure 7.1a). The position and orientation of the webcam and the PST-Base were roughly aligned in order to achieve a similar tracking volume. The testing setup is shown in figure 7.1b.



(a)



(b)

Figure 7.1: (a) PST-Base and hand-device; (b) System setup.

7.2. Testing Procedure

The following steps were performed during the testing procedure:

- Calibration of accelerometer, gyroscope and magnetometer.
- Collection of 4 points to define user-coordinate system.
  Note: These collected points are also used to align the PST-Base coordinate system with the user-coordinate system.
- Running the test sequence.

7.3. Results

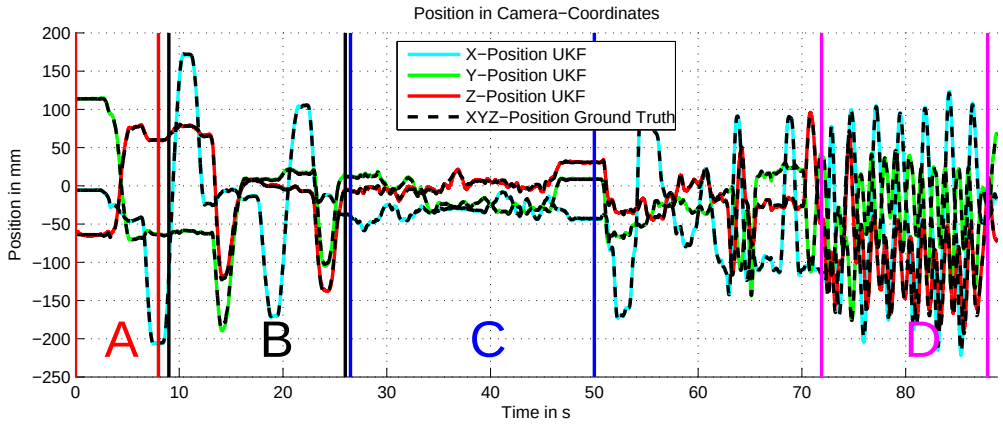
For testing, one complex sequence was recorded containing the following tasks:

A: Initialization phase,
B: Translation,
C: Rotation,
D: Combined fast rotations and translations.

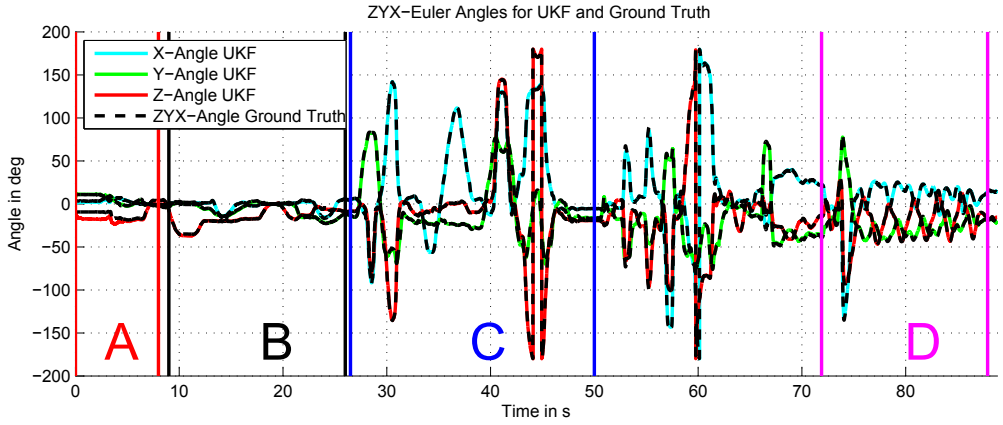
The complete sequence with the marked tasks (A-D) is shown in figure 7.2. It should be noted, that all positions are expressed in camera-coordinates. In order to enable the plotting of the position in the same axes, the z-position was shifted 600 mm in the negative direction. Furthermore, the orientations are given as ZYX-Euler angles, which represent the rotation from the NED-frame to the body-frame. The position error is given by the euclidean distance. The orientation error between the ground truth and the estimated orientation is given as magnitude of the rotation vector expressed in degrees. The RMSE for position and orientation is given in table 7.1.

Table 7.1: Root-Mean-Squared-Error for complete sequence A-D.

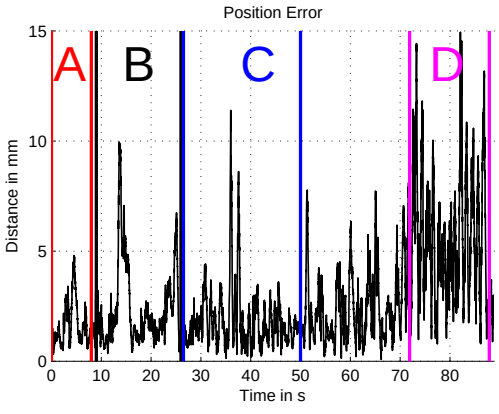
	RMSE			
	Camera Coordinates			Rotation-Angle
	mm			deg
Part A-D	1.49	1.33	2.94	2.06
Part B-D	1.55	1.37	3.05	1.31



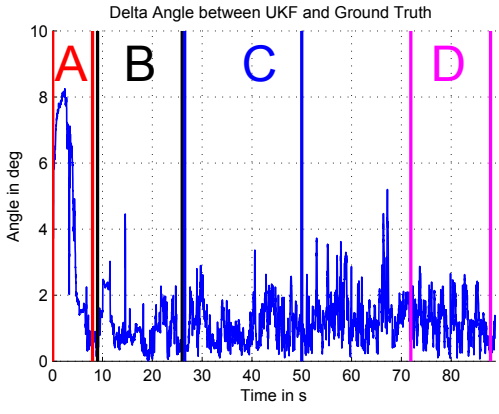
(a)



(b)



(c)



(d)

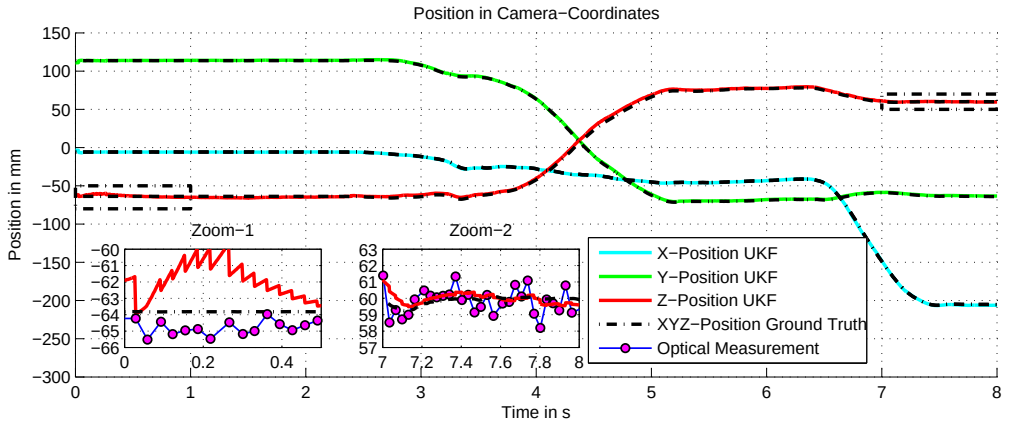
Figure 7.2: Complete sequence with marked tasks: (a) XYZ Position in mm; (b) ZYX-Euler angles in deg; (c) Position error in mm; (d) Orientation error in deg.

7.3.1. Initialization Phase - A

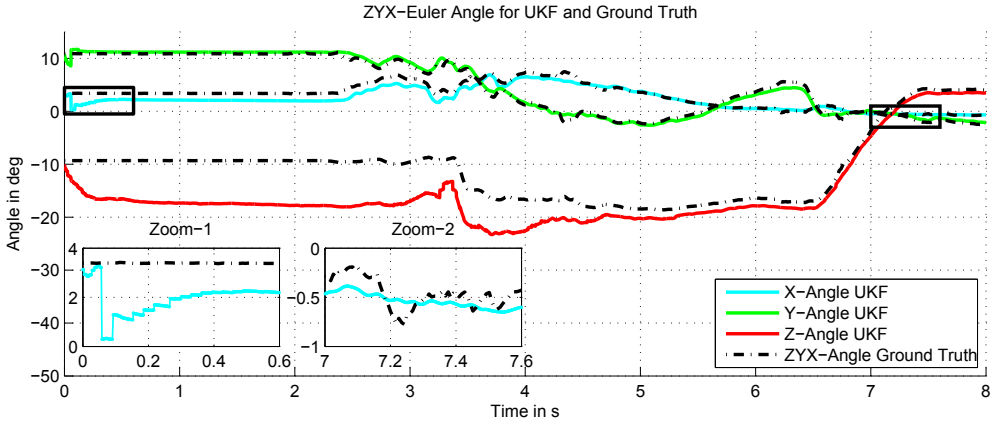
The results of the initialization phase are shown in figure 7.3. At the beginning the orientation of the object is not known exactly, therefore the acceleration due to gravity cannot be compensated correctly and causes the estimated position to diverge (see figure 7.3a Zoom-1). Every time a new position measurement is received, the filter corrects the estimation of the orientation and the gravity vector. After about 0.3 seconds the orientation seems to be correctly estimated and the position estimation shows good results (Zoom-1 / Zoom-2).
However, due to the singularity of the gravity vector around the z-axis, the rotation around the z-axis could not be corrected and is still wrong (see figure 7.3b). With the use of the gravity vector the rotation around the x and y-axis is correctly estimated. When the object starts to move ($t \approx 4 \ldots 5\,s$), the unknown orientation around the z-axis can be corrected by aligning the object's displacement with the sensed acceleration. After initialization the estimated position and orientation show a correct and smooth result (see figure 7.3a and figure 7.3b Zoom-2).
In table 7.2 the RMSE for the orientation and position are given. As expected, the estimated orientation is strongly error affected. However, the estimated position shows a very good result for the x and y coordinates. Because the monocular position tracking system shows a strong error sensitivity for depth information, the RMSE of the z-coordinate is larger. Simulations showed that an error of only one pixel for the estimated circle radius results in a depth error of $\approx 9\,mm$. For the simulation the object was placed $\approx 0.8\,m$ from the camera and at the center of the camera axis.

Table 7.2: Root-Mean-Squared-Error for part A.

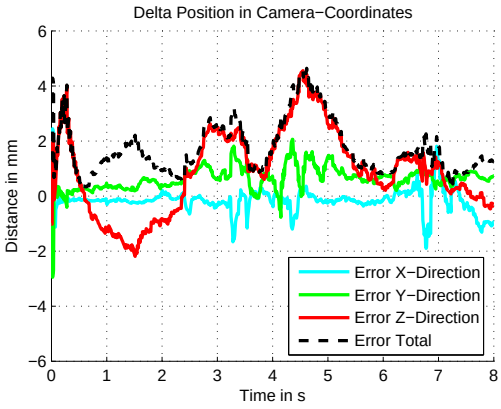
	RMSE			
	Camera Coordinates			Rotation-Angle
	mm			deg
Part A	0.46	0.76	1.76	5.49



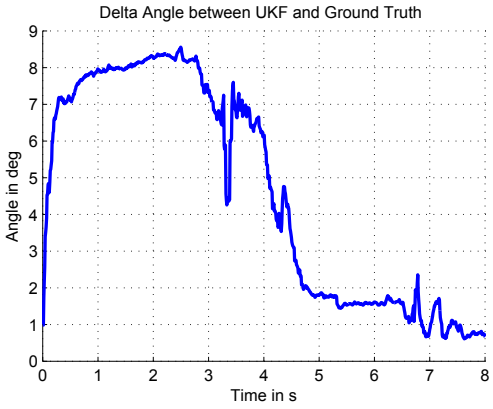
(a)



(b)



(c)



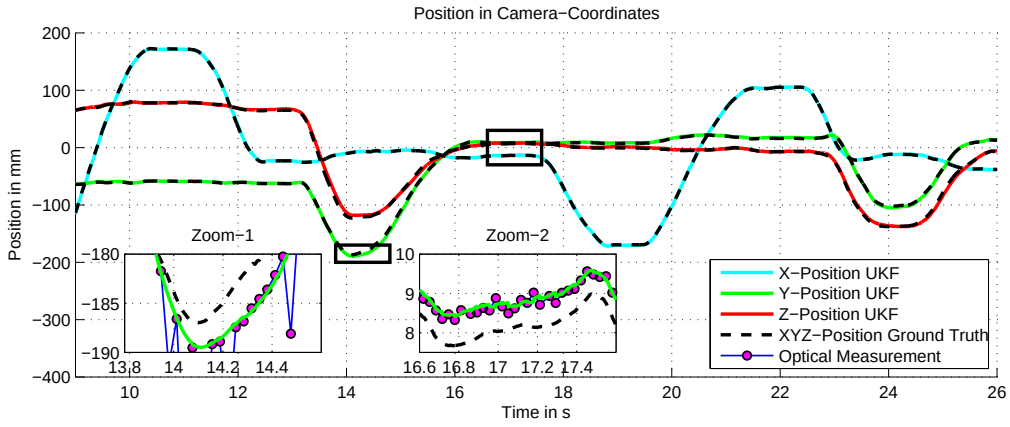
(d)

Figure 7.3: Results for initialization phase: (a) XYZ Position in mm; (b) ZYX-Euler angles in deg; (c) Position error in mm; (d) Orientation error in deg.

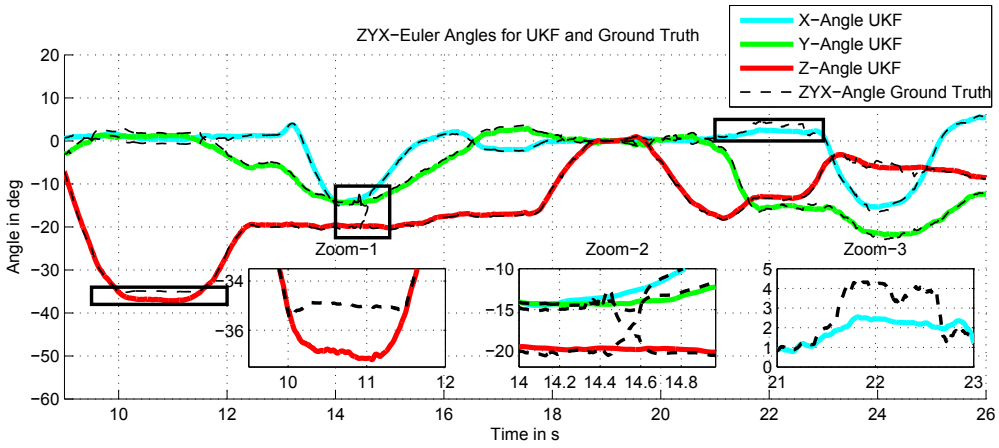
7.3.2. Translation - B

The results for the translation task are given in figure 7.4, figure 7.5 and in table 7.3. In general, the position estimation shows a very accurate result. However, the estimation contains two sections with significant errors. A zoomed graph of the first error section for $t = 14 \ldots 16\,\mathrm{s}$ is presented in figure 7.4a Zoom-1 and in figure 7.5. Both show a significant deviation from the ground truth. However, the graph shows a smooth estimation and the position calculation based on the inertial information seems to agree with the obtained optical measurements. A similar behavior can be observed in figure 7.4a Zoom-2.
The following sources could be candidates for this error: Uncompensated lens distortion, incorrect camera matrix, wrong circle fit, out of focus errors or ground truth error. However, further research has to be done to determine the source of these errors. The offset of the position estimation deviates slowly and in a smooth manner, hence the effects on user interaction can be assumed to be neglectable. Furthermore, a successful outlier rejection of faulty position measurements is demonstrated in figure 7.5.
The errors of the estimated orientations are mostly under 1 deg, however at some points the error suddenly significantly increased (see figure 7.4b Zoom-1, Zoom-2 and Zoom-3). It seems that the ground truth jumps between two possible orientations. After a discussion with the developers of the PST-Base, the following conclusion was found: Due to the small size of the tracked object, only few markers ($\approx 3 - 4$) are visible to the tracking unit. If in addition one or more of these markers are (partly) occluded, the tracker could provide faulty orientation information with the described behavior.
In addition, the estimated orientation shows a smoother and less noisy behavior than the ground truth. For that reason, it is assumed that the estimated orientation outperforms the PST-Base for this task.

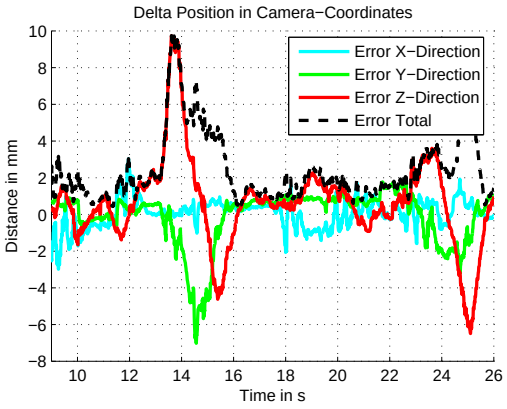
The RMSE for the orientation and position represents a very good result, but could be further improved, if the above mentioned effects are compensated.



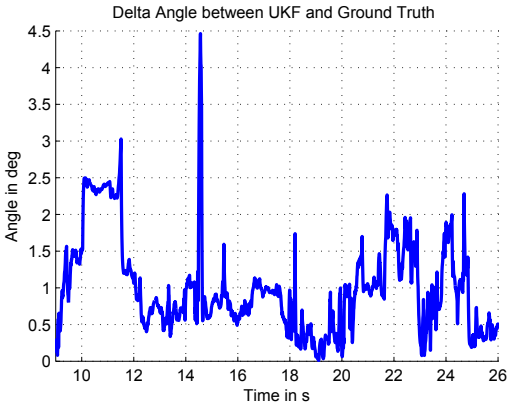
(a)



(b)



(c)



(d)

Figure 7.4: Results for translation task: (a) XYZ Position in mm; (b) ZYX-Euler angles in deg; (c) Position error in mm; (d) Orientation error in deg.

Table 7.3: Root-Mean-Squared-Error for part B.

	RMSE			
	Camera Coordinates			Rotation-Angle
	mm			deg
Part A	0.76	1.52	2.48	1.20

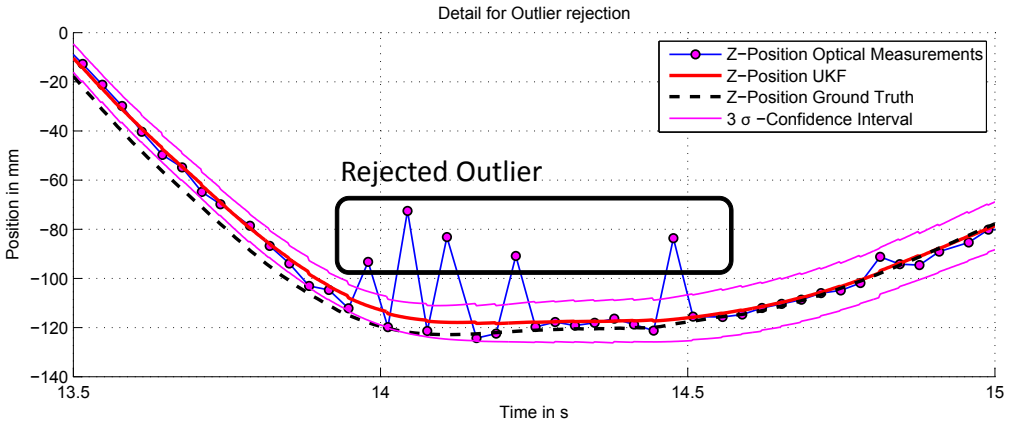


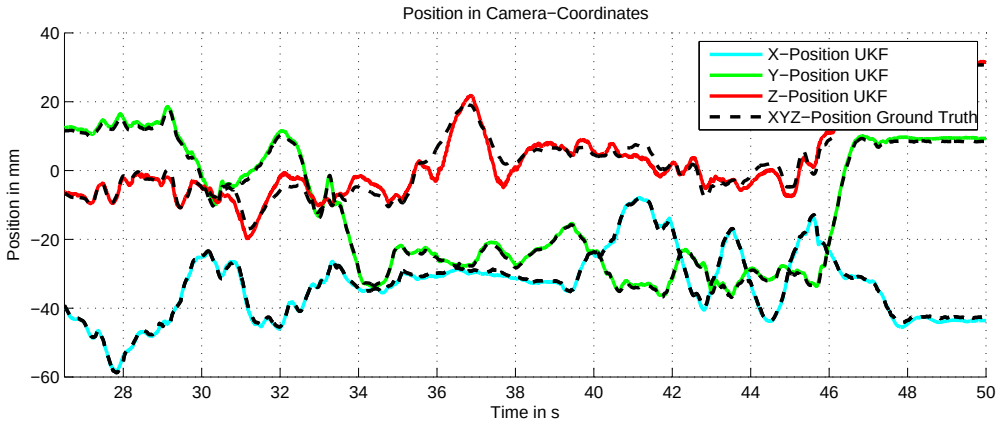
Figure 7.5: Zoomed detail for Z-position: Showing successful outlier rejection.

7.3.3. Rotation - C

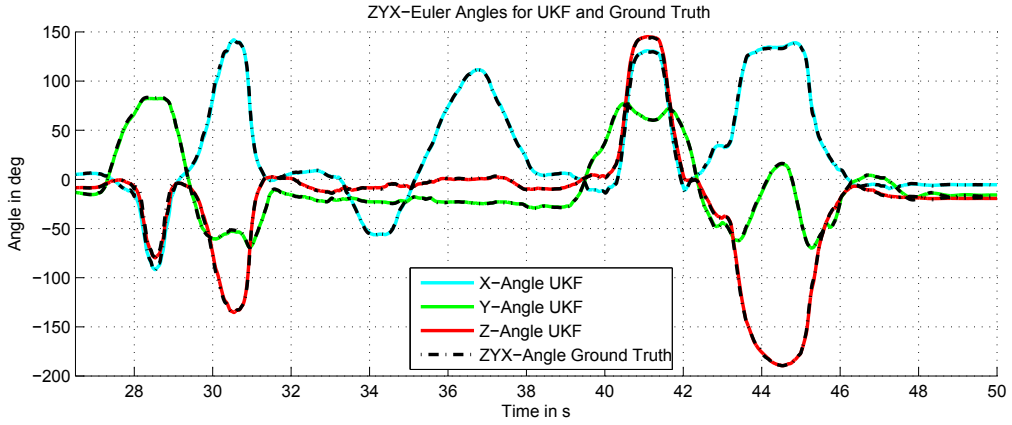
The results of the rotation task are given in figure 7.6. With the exception of two sections, the estimated position is very accurate. The section showing the largest position error in the z-direction is shown in figure 7.7. The position error is caused by faulty monocular position information in combination with a failed outlier rejection.

A strong correction of the estimated position is performed on every faulty position information. This results in a noisy estimation, which is an indicator that the monocular position information and the calculation of the position using inertial data, do not match. The error affected position slowly diverges from the correct position and as a result pushes the estimation in the direction of the faulty measurements. Every new obtained position is statistically possible and therefore the outlier rejection failed. An incorrect RANSAC circle fit is assumed to be the cause of the faulty position measurements.

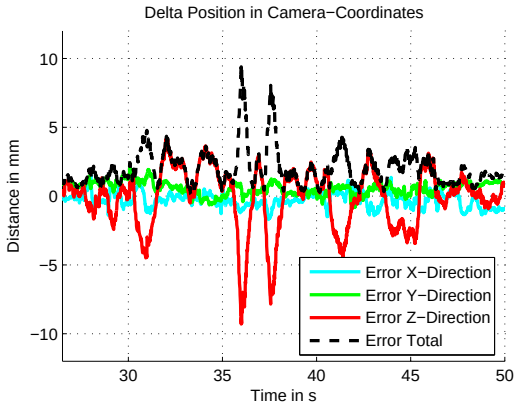
However, even with these uncompensated outliers, a decent RMSE for the position was obtained.



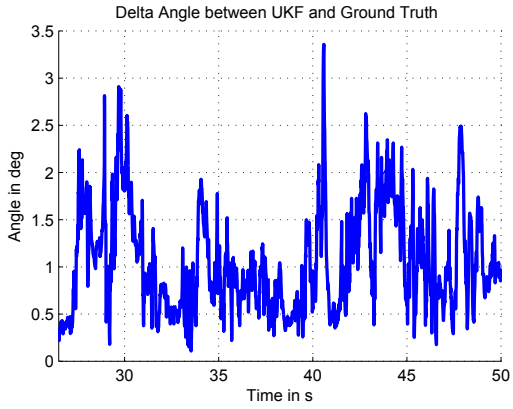
(a)



(b)



(c)



(d)

Figure 7.6: Results for rotation task: (a) XYZ Position in mm; (b) ZYX-Euler angles in deg; (c) Position error in mm; (d) Orientation error in deg.

Despite the assumption that the orientation error will increase with faster rotations, the lowest RMSE of all tasks was achieved. The RMSE values are given in table 7.4.

Table 7.4: Root-Mean-Squared-Error for part C.

	RMSE			
	Camera Coordinates			Rotation-Angle
	mm			deg
Part C	0.60	0.74	2.20	1.19

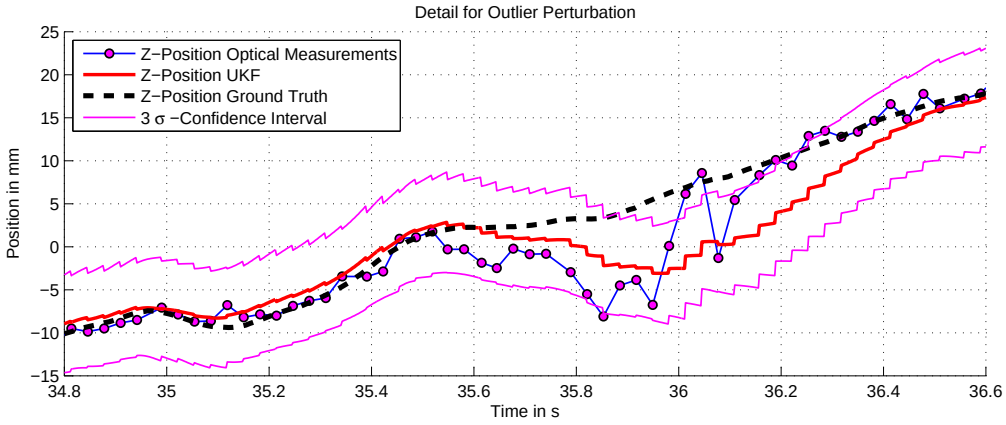


Figure 7.7: Perturbated position due to outliers.

7.3.4. Combined fast Rotation and Translation - D

The results of task D are presented in figure 7.10. It clearly shows a significant higher error for the position. It can be seen in figure 7.8, that the monocular position measurements are strongly affected by errors and therefore decreased the performance of the estimation. Yet, even with many position measurements, which are affected by errors, a smooth and accurate estimation is achieved. So far motion blur is assumed to be the reason for the error affected position measurements. The RMSE for task D is presented in table 7.5.

The RMSE for the orientation estimation shows a slightly poorer performance than the other tasks. Two possible reasons are considered. Either the results for the orientation of the ground truth is affected by the fast movement or/and the lack of correct position information degrades the estimation. It is still assumed that the obtained accuracy is good enough for user interactions.

In addition, the scale factor of the adaptive noise scaling algorithm of the optical position measurement noise is shown in figure 7.9. It clearly demonstrates the successful adjustment for the increased measurement noise in task-D.

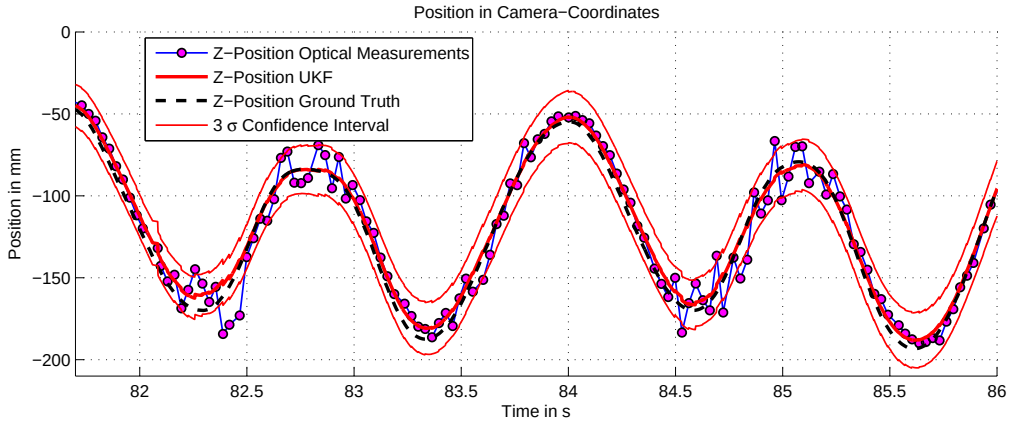


Figure 7.8: Position detail: Strong perturbed data of the monocular position tracking system.

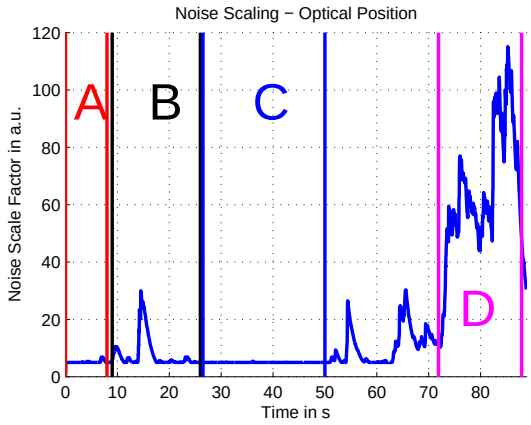
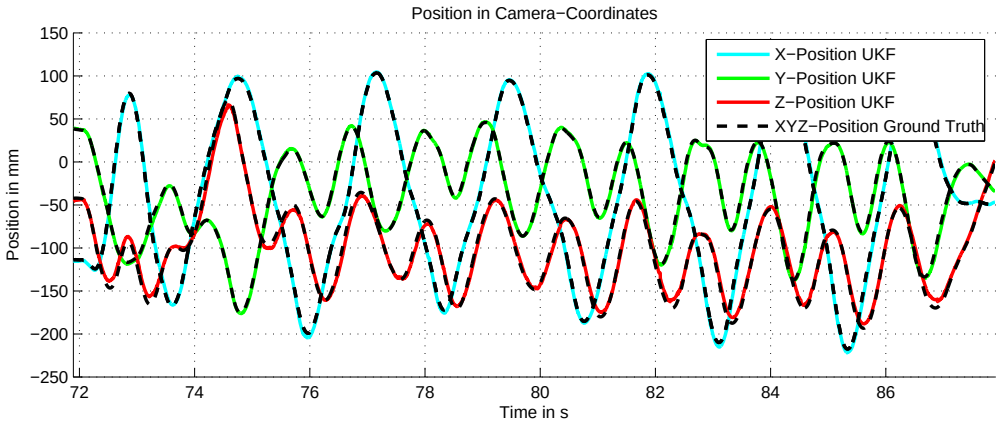


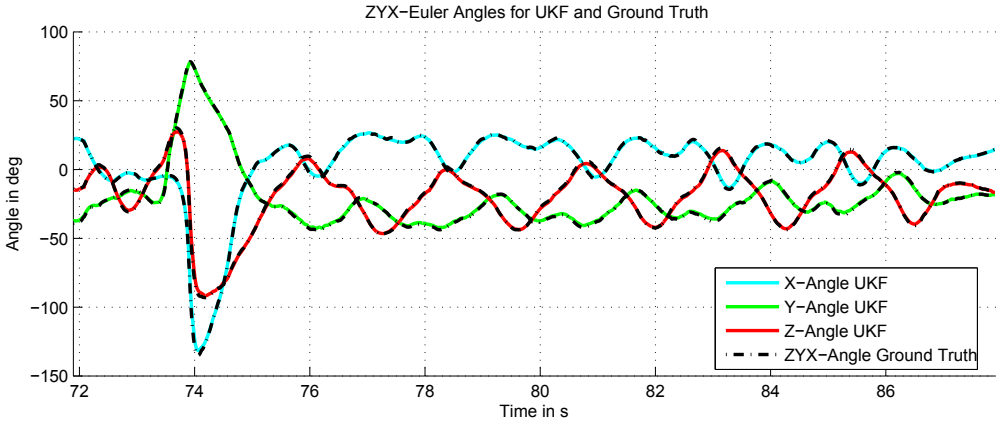
Figure 7.9: Optical position noise scaling.

Table 7.5: Root-Mean-Squared-Error for part D.

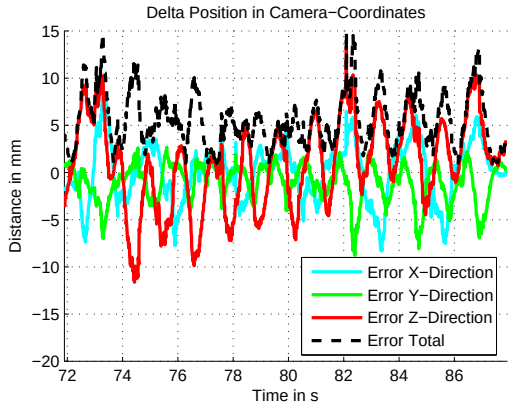
	RMSE			
	Camera Coordinates			Rotation-Angle
	mm			deg
Part D	3.05	2.29	4.83	1.30



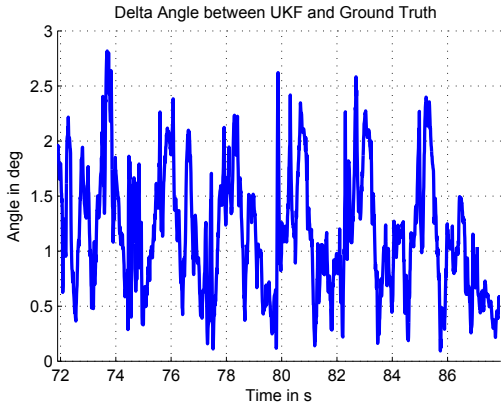
(a)



(b)



(c)



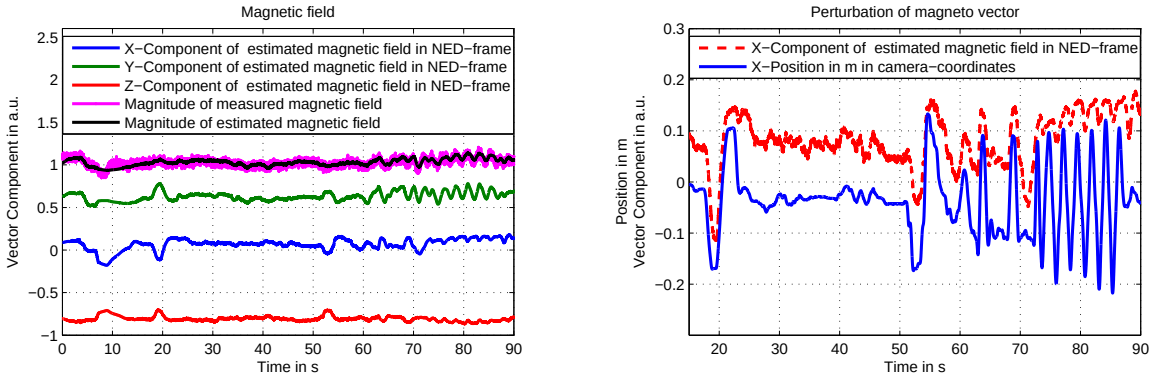
(d)

Figure 7.10: Results for combined rotation and translation task: (a) XYZ Position in mm; (b) ZYX-Euler angles in deg; (c) Position error in mm; (d) Orientation error in deg.

7.3.5. Influence on Magnetic Field

During the evaluation of the filter, the following effect was observed. If no perturbation of the environment is present, the vector of the magnetic field, which is expressed in the NED-frame or user-frame, should be constant. However, the estimated magneto vector shows strong variations. In order to capture the whole dynamic of the changing magnetic field, the RW noise was increased by a factor of 10.

The results of the estimated magneto-vector expressed in the NED-frame are shown in figure 7.11.



(a) (b)

Figure 7.11: Perturbation of the magnetic field: (a) Estimation of the magnetic field in NED-frame; (b) Comparison of the x-component of the magnetic field with the x-component of the position in camera-coordinates.

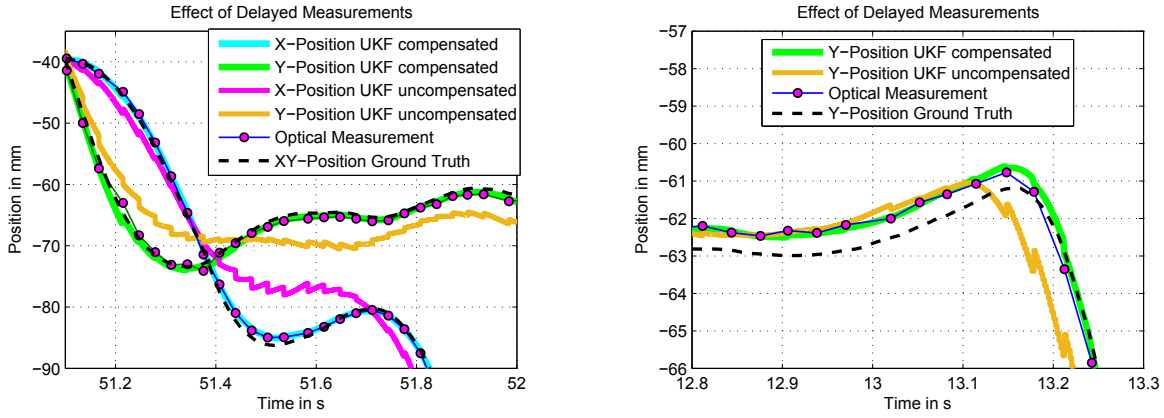
The magnitude of the estimated magneto vector fits well the measured magneto vector and therefore it is assumed that the estimation is correct.

Figure 7.11b presents the x-component of the magnetic field and the x-component of the estimated position in camera-coordinates. The graphs show strong correlation. This indicates that an external magnetic source perturbs the natural magnetic field. It should be noted, that for testing all objects, which could perturb the magnetic field, were removed from the close environment. However, it seems that even in a carefully chosen environment the perturbation is significant. For that reason, the measurement of the magnetometer is considered of limited use to improve the orientation estimation.

Since the position and the magnetic field show strong correlation, an interesting approach would be the following. Estimate the location of the source which is perturbing the magnetic field. By using the signal strength and the location of the source, distance information with respect to the extern magnetic source could be provided.

7.3.6. Influence of Delayed Measurements

As already mentioned in section 5.5.3, the delay of the optical position measurement system could significantly degrade the estimation result. Figure 7.12 provides two examples for the comparison between compensated and uncompensated estimations.



(a) (b)

Figure 7.12: Influence of time delays: (a) Example 1: Position in mm; (b) Example 2: Position in mm.

Figure 7.12b demonstrates the effect of delayed measurements on the estimation. This effect can be explained as follows:
The object changes the direction of its movement at time $t = 13.15\,\mathrm{s}$. The inertial measurement unit measures a negative x-acceleration and therefore calculates the position in the negative direction (see $t = 13.1 \ldots 13.15\,\mathrm{s}$). However, the according position measurement is received delayed and still represents the position before the turn. Therefore the filter corrects the position again in the positive direction. This cycle continues and the discrepancy between the inertial and optical measurements leads to a very noisy and inaccurate estimation. In some cases, this could lead to divergence of the filter. This bad performance of the uncompensated filter is also

reflected in the RMSE of the position and is given in table 7.6. The accuracy of the orientation is also affected, but still represents a decent result.

Table 7.6: Root-Mean-Squared-Error for part B-D for uncompensated timing delays.

	RMSE			
	Camera Coordinates			Rotation-Angle
	mm			deg
Part B-D	9.03	5.83	6.67	2.66

8. Conclusion and Future Work

The application of the proposed sensor fusion framework to provide a low cost 3D interaction device shows very good results. For the presented complex sequence, a RMSE $\approx 1.5\,\text{mm}$ for the x and y coordinates and a RMSE $\approx 3\,\text{mm}$ for the z-coordinate was achieved. The worse performance of the z-direction is caused by a larger error sensitivity for depth information of the monocular position tracking system.

For the estimation of the orientation, a RMSE $\approx 1.3\,\text{deg}$ was achieved. However, some results pointed out that the presented application outperformed the accuracy of the ground truth. It is assumed that the accuracy and smoothness of the orientation estimation satisfies the requirements for user interaction applications. Even though a pretty good result for the orientation was achieved, the performance should be further improved. The calculated acceleration is strongly depending on the orientation and therefore a better estimation of the orientation also improves the position estimation.

A large part of the RMSE for the position is caused by an offset in the provided positions. Since the estimated positions are smooth and still capture the complete dynamic of the movement, it is assumed that this slowly changing offset is hardly noticeable by the user.

The outlier rejection worked properly, for obvious outliers. Nevertheless, if measurements degrade slowly in one directions, the outlier rejection could fail. Further research has to be conducted in order to identify and to proper handle these faulty position measurements. However, this kind of outliers rarely occurred and only showed a short presence.
One obvious approach to handle those undetected outliers is to improve the accuracy of the IMU. This would yield a smaller estimated covariance for the position and therefore minimizes the range of accepted faulty measurements.
The nonlinearity of the inertial sensor presents the largest error source. For that reason a more precise calibration of the sensors and its nonlinearity would strongly improve the estimation results. Unfortunately, calibrations are time consuming and expensive. An online estimation of the nonlinearity parameters could be considered.

In the presence of changing measurement noise, such as in task D, the adaptive noise scaling method worked successfully. The proposed adaptive measurement scaling algorithm is only based on simple MD matching. Therefore a more profound method, such as simultaneously measurement noise $\mathbf{R}$ and process noise $\mathbf{Q}$ adaption algorithms, could further improve the result.

Motion blur is assumed to be the cause for the degraded performance in fast moving scenes. Since the position and velocity of the object are estimated, a simple motion deblurring algorithm could be applied and could strongly improve the performance for fast moving objects.

The results for uncompensated delays of the measurements showed that even a delay of $\approx 50\,\text{ms}$ strongly degrades the performance. In this thesis a simple approach for compensation was chosen. It results in an overall delay of $\approx 50\,\text{ms}$. An algorithm to incooperate delayed measurements, without delaying the estimation, is proposed in [31]. For further improvements this approach should be considered in order to provide 6-DOF tracking in delay sensitive applications.

The estimated magnetic field showed very strong variations and therefore pointed out that the magnetometer measurements hardly can be used for precise position and orientation estimation indoors. For the developed interaction device, it could be considered to even remove the magnetometer.

Calibration procedures are time consuming and further research and testing should be done to include the calibration of the inertial sensors in the estimation framework, similar to the online estimation of the bias and reference vectors. However, it should be noted that the estimation can get unstable and error sensitive when more parameters are added. Especially outliers could degrade the result drastically.

In order to develop a product ready for the market, further research and testing should be conducted to clarify the causes of outliers and to develop methods to properly handle them.

There are still many ways to improve the existing framework and application. Due to limited time and because it exceeds the scope of this thesis, it will no be covered here. The interested reader is invited to contact me for further questions.

Bibliography

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, 2004.

[2] Digi-Capital. (April 2015). Augmented/virtual reality to hit $150 billion disrupting mobile by 2020, Digi-Capital, [Online]. Available: http://www.digi-capital.com/news/2015/04/augmentedvirtual-reality-to-hit-150-billion-disrupting-mobile-by-2020/#.ViTQ7yvLI2u (visited on 10/19/2015).

[3] M. Arikawa, S. Konomi, and K. Ohnishi, "Navitime: supporting pedestrian navigation in the real world," *IEEE Pervasive Computing*, vol. 6, no. 3, pp. 21–29, July 2007.

[4] S. Li, A. Zhan, X. Wu, *et al.*, "ERN: emergence rescue navigation with wireless sensor networks," IEEE, 2009, pp. 361–368, ISBN: 978-1-4244-5788-5.

[5] C. Fischer and H. Gellersen, "Location and navigation support for emergency responders: a survey," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 38–47, 2010.

[6] M. Angermann, M. Khider, and P. Robertson, "Towards operational systems for continuous navigation of rescue teams," in *Position, Location and Navigation Symposium, 2008 IEEE/ION*, IEEE, 2008, pp. 153–158.

[7] R. Bill, C. Cap, M. Kofahl, *et al.*, "Indoor and outdoor positioning in mobile environments - a review and some investigations on wlan-positioning," *Geographic Information Sciences*, vol. 10, no. 2, 2004.

[8] I. Skog and P. Handel, "In-car positioning and navigation technologies-a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 4–21, March 2009.

[9] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

[10] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," vol. 15, no. 3, pp. 1281–1293, 2013.

[11] G. Deak, K. Curran, and J. Condell, "A survey of active and passive indoor localisation systems," *Computer Communications*, vol. 35, no. 16, pp. 1939–1954, September 2012.

[12] D. Maier, A. Hornung, and M. Bennewitz, "Real-time navigation in 3d environments based on depth camera data," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, IEEE, 2012, pp. 692–697.

[13] A. D. King, "Inertial navigation-forty years of evolution," *GEC review*, vol. 13, no. 3, pp. 140–149, 1998.

[14] J. Wendel, *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation*. Oldenbourg Verlag, 2007, 352 pp., ISBN: 9783486581607.

[15] D. J. Biezad, *Integrated Navigation and Guidance Systems*. AIAA, 1999, 262 pp., ISBN: 9781600860713.

[16] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley & Sons, March 5, 2007, 553 pp., ISBN: 9780470099711.

[17] J. Bortz, "A new mathematical formulation for strapdown inertial navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-7, no. 1, pp. 61–66, January 1971.

[18] I. Inc., "MPU-9150_datasheet_v4_3.pdf." [Online]. Available: http://store.invensense.com/datasheets/invensense/MPU-9150_DataSheet_V4%203.pdf (visited on 06/20/2015).

[19] W. Khoder and B. Jida, "A quaternion scaled unscented kalman estimator for inertial navigation states determination using INS/GPS/magnetometer fusion," *Journal of Sensor Technology*, vol. 04, no. 2, pp. 101–117, 2014.

[20] C. Jekeli, *Inertial Navigation Systems with Geodetic Applications*. Walter de Gruyter, January 1, 2001, 370 pp., ISBN: 9783110159035.

[21] P. G. Savage, "Strapdown inertial navigation integration algorithm design part 2: velocity and position algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 208–221, 1998.

[22] F. Gustafsson, *Statistical Sensor Fusion*. Studentlitteratur, 2010.

[23] O. J. Woodman, "An introduction to inertial navigation," *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, vol. 14, p. 15, 2007.

[24] D. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology, 2nd Edition*. IET, January 1, 2004, 578 pp., ISBN: 9780863413582.

[25] "IEEE standard for inertial sensor terminology," *IEEE Std 528-2001*, 2001.

[26] I. Frosio, F. Pedersini, and N. Alberto Borghese, "Autocalibration of MEMS accelerometers," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 2034–2041, June 2009.

[27] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003, 676 pp., ISBN: 9780521540513.

[28] T. Ozyagcilar, "Calibrating an eCompass in the presence of hard and soft-iron interference," *Freescale Semiconductor Ltd*, 2012.

[29] W. Gander and J. Hrebicek, *Solving Problems in Scientific Computing Using Maple and MATLAB®*. Springer Science & Business Media, June 27, 2011, 479 pp., ISBN: 9783642188732.

[30] I. D. Coope, "Circle fitting by linear and nonlinear least squares," *Journal of Optimization Theory and Applications*, vol. 76, no. 2, pp. 381–388, February 1993.

[31] R. Van Der Merwe, "Sigma-point kalman filters for probabilistic inference in dynamic state-space models," Doctoral dissertation, Oregon Health & Science University, 2004.

[32] N. El-Sheimy, H. Hou, and X. Niu, "Analysis and modeling of inertial sensors using allan variance," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 1, pp. 140–149, January 2008.

[33] M. El-Diasty and S. Pagiatakis, "Calibration and stochastic modelling of inertial navigation sensor errors," *Journal of Global Positioning Systems*, vol. 7, no. 2, pp. 170–182, 2008.

[34] S. Nassar, *Improving the inertial navigation system (INS) error model for INS and INS/DGPS applications*. University of Calgary, Department of Geomatics Engineering, 2003.

[35] *IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros*. 1998, ISBN: 9781559379618 1559379618.

[36] R. C. Gonzalez, *Digital Image Processing*. Pearson Education India, 2009.

[37] A. R. Smith, "Color gamut transform pairs," in *ACM Siggraph Computer Graphics*, vol. 12, ACM, 1978, pp. 12–19.

[38] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, November 1986.

[39] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.

[40] C. W. Misner, K. S. Thorne, and J. A. Wheeler, *Gravitation*. Macmillan, 1973.

[41] F. Tarsha-Kurdi, T. Landes, and P. Grussenmeyer, "Hough-transform and extended RANSAC algorithms for automatic detection of 3d building roof planes from lidar data," presented at the ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, vol. XXXVI, September 2007, pp. 407–412.

[42] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with apphcatlons to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[43] J. O'Rourke, *Computational Geometry in C*. Cambridge University Press, October 13, 1998, 396 pp., ISBN: 9780521649766.

[44] W. Elmenreich, "Sensor fusion in time-triggered systems," Doctoral dissertation, 2002.

[45] E. Bossé, J. Roy, and D. Grenier, "Data fusion concepts applied to a suite of dissimilar sensors," in *Canadian Conference on Electrical and Computer Engineering, 1996*, vol. 2, 1996, pp. 692–695.

[46] A. Abdelgawad and M. Bayoumi, "Data fusion in WSN," in *Resource-Aware Data Fusion Algorithms for Wireless Sensor Networks*, Springer, 2012, pp. 17–35.

[47] Reinstein Michal, "Use of adaptive filtering methods in inertial navigation systems," Doctoral dissertation, Czech technical university in prague, Prague, 2010.

[48] S. Särkkä, "Recursive bayesian inference on stochastic differential equations," Doctoral dissertation, Helsinki University of Technology, Espoo, 2006.

[49] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[50] S. Kay, *Fundamentals of statistical signal processing: Estimation theory*. Prentice-Hall PTR, April 5, 2010, ISBN: 0133457117.

[51] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, March 2004.

[52] S. Julier, "The scaled unscented transformation," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 6, 2002, pp. 4555–4559.

[53] B. P. Gibbs, *Advanced Kalman Filtering, Least-Squares and Modeling: A Practical Handbook*. John Wiley & Sons, March 29, 2011, 692 pp., ISBN: 9781118003169.

[54] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons, June 19, 2006, 555 pp., ISBN: 9780470045336.

[55] J.-S. Jang, "ANFIS : adaptive-ne twork-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[56] W. Abdel-Hamid, "Accuracy enhancement of integrated MEMS-IMU/GPS systems for land vehicular navigation applications," Doctoral dissertation, University of Calgary, Department of Geomatics Engineering, 2005.

[57] A. M. Hasan, K. Samsudin, and A. R. Ramli, "GPS/INS integration based on dynamic ANFIS network," *International Journal of Control and Automation, SERSC*, vol. 6, no. 3, pp. 1–22, 2012.

[58] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, IEEE, 2000, pp. 153–158.

[59] A. L. Ruina and R. Pratap, *Introduction to Statics and Dynamics*. Pre-print for Oxford University Press, 2008.

[60] E. Kraft, "A quaternion-based unscented kalman filter for orientation tracking," in *Proceedings of the Sixth International Conference of Information Fusion, 2003*, vol. 1, July 2003, pp. 47–54.

[61] X. Pennec, "Computing the mean of geometric features application to the mean rotation," report, March 1998.

[62] PS-Tech, *PST base documentation*. [Online]. Available: http://www.ps-tech.com/public/software/optical-trackers/optical-tracker-pst-base/softwarelist/20141204%20%20PST%20Base%20complete.pdf (visited on 07/09/2015).

A. Abbreviations

ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Network
AR	Autoregressive
AR	Augmented Reality
PCB	Printed Circuit Board
DCM	Direction Cosine Matrix
DOF	Degree of Freedom
DSSM	Dynamic State Space Model
ECEF	Earth-Centered-Earth-Fixed-Frame
ECI	Earth-Centered-Inertial-Frame
ENU	East-North-Up-Frame
FIS	Fuzzy Interference System
FSR	Full Scale Range
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HT	Hough Transform
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filter
LED	Light Emitting Diode
LUT	Look up Table
LLH	Representation of an object's location in Latitude, Longitude and Height
LSB	Least Significant Bit
MAP	Maximum a Posteriori
MD	Mahalanobis Distance
MEMS	Micro-Electro-Mechanical Systems
ML	Maximum Likelihood
MMSE	Minimum Mean Squared Error

MSE	Mean Squared Error
μC	Micro Controller
NED	North-East-Down-Frame
NL	Nonlinearity
pdf	Probability Density Function
RANSAC	Random Sample Consensus
RMSE	Root Mean Squared Error
RV	Random Variable
RW	Random Walk
SUT	Scaled Unscented Transformation
UDP	User Datagram Protocol
UKF	Unscented Kalman Filter
UT	Unscented Transformation
VR	Virtual Reality
VRPN	Virtual Reality Peripheral Network
ZOH	Zero-order Hold
3D	3 Dimensional

B. Notation

$\mathbf{a}, \boldsymbol{\sigma}$	Represents a vector; all vectors are lower case, bold letters or symbols
$\mathbf{A}, \boldsymbol{\Sigma}$	Represents a matrix; all matrices are capital, bold letters or symbols
$(\cdot)^T$	Represents the transpose
$\bar{(\cdot)}$	Represents the mean
$\hat{(\cdot)}$	Represents an estimation
$diag(\cdot)$	Represents the diagonal elements of the matrix
∇	Represents the vector differential operator
$\frac{\delta A}{\delta x}$	Represents partial derivative of A with respect to x
$(\dot{x})$	Represents the time derivative of x
$\|\| \cdot \|\|$	Represents the euclidean norm
$\bullet$	Represents the quaternion multiplication
$(\cdot)^{-1}$	Represents the matrix inverse
$\ominus_k$	Represents the k-times iterated morphologic erosion
$\oplus_k$	Represents the k-times iterated morphologic dilation
$\mathrm{E}[\cdot]$	Represents the expectation operator

C. List of Symbols

The List of Symbols is in chronological order according to its first definition:

Inertial Navigation System

$\mathbf{r}_B^A$	Absolute vector pointing from A to B
$\mathbf{v}_{EB}^N$	Velocity measured in the N-frame of the B-frame with respect to the E-frame
$\mathbf{R}_N^B$	Rotation-matrix, which rotates a vector from N-frame to B-frame
ϕ, θ, ψ	Euler angles as roll, pitch and yaw, respectively
σ	Rotation vector
$\mathbf{n}$	Axis of rotation
θ	Angle of rotation
$\mathbf{q}_A^B$	Quaternion representing the rotation from A-frame to B-frame
q_1, q_2, q_3, q_4	Elements of the quaternion
$s, \mathbf{v}$	Scalar and vector part of the quaternion, respectively
σ_N^B	Rotation vector representing the rotation from the navigation-frame to the body-frame
$v_{EB,\text{east}}^N$	Velocity of the body-frame in the east-direction with respect to the ECEF-frame measured in the navigation-frame.
$v_{EB,\text{north}}^N$	Velocity of the body-frame in the north-direction with respect to the ECEF-frame measured in the navigation-frame.
$\boldsymbol{\omega}_{NB}^B$	Rotation rate of the body-frame with respect to the navigation-frame
$\boldsymbol{\omega}_{IB}^B$	Measured rotation rate of the IMU with respect to the I-frame
$\boldsymbol{\omega}_{IE}^N$	Earth rotation in navigation-frame
$\boldsymbol{\omega}_{EN}^N$	Transport rate
φ	Latitude
Ω	Earth rotation
R_e	Radius of the curvature in the prime vertical
R_n	Meridian radius of the curvature
h	Height of the object

r_m	Mean earth radius
$\omega^N_{EN,\max}$	Maximal transport rate
$v_{\max}$	Maximal velocity
$\Delta\boldsymbol{\sigma}^{B,k}_{B,k-1}$	Rotation vector which rotates the B-frame from time t_{k-1} to t_k
$\Delta\boldsymbol{\theta}_k$	Output of the gyroscope integrated from time t_{k-1} to t_k
$\boldsymbol{\omega}^B_{IB,k}$	Output of the gyroscope at time t_k
ΔT	Sampling time
$\mathbf{q}^{B,k}_{B,k-1}$	Quaternion which rotates the B-frame from time t_{k-1} to t_k
$\mathbf{q}^{B,k}_{N}$	Quaternion which rotates the N-Frame to the B-frame at time t_k
$\mathbf{v}^N_{NB}$	Velocity of the B-frame with respect to the N-frame represented in the N-frame
$\mathbf{a}^B_{IB}$	Sensed acceleration by the IMU, also called specific force
$\dot{\mathbf{v}}_{\text{Cor,max}}$	Maximum acceleration due the Coriolis force
$\Delta\mathbf{v}^N_{NB,a,k}$	Velocity increment due to the specific force
$\mathbf{v}^N_{NB,k}$	Velocity of the B-frame with respect to the N-frame represented in the N-frame at time t_k
$\mathbf{g}^N$	Gravity-vector in the navigation-frame (NED)
$\Delta\mathbf{v}^N_{NB,k}$	Velocity increment of the B-frame with respect to the N-frame represented in the N-frame at time t_k
$\Delta\mathbf{v}^N_{NB,a,k}$	Velocity increment due to the specific force of the B-frame with respect to the N-frame represented in the N-frame at time t_k
$\Delta\mathbf{v}^N_{NB,g,k}$	Velocity increment due to gravity
$\Delta\mathbf{v}^B_{IB,k}$	Output of the accelerometer integrated from time t_{k-1} to t_k
$\mathbf{p}^N_B$	Position of the B-frame with respect to the N-frame
$\mathbf{p}^N_{B,k}$	Position of the B-frame with respect to the N-frame at time t_k

Inertial Sensors / Deterministic-Errors

$\tilde{\mathbf{x}}_c$	Calibrated measurement
$\tilde{\mathbf{x}}_d$	Measurement corrupted by deterministic-errors
$\mathbf{M}_{\text{Dev}}$	The scale factor matrix of device Dev, such as a for accelerometer
$\mathbf{b}_{\text{Dev},d}$	Deterministic bias of the device Dev, such as a for accelerometer
$\tilde{\mathbf{a}}^B_{IB,c}$	Compensated accelerometer measurement
$\tilde{\mathbf{a}}^B_{IB,d}$	Accelerometer measurement affected by deterministic errors
$\mathbf{M}_a$	Scale factor matrix for accelerometer
$\mathbf{b}_{a,d}$	Deterministic bias for accelerometer
$\mathbf{M}_{\text{asym}}$	Asymmetric part of scale factor matrix
$\mathbf{M}_{\text{sym}}$	Symmetric part of scale factor matrix
$\mathbf{R}$	Rotational part of scale factor matrix
$S_{xy}, S_{yx}, S_{xz},$ S_{zx}, S_{yz}, S_{zy}	Cross correlation factors of the scale factor matrix
$\mathbf{r}$	Error vector for optimization
E^t	The accumulative error at iteration t
$\mathbf{p}^t$	Parameter vector for optimization at iteration t
δ^t	Correction term at iteration t
α^t	Damping factor at iteration t
$\mathbf{J}^t$	Jacobian matrix at iteration t
τ	Fading factor
ϵ	Convergence threshold
$\sigma^2_{L,k}$	The local variance at time step k
N	Window length of the local variance estimation
$\mu_{L,k}$	Local mean at time step k
$\tilde{\mathbf{a}}_i$	Measured acceleration-vector at time step i.
$\sigma^2_{L,\text{min}}$	Minimal local variance
β	Local variance threshold for static-conditions
$\mathbf{k}_{\text{SP}}$, $\mathbf{k}_{\text{EP}}$	Vector for starting and ending point of static conditions, respectively
$k_{\text{SP},n}$, $k_{\text{EP,n}}$	Starting and ending point of static conditions for the n-th orientation, respectively
L_{min}	Minimal static condition length

$\tilde{\mathbf{a}}_n$	Mean acceleration for the n-th orientation
$\tilde{\mathbf{m}}_c^B$	Compensated magnetometer measurement
$\tilde{\mathbf{m}}_d^B$	Magnetometer measurement affected by deterministic errors
$\mathbf{M}_m$	Scale factor matrix for magnetometer
$\mathbf{b}_{m,d}$	Deterministic bias for magnetometer
$\tilde{\boldsymbol{\omega}}_{IB,c}^B$	Compensated gyroscope measurement
$\tilde{\boldsymbol{\omega}}_{IB,d}^B$	Gyroscope measurement affected by deterministic errors
$\mathbf{M}_\omega$	Scale factor matrix for Gyroscope
$\mathbf{b}_{\omega,d}$	Deterministic bias for gyroscope
$\sigma_{t_1\,\text{gyro}}^{t_2}$	Rotation vector from time t_1 to t_2 using gyroscope data
$\sigma_{t_1\,\text{gravity}}^{t_2}$	Rotation vector from time t_1 to t_2 using gravity data
$\boldsymbol{\Delta\Phi}_j$	Vector of accumulated rotation for the j-th experiment
$\mathbf{p}_i$	The i-th point of is a set of N points
$\mathbf{p}_0$	Origin of the plane
$\mathbf{P}$	Point on a plane
$\mathbf{n}$	Normal of the plane
$\mathbf{v}_1, \mathbf{v}_2$	Orthogonal basis of a plane
$\mathbf{p}_{\text{proj}}$	Projected point
R	Radius of a circle
$\mathbf{c}$	Center of a circle
Θ	Rotation angle
$\mathbf{r}_{\text{start}}, \mathbf{r}_{\text{stop}}$	Projected start and stop points, respectively

Inertial Sensors / Random-Errors

$\tilde{\mathbf{x}}$	Measurement affected by random errors
b	Drifting bias
ν, σ_ν^2	Error variable due to measurement noise and its according variance, respectively
$\tilde{\mathbf{x}}_c$	Measurement, which has already been compensated for deterministic errors
$\nu_{\text{NL}}, \sigma_{\text{NL}}^2$	Error variable due to nonlinearity and its according variance, respectively
$\nu_{\text{noise}}, \sigma_{\text{noise}}^2$	Error variable due to sensor noise and its according variance, respectively

NL(%)	Nonlinearity of the sensor expressed in % of the Full Scale Range (FSR)
b_k	Drifting bias at time step k
$w_{\text{RW}}, \sigma^2_{\text{RW}}$	Random walk error and its according variance, respectively
$\sigma^2_{\text{Allan}}(\tau)$	Allan variance at cluster time τ
τ	Cluster time
m	Number of clusters
n	Length of a cluster
$\Theta_i(\tau)$	Average of cluster i for cluster time tau
$N_{\text{Allan}}, K_{\text{Allan}}$	Noise terms in the Allan variance analysis

Monocular Position Tracking

p_{last}	Percent of pixels in the last N histogram bins
B	Camera setting for brightness
$B_{\text{min}}, B_{\text{max}}$	Minimum and maximum brightness values to increase or decrease exposure
E	Camera setting for exposure
$\mathbf{p}$	Position of the tracked object
$\mathbf{v}$	Velocity of the tracked object
$\mathbf{n}_a$	Unobserved acceleration modeled as zero mean white Gaussian process noise
$\mathbf{Q}_a$	Covariance matrix of $\mathbf{n}_a$
$\mathbf{y}_k$	Measurement at time step k
$\mathbf{v}_p$	Measurement noise
$\mathbf{R}_p$	Covariance matrix of $\mathbf{v}_p$
$\mathbf{\Delta}T$	Sampling time
u, v	Image coordinates expressed in pixels
$u_{\text{min}}, u_{\text{max}},$ $v_{\text{min}}, v_{\text{max}}$	Minimum and maximum image coordinates for search region
$u_{\text{pred}}, v_{\text{pred}}$	Predicted image coordinates
$\Delta u, \cdot \Delta v$	Standard deviation of the predicted image coordinates
r	Predicted radius of the projected object
Δr	Standard deviation of the predicted radius

$\mathbf{M}_{\text{cam}}$	Intrinsic camera matrix
f	Focal length
R	Radius of the object
z	z-Position of the object expressed in camera coordinates
$\mathbf{J}$	Jacobian of image projection
$I_{\text{Hue}}(u, v)$	Hue at pixel points (u,v)
$I_{\text{Sat}}(u, v)$	Saturation at pixel points (u,v)
$I_{\text{Val}}(u, v)$	Value at pixel points (u,v)
$I_{\text{Hue,min}}$, $I_{\text{Hue,max}}$	Minimum and maximum thresholds for Hue, Saturation and Value,
$I_{\text{Sat,min}}$, $I_{\text{Sat,max}}$	respectively
$I_{\text{Val,min}}$, $I_{\text{Val,max}}$	
$\mathbf{I}_{\text{Clean}}$	Morphologic cleaned image
$\mathbf{I}$	Image
SE	Structure element for morphologic clean up
T_{upper}, T_{lower}	Upper and lower threshold for canny edge detection
f_x, f_y	Focal lengths in x and y direction, respectively, expressed in pixels
c_x, c_y	Principal points in x and y direction, respectively, expressed in pixels
$X'_{\text{corr}}, Y'_{\text{corr}}$	Points corrected for radial and tangential distortion
$k_1, k_2, k_3, k_4, k_5, k_6$	Coefficients for radial distortion
p_1, p_2	Coefficients for tangential distortion
$\mathbf{p}_{\text{norm}}$	Contour points in normalized coordinates
$\mathbf{p}_{\text{Unit}}$	Contour points projected on unit sphere
$\mathbf{r}_{\text{axis}}$	Symmetry axis a the cone
$\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}, \hat{\mathbf{r}}$	Unit vectors of the spherical coordinate system
θ, ϕ	Angles of the spherical coordinate system
u', v', λ	Reprojected image coordinates (projective compensation)
$r, \mathbf{c}_{u'}, \mathbf{c}_{v'}$	Radius, x and y coordinates for the fitted circle, respectively
$\mathbf{p}_{u'}, \mathbf{p}_{v'}$	Coordinates of the reprojected point
ϵ	Distance threshold for RANSAC
S_i	Set of inliers
w	Probability of drawing an inlier
s	Sample size

p	Probability of drawing at least once a sample, which is outlier free
N_p	Number of samples to draw at least once a sample, which is outlier free with probability p
N	Actual number of RANSAC iterations
$N_{\max}$	Maximum number of RANSAC iterations
$w_{\min}$	Minimum inlier ratio
FG_{Circle}	Amount of foreground pixels within the fitted circle
FG_{Total}	The total amount of foreground pixels
A_{Circle}	Area of the fitted circle, expressed in pixels
d	Distance of the object
$\mathbf{r}_{\text{sphere}}$	Estimated circle center
$\mathbf{n}_{\text{sphere}}$	Unit vector pointing in the direction of the object
$\mathbf{P}_{\text{sphere}}$	Location of the sphere

Sensor Fusion for Tracking / Overview

$\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{n}_k)$	Discrete-time transfer function
$\mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)$	Discrete-time measurement function
$\mathbf{x}_k$	Discrete-time state vector at time step k
$\mathbf{y}_k$	Discrete-time measurement vector at time step k
$\mathbf{u}_k$	Discrete-time known exogenous input at time step k
$\mathbf{n}_k$	Discrete-time process noise at time step k
$\mathbf{v}_k$	Discrete-time measurement noise at time step k
k	Discrete time step
$\hat{\mathbf{x}}_{k\|k-1}$	Predicted *a priori* mean of state vector **x** at time step k given measurements up to and including time step k-1
$\hat{\mathbf{y}}_{k\|k-1}$	Predicted *a priori* mean of the expected measurement at time step k given measurements up to and including time step k-1
$\hat{\mathbf{x}}_k$	Corrected *a posteriori* state vector at time step k given measurements up to and including time step k
$p(\mathbf{x}\|\mathbf{y}_1, \ldots, \mathbf{y}_n)$	The posterior probability distribution is the distribution over the states **x** , which is conditioned on the observed measurements.

$p(\mathbf{y}_1, \ldots, \mathbf{y}_n \vert \mathbf{x})$	Likelihood function representing the joint probability of the measurements $\mathbf{y}_1, \ldots, \mathbf{y}_n$ given the states $\mathbf{x}$.
$p(\mathbf{x})$	The prior distribution over the states representing the information of the states before using information of any measurement.

Sensor Fusion for Tracking / Selected Fusion Strategy

$\mathbf{x}$	Random variable
$\mathbf{y}$	Random variable as output of an arbitrary non-linear function
L	Dimension of random variable $\mathbf{x}$
$g(\mathbf{x})$	Arbitrary non-linear function
$\bar{\mathbf{x}}$	Mean of random variable $\mathbf{x}$
$\mathbf{P}_\mathbf{x}$	Covariance matrix of random variable $\mathbf{x}$
$\mathbf{S}$	Set of deterministic chosen samples based on a random variable
$\mathbf{S}_i$	i-th element of a set of deterministic chosen samples
$\mathcal{X}$	Sigma points of random variable $\mathbf{x}$
$\mathcal{X}_i$	i-th element of a set of sigma points
w_i	Weighting factor associated with i-th element of a set of sigma points
$w_i^{(m)}$	Weighting factor for mean calculation, associated with i-th element of a set of sigma points
$w_i^{(c)}$	Weighting factor for covariance calculation, associated with i-th element of a set of sigma points
λ	Scaling parameter in the Scaled Unscented Transformation
α	Scaling parameter in the Scaled Unscented Transformation
β	Scaling parameter in the Scaled Unscented Transformation
κ	Scaling parameter in the Scaled Unscented Transformation
$\mathcal{Y}$	Transformed sigma points of random variable $\mathbf{x}$
$\mathcal{Y}_i$	i-th element of a set of transformed sigma points
$\bar{\mathbf{y}}$	Mean of the transformed random variable y
$\mathbf{P}_\mathbf{y}$	Covariance matrix of the transformed random variable $\mathbf{y}$
$\mathbf{P}_\mathbf{xy}$	Cross-covariance matrix of the random variables x and y
$a^G_{\text{lin},x}$	Linear acceleration projected on x-axis in global-frame
$a^G_{\text{lin},y}$	Linear acceleration projected on y-axis in global-frame

$\mathbf{a}_{\text{lin}}^{G}$	Linear acceleration-vector in global-frame
θ	Angle or the car's orientation
$\mathbf{a}_{m}^{B}$	Measured acceleration-vector in body-frame
$\mathbf{g}^{G}$	Gravity-vector in global-frame
$\bar{\mathbf{a}}_{\text{lin}}^{\mathbf{G},1^{st}O.}$	Mean of the linear acceleration in global-frame using 1^{st} order linearization
$\mathbf{P}_{a_{\text{lin}}^{G}}^{1^{st}O.}$	Covariance matrix of the linear acceleration in global-frame using 1^{st} order linearization
$\mathbf{x}_k$	State vector at time step k
$\mathbf{x}_k^a$	Augmented state vector at time step k
$\mathbf{P}_{\mathbf{x}}$	Covariance matrix of the state vector $\mathbf{x}$
$\mathbf{P}_{\mathbf{x}}{}^{a}$	Covariance matrix of the augmented state vector $\mathbf{x}^a$
$\mathbf{R}_{\mathbf{v}}$	Covariance matrix of the process noise $\mathbf{v}_k$
$\mathbf{R}_{\mathbf{n}}$	Covariance matrix of the measurement noise $\mathbf{n}_k$
$\mathbf{x}_0$	Initial state vector
$\mathbf{P}_{\mathbf{x_0}}$	Covariance matrix of initial state vector $\mathbf{x}_0$
$\mathcal{X}^x$	Sigma points of the original state vector $\mathbf{x}$
$\mathcal{X}^a$	Sigma points of the augmented state vector $\mathbf{x}^a$
$\mathcal{X}^v$	Sigma points of the process noise $\mathbf{v}_k$
$\mathcal{X}^n$	Sigma points of the measurement noise $\mathbf{n}_k$
$\mathcal{X}_{k\|k-1}$	Predicted *a priori* sigma points at time step k given measurements up to and including time step k-1
$\hat{\mathbf{x}}_{k\|k-1}$	Predicted *a priori* state estimation at time step k given measurements up to and including time step k-1
$\mathbf{P}_{\mathbf{x}_{k\|k-1}}$	Predicted *a priori* covariance matrix of state vector $\mathbf{x}$ at time step k given measurements up to and including time step k-1
$\mathcal{Y}_{k\|k-1}$	Predicted *a priori* sigma points of the expected measurement at time step k given measurements up to and including time step k-1
$\hat{\mathbf{y}}_{k\|k-1}$	Predicted *a priori* mean of the expected measurement at time step k given measurements up to and including time step k-1
$\mathbf{P}_{\tilde{\mathbf{y}}_{k\|k-1}}$	Predicted *a priori* covariance of the expected measurement at time step k given measurements up to and including time step k-1

$\mathbf{P}_{\mathbf{x}_k,\mathbf{y}_k}$	State-measurement cross-covariance at time step k
$\mathbf{K}_k$	Optimal Kalman gain at time step k
$\mathbf{P}_{\mathbf{x}_k}$	Corrected *a posteriori* covariance matrix of state vector $\mathbf{x}$ at time step k given measurements up to and including time step k
$\Delta\mathbf{x}_k$	Kalman updated correction term for time step k
$\mathbf{y}_k$	Actual measurement at time step k
$\hat{\mathbf{x}}_k$	Corrected *a posteriori* estimation at time step k given measurements up to and including time step k
k	Discrete time step

$\mathbf{q}_U^{B,k}$	Rotation from the user-frame to the body-frame at time step k
$\mathbf{p}_{B,k}^U$	Position of the body-frame with respect to the user-frame at time step k
$\mathbf{v}_{UB,k}^U$	The velocity of the body-frame with respect to the user-frame, measured in the user-frame at time step k
$\mathbf{b}_{\omega_{\text{RW}},k}$, $\mathbf{b}_{a_{\text{RW}},k}$, $\mathbf{b}_{m_{\text{RW}},k}$	RW biases for the gyroscope, accelerometer and magnetometer, at time step k,respectively
$\mathbf{g}_k^U$	Reference vector the gravity-field with respect to the user-frame at time step k
$\mathbf{m}_k^U$	Reference vector the magnetic field with respect to the user-frame at time step k
$\mathbf{r}_{\text{lever},k}^B$	Lever arm at time step k, displacement of the sensor-frame to the body-frame
$\boldsymbol{\omega}_k$	At time step k measured angular rate
$\mathbf{a}_k$	At time step k measured acceleration
ΔT_k	Sampling time at time step k
$\mathbf{n}_{b,\omega_{\text{RW}}}$, $\mathbf{n}_{b,a_{\text{RW}}}$, $\mathbf{n}_{b,m_{\text{RW}}}$	Zero mean Gaussian noises driving the bias random walk of the gyroscope, accelerometer and magnetometer, respectively.
$\sigma^2_{b,\omega_{\text{RW}}}$, $\sigma^2_{b,_{\text{RW}}}$, $\sigma^2_{b,m_{\text{RW}}}$	Variance for the noises driving the bias random walk of the gyroscope, accelerometer and magnetometer, respectively
$\mathbf{n}_{m_{\text{RW}}}$, $\sigma^2_{m_{\text{RW}}}$	Zero mean white Gaussian noise driving the RW of the reference vector of the magnetic field and its variance, respectively
$\mathbf{n}_{a,v,k}$, $\sigma^2_{a,v}$	Zero mean white Gaussian noise of the accelerometer measurements noises and its variance, respectively
$\mathbf{n}_{\omega,v,k}$, $\sigma^2_{\omega,v}$	Zero mean white Gaussian noise of the gyroscope measurements noises and its variance, respectively
$\boldsymbol{\Sigma}^2$	Covariance matrix for uncorrelated variables
$\mathbf{f}_q$	Transfer function for the quaternion
$\mathbf{f}_p$	Transfer function for the position
$\mathbf{f}_v$	Transfer function for the velocity

$\mathbf{q}_U^N$,$\mathbf{R}_U^N$	Quaternion and rotation matrix representing the rotation from the navigation-frame to the user-frame, respectively
$\boldsymbol{\omega}_{\text{corr},k}$	Gyroscope measurement corrected for deterministic and random errors at time step k
$\mathbf{a}_{\text{corr},k}$	Accelerometer measurement corrected for deterministic and random errors at time step k
$\Delta\mathbf{v}_{UB,a,k}^U$	Velocity update due to the specific force for the user-frame at time step k
$\Delta\mathbf{v}_{UB,g,k}^U$	Velocity update due to gravity for the user-frame at time step k
$\mathbf{h}_m$	Measurement function for the magnetometer
$\hat{\mathbf{m}}_k^B$	Estimate for the real magnetometer measurement at time step k
$\mathbf{m}_k^B$	Real magnetometer measurement at time step k
$\mathbf{v}_{m,k}$, $\sigma_{m,\text{noise}}^2$	The measurement noise for the magnetometer and its according variance, respectively
$\mathbf{m}_{\text{corr},k}^B$	Magnetometer measurement, corrected for deterministic errors, at time step k
$\mathbf{a}_{\text{lever}}$	Acceleration due to lever arm effect
$\Delta\mathbf{p}_{\text{lever},k}$	Offset of the position due to the lever arm effect
$\mathbf{h}_p$	Measurement function for the monocular position tracking system
$\mathbf{v}_{p,k}$, $\mathbf{R}_{p,U}$	The measurement noise for the monocular position tracking system and its according covariance matrix in the user-frame, respectively
$\mathbf{R}_{p,\text{Cam}}$	Covariance matrix in the camera-frame for measurement noise of the monocular position tracking system
$\hat{\mathbf{p}}_{pm,k}^U$	Estimation of the position-measurement-frame with respect to the user-frame
$\mathbf{p}_{pm,k}^{\text{Cam}}$	Location of position-measurement-frame with respect to the camera-frame at time step k
$\mathbf{p}_{pm,k}^U$	Location of position-measurement-frame with respect to the user-frame at time step k
$\mathbf{p}_U^{\text{Cam}}$	Location of user-frame with respect to the camera-frame at time step k
$\mathbf{R}_{\text{Cam}}^U$	Rotation matrix, representing the rotation from the camera-frame to the user-frame

Sensor Fusion for Tracking / Adaptations

$\mathbf{x}_k^{\text{quat}}$	Quaternion part of the state vector
$\mathbf{x}_k^{\text{rest}}$	Rest part of the state vector
$\Delta\mathbf{x}_k^{\text{quat}}$	Quaternion perturbation
$\tilde{\mathbf{x}}_k^{\text{quat}}$	Perturbed quaternion
$\mathcal{X}_k^{\text{quat}}$	Sigma points of the quaternion states
$\mathcal{X}_k^{\text{rest}}$	Sigma points of the rest states
$\Delta\mathcal{X}_k^{\sigma}$	Perturbation vector for the orientation in terms of rotation vectors
$\Delta\mathcal{X}_k^{\text{rest}}$	Perturbation vector for the rest states
$\Delta\mathcal{X}_k^{\text{quat}}$	Perturbation vector for the orientation in terms of quaternions
$(\Delta\mathcal{X}_{k\|k-1}^{\text{quat}})_{t,i}$	i-th column of the predicted perturbation vector at iteration t represented as quaternion
$(\hat{\mathbf{x}}_{k\|k-1}^{\text{quat}})_t$	Quaternion part of the predicted mean at iteration t
$(\hat{\mathbf{x}}_{k\|k-1}^{\text{rest}})$	Rest part of the predicted mean
$(\mathcal{X}_{k\|k-1}^{\text{quat}})_{t,i}$	i-th column of quaternion part of the predicted sigma points at iteration t
$\bar{e}_t$	Barycentric mean of the delta quaternions
θ	Average angle of rotation for the mean delta quaternion
$(\Delta\mathcal{X}_{k\|k-1}^{\sigma})$	Predicted perturbation vector for the orientation, represented as rotation vector
$(\Delta\mathcal{X}_{k\|k-1}^{\text{rest}})$	Predicted perturbation vector for the rest states
$\Delta\mathbf{x}_k^{\sigma}$	Correction term for the orientation, represented as rotation vector
$\Delta\mathbf{x}_k^{\text{quat}}$	Correction term for the orientation, represented as quaternion
$\Delta\mathbf{x}_k^{\text{rest}}$	Correction term for rest states
$\hat{\mathbf{x}}_k^{\text{quat}}$	Corrected *a posterior* estimation of the quaternion states
$\hat{\mathbf{x}}_k^{\text{rest}}$	Corrected *a posterior* estimation of the rest states
t_{delay}	Time delay between IMU and the monocular positional tracking system
n_{queue}	Size of the delay queue
f_{Filter}	Operation frequency of the UKF
$\mathbf{x}$	Sequence of intensities
$\mathbf{x}_{\text{norm}}$	Normalized sequence of intensities
$R_{xy}(l)$	Cross-correlation for the lag l

ΔT_{base}	Time base for the interpolation
n_{delay}	Time delay expressed in samples
$t_{\text{delay,transmission}}$	Time delay of the data transmission
$t_{\text{delay,processing}}$	Time delay of the data processing
MD	Mahalanobis distance
d	Degree of freedom for the χ^2 distribution
$\chi^2_{d,\max}$	Threshold to reject outliers
$\bar{\chi}^2_d$	Mean of χ^2 distribution with DOF d
c_k	Covariance scaling factor at time step k
MD_k	Mahalanobis distance at time step k
$\mathbf{R}_k$	Adapted covariance matrix at time step k
$\mathbf{R}_0$	Initial covariance
G	Noise scaling adaption gain

Application: Low Cost 3D Interaction Device

$\mathbf{P}_1^{Cam}, \mathbf{P}_2^{Cam}, \mathbf{P}_3^{Cam}, \mathbf{P}_4^{Cam}$	User chosen points defining the user coordinate system
$\mathbf{x}_U^{Cam}, \mathbf{y}_U^{Cam}, \mathbf{z}_U^{Cam}$	X, Y, Z axes of the user coordinate system expressed in camera coordinates
$\mathbf{R}_U^{Cam}$	Rotation from user-frame to camera-frame
$\mathbf{R}_U^B$	Rotation from user-frame to body-frame
$\mathbf{R}_B^{NED}$	Rotation from body-frame to NED-frame
$\mathbf{R}_U^{NED}$	Rotation from user-frame to NED-frame

D. List of Figures

E. List of Tables