TUHH

Technische Universität Hamburg-Harburg

Bachelor Thesis

2010

Data Fusion of Accelerometric, Gyroscopic, Magnetometric and Barometric Information for Attitude and Altitude Estimation of an Unmanned Quadrocopter

Robert Kesten

Institut für Zuverlässigkeitstechnik
Technische Universität Hamburg–Harburg

Hamburg, den 8. Februar 2010

Gutachter:

Prof. Dr. Uwe Weltin

Prof. Dr. Herbert Werner

Betreuer:

Dipl. Ing. Jonas Witt

Hamburg, den 8. Februar 2010

Ich, Robert Kesten (Student an der Technischen Universität Hamburg–Harburg, Matrikelnummer 30538), versichere an Eides statt, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Die Arbeit wurde in dieser oder ähnlicher Form noch keiner Prüfungskommission vorgelegt.

Robert Kesten

Abstract

This thesis focuses on creating a state observer in order to estimate values based on partly redundant, noise afflicted sensor data. Filters for attitude and altitude estimation are developed. The Kalman filter technique is applied and modified to introduce a new attitude quaternion filter concept. This modified filter outperforms a classical extended Kalman filter design for attitude quaternion estimation. An altitude controller is developed using the LQR method and a roll/pitch/yaw angle attitude controller is fitted for quaternion input. The results are applied to improve the TUHH quadrocopter platform developed by Dipl. Ing. Jonas Witt.

Contents

Chapter 1

Basics

This thesis is concerned with state estimation and control of the attitude and altitude of an aerial vehicle based on information given by on board sensors. These sensors do not directly measure attitude and altitude, but they allow for a conclusion about both.

State estimation is often necessary as a prerequisite for control. This is illustrated in figure 1.1. Due to limitations of the measurement, not the system state vector $\vec{x}$, but a disturbed measurement vector $\tilde{\vec{z}}$ is available. From this measurement vector, an estimator tries to deduce the actual system state. This estimated system state $\hat{\vec{x}}$ can then be used by a controller.

Data fusion is encountered when dealing with state estimation. The output of different sensors is combined to draw a conclusion about the state to be estimated. Fusing accelerometric and magnetometric data allows for attitude calculation. By further fusing this calculated attitude with gyroscopic data, the effect of uncorrelated disturbances acting on the different sensors on the state estimate can be reduced. Barometric altimeter data allows for altitude estimation directly. The accuracy of the altitude estimate can be improved by fusion with data concerning the vertical acceleration.

Estimation can also involve processing of other information like the controller's output. This aspect of state estimation is not covered in this thesis.

In the following sections, a quick glance is thrown at the principle of a quadrocopter, state estimation using the Kalman filter technique and quaternions and their utilization for attitude representation. The second chapter deals with the design of the actual state estimation filters. Attitude estimation based on accelerometric, magnetometric and gyroscopic data and altitude estimation based on barometric altimeter data and vertical acceleration data are considered separately. Attitude and altitude controllers are engineered in chapter three. Chapter four deals with employing the developments on a real

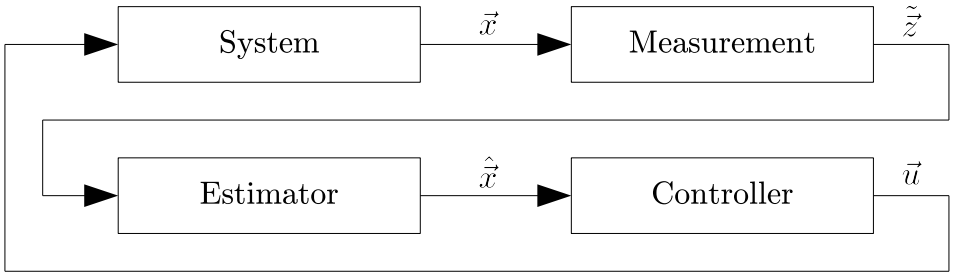

Figure 1.1: Problem structure overview

quadrocopter. Finally, in chapter five, test results are shown and discussed. A conclusion and an outlook on possible future work is presented.

1.1 Quadrocopter

The developed estimation and control routines are to be deployed on a small-scale unmanned quadrocopter. A quadrocopter, as sketched in figure 1.2, is an aerial vehicle with four rotors in a plane. It is capable of vertical take off and landing. Quadrocopters are related to helicopters and are sometimes classified as such.

As opposed to standard helicopters, the quadrocopter's rotors have fixed pitch blades. One pair of rotors rotates clockwise, the other pair rotates counterclockwise. By adjusting the speeds of rotation, the moments $\vec{M}$ and forces $\vec{F}$ generated by the rotors can be altered which allows the quadrocopter to turn around the roll, pitch and yaw axes and accelerate along the yaw axis. The system is not fully actuated and looses two of its six degrees of freedom at any instant.

The quadrocopter considered has a sensor accuracy, operating range and flight speed that allow for some model simplifications compared to aerial vehicles in general. Factors of influence like the Coriolis effect, the earth's turn rate, the earth's curvature and the transport rate, all described in [8], can be neglected.

1.2 Coordinate Systems

The following coordinate systems are used in this thesis:

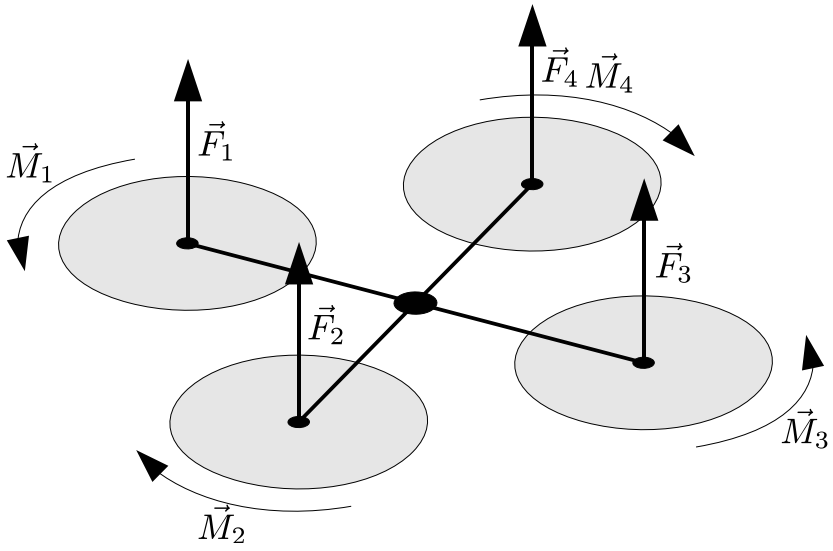

Figure 1.2: Sketch of a quadrocopter including forces and moments generated by the rotors

Body Coordinate System

The body coordinate system shall be indexed by b. The origin lies in the center of the quadrocopter. The x^b- axis points forward, while forward is characterized by the position of the first rotor. The y^b- and z^b axes point toward the right and bottom side, respectively.

Navigation Coordinate System

The navigation coordinate system shall be indexed by n. The origin coincides with that of the body coordinate system. The x^n- axis points in north direction, the y^n points toward east. z^n denotes the direction of gravity.

Ambiance Coordinate System

The ambiance coordinate system shall be indexed by a. While having the same orientation as the navigation coordinate system, its origin is earth-fixed at the starting location of the quadrocopter.

As nomenclature, vectors will have an upper index showing what coordinate system they are declared in, e.g. $\vec{v}^n$ is a speed given in the navigation coordinate system.

A quaternion q_b^n or a directional cosine matrix $\mathbf{C}_b^n$ denotes a rotational transformation from coordinates of system b into coordinates of system n, while $q_n^b = (q_b^n)^{-1}$ or $\mathbf{C}_n^b$ describe the inverse transformation. Thus, $\vec{v}^n = \mathbf{C}_b^n \vec{v}^b$ transforms a speed vector given in body coordinates into a speed in navigation coordinates.

1.3 Quaternions

Quaternions are an extension to complex numbers, introducing two additional complex parts

$$\vec{q} = a1 + bi + cj + dk \tag{1.1}$$

where a, b, c and d are real numbers while for 1, i, j, k the following holds:

$$i^2 = j^2 = k^2 = ijk = -1 \tag{1.2}$$

Consequently, as a set, a quaternion is an element of a four-dimensional vector space over the real numbers. We will usually use this representation:

$$\vec{q} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \tag{1.3}$$

a is also referred to as the scalar part of the quaternion, while $\begin{pmatrix} b & c & d \end{pmatrix}^T$ is called the vector part.

Quaternions can be multiplied, the product is given by equation (1.2). The quaternion product of $\vec{q}_1(a, b, c, d)$ and $\vec{q}_2(e, f, g, h)$ will be symbolized by $\bullet$. It can also be interpreted as a matrix/vector product:

$$\vec{q}_1 \bullet \vec{q}_2 = \begin{pmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{pmatrix} \begin{pmatrix} e \\ f \\ g \\ h \end{pmatrix} \tag{1.4}$$

The quaternion product is non commutative.

1.3.1 Rotations

Quaternions can be and are mostly used to describe rotations in $\Re^3$.

Any attitude in $\Re^3$ can be uniquely described with an orientation vector

$$\vec{\theta}_1 = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \end{pmatrix}. \tag{1.5}$$

The direction of $\vec{\theta}$ defines a rotation axis. A coordinate system has to be turned around this axis by an angle $\theta_1 = |\vec{\theta}_1|$ to describe the desired attitude.

Let $\theta \in [0, 2\pi\}$, then, there is one other orientation vector describing the same attitude with a rotation in the opposite direction:

$$\vec{\theta}_2 = -\frac{\vec{\theta}_1}{\theta_1}\theta_2 \tag{1.6}$$

$$\theta_2 = 2\pi - \theta_1 \tag{1.7}$$

An orientation vector can be stored in a unit quaternion, a quaternion of length $|\vec{q}| = 1$:

$$\vec{q} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) \frac{\vec{\theta}}{\theta} \end{pmatrix} \tag{1.8}$$

Let $\vec{q}_b^n = (a, b, c, d)^T$ be a quaternion describing the transformation from a coordinate system n to a coordinate system b. The inverse of the quaternion describes the inverse transformation:

$$\vec{q}_n^b = (\vec{q}_b^n)^{-1} = \begin{pmatrix} a \\ -b \\ -c \\ -d \end{pmatrix} \tag{1.9}$$

A directional cosine matrix can be transformed into a quaternion and vice versa, which is described in [8].

Coordinate transformations can be handled directly using quaternions,

$$\vec{v}^n = \mathbf{C}_b^n \vec{v}^b \tag{1.10}$$

$$\begin{pmatrix} 0 \\ \vec{v}^n \end{pmatrix} = \vec{q}_b^n \bullet \begin{pmatrix} 0 \\ \vec{v}^b \end{pmatrix} \bullet \vec{q}_n^b \tag{1.11}$$

both perform the same transformation.

In the following, $\vec{\theta}(\vec{q}_1)$ denotes the shortest orientation vector representing the same attitude as $\vec{q}_1$. Conversely, $\vec{q}(\vec{\theta}_1)$ denotes a quaternion representing the same attitude as $\vec{\theta}_1$.

For more information on quaternions, refer to [8].

1.4 Kalman Filtering

1.4.1 Linear Kalman Filter

Kalman Filtering is a concept for state estimation. For linear systems and measurements only afflicted with white noise, it gives the optimal system state estimate with respect to the sum of squared estimate errors.

Consider a system model

$$\vec{x}_{k+1} = \mathbf{\Phi}_k \vec{x}_k + \mathbf{\Gamma}_k \vec{u}_k + \mathbf{G}_k \vec{w}_k \tag{1.12}$$

with state vector $\vec{x}_k$ and input vector $\vec{u}_k$. $\vec{w}_k$ is a white noise term with expected value

$$E[\vec{w}_i \vec{w}_k^T] = \begin{cases} \mathbf{Q}_k & i = k \\ \mathbf{0} & i \neq k \end{cases} \tag{1.13}$$

The measurement equation

$$\vec{z}_k = \mathbf{H}_k \vec{x}_k + \vec{v}_k \tag{1.14}$$

with measurement vector $\vec{z}_k$ contains a white noise term $\vec{v}_k$ for measurement noise, where

$$E[\vec{v}_i \vec{v}_k^T] = \begin{cases} \mathbf{R}_k & i = k \\ \mathbf{0} & i \neq k \end{cases} \tag{1.15}$$

Prediction

The predicted (a priori) system state vector

$$\hat{\vec{x}}_{k+1} = \mathbf{\Phi}_k \hat{\vec{x}}_k^+ + \mathbf{\Gamma}_k \vec{u}_k \tag{1.16}$$

has an uncertainty given by its covariance matrix

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k \mathbf{P}_k^{+} \mathbf{\Phi}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T. \tag{1.17}$$

Estimation

The predicted system state is improved by taking into account the measurement data. Using the Kalman gain

$$\mathbf{K}_k = \mathbf{P}_k^{-} \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^{-} \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1} \tag{1.18}$$

the estimated (a posteriori) system state vector and the corresponding covariance matrices are

$$\hat{\vec{x}}_k^{+} = \hat{\vec{x}}_k^{-} + \mathbf{K}_k \left(\tilde{\vec{z}}_k - \mathbf{H}_k \hat{\vec{x}}_k^{-} \right) \tag{1.19}$$

$$\mathbf{P}_k^{+} = \mathbf{P}_k^{-} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^{+}. \tag{1.20}$$

More information can be found in [3].

1.4.2 Extended Kalman Filter

The Extended Kalman Filter (EKF) is used to consider nonlinearities. The system matrix $\mathbf{\Phi}_k$ is a linearization of the nonlinear system equation around $\hat{\vec{x}}_k$ calculated for each time step k.

$$\hat{\vec{x}}_k^{-} = \hat{\vec{f}} \left(\hat{\vec{x}}_k^{+}, \vec{u}_k \right) \tag{1.21}$$

$$\mathbf{\Phi}_k = \frac{\partial \vec{f} \left(\hat{\vec{x}}_k^{+}, \vec{u}_k \right)}{\partial \vec{x}_k} \tag{1.22}$$

$$\vec{z}_k = \vec{h} \left(\vec{x}_k \right) + \vec{v}_k \tag{1.23}$$

$$\mathbf{H}_k = \frac{\partial \vec{h} \left(\hat{\vec{x}}_k^{+} \right)}{\partial \vec{x}_k} \tag{1.24}$$

The update equation becomes

$$\hat{\vec{x}}_k^{+} = \hat{\vec{x}}_k^{-} + \mathbf{K}_k \left(\tilde{\vec{z}}_k - \vec{h}_k \left(\hat{\vec{x}}_k^{-} \right) \right) \tag{1.25}$$

while the covariance propagation and Kalman gain equations remain unchanged, using the linearized matrices from above.

Chapter 2

Estimator Design

In this chapter, the estimator block, previously shown in figure 1.1, is examined in detail. A concept is developed for fusing measured gyroscopic data $\tilde{\vec{\omega}}$, magnetometric data $\tilde{\vec{m}}$, accelerometric data $\tilde{\vec{a}}$ and altimetric data $\tilde{s}$ in order to obtain estimates for the angular rate $\hat{\vec{\omega}}^b$, the attitude $\hat{\vec{q}}_b^n$, the altitude $\hat{s}$ and the vertical velocity $\hat{v}$.

The estimator's structure is shown in figure 2.1.

2.1 Measurement Data Characteristics

The gyroscopic three component angular rate data is denoted as $\vec{\omega}$. We assume that we do not possess exact angular rate information, but a measurement $\tilde{\vec{\omega}}$. Angular rate measurements are available to us at a sample frequency of $f_\omega = 1/\tau_\omega$. The measurement is bias free, but it is disturbed by an additive white Gaussian noise vector $\vec{\mu}_\omega$:

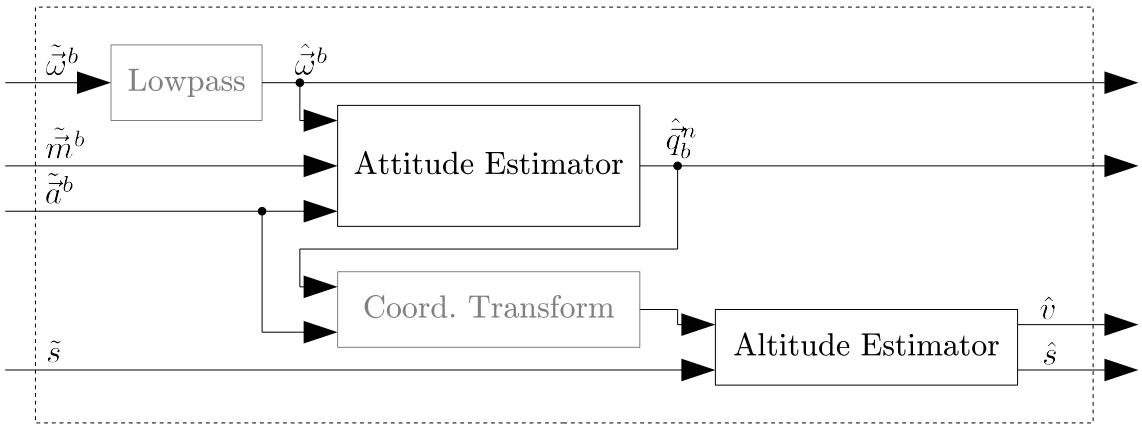

Figure 2.1: Estimator structure overview

$$\tilde{\vec{\omega}} = \vec{\omega} + \vec{\mu}_\omega \tag{2.1}$$

The white noise vector has the same magnitude in all components. The magnitude can be characterized by the variance σ_ω^2.

Besides the angular rates, we possess information about an acceleration vector $\vec{a}$, a magnetic field vector $\vec{m}$ and an altitude in form of a scalar s. Everything previously mentioned about $\vec{\omega}$ also holds for these vectors, and $\tilde{\vec{a}}$, $\vec{\mu}_a$, σ_a^2, τ_a, $\tilde{\vec{m}}$, $\vec{\mu}_m$, σ_m^2, τ_m as well as $\tilde{s}$, μ_s, σ_s^2, τ_s are defined analogically.

The vectors $\tilde{\vec{\omega}}$, $\tilde{\vec{a}}$ and $\tilde{\vec{m}}$ are all thought of as measurements of strap down sensors. These measurements are given in body coordinates.

In a real system, there are other, additional systematic disturbances influencing the measured data. The least significant such disturbances can be expected for the angular rate measurements, the most significant ones occur for the altitude measurements using a barometric altimeter. This was minded when designing the estimator's structure in figure 2.1. The altitude measurement does not influence the attitude estimation, and magnetometer and acceleration measurements do not influence the angular rate estimation.

Of course, it would also be possible to design one single (Kalman) filter for attitude- and altitude estimation. Anyhow, such an approach will not be examined in this paper.

2.2 Attitude Estimation

In this section, strategies for attitude estimation are developed, described and evaluated. Accelerometric, magnetometric and gyroscopic data are used in this process. At first, it is shown how this data can generally be used in order to obtain attitude information. Then, different approaches for data fusion and filtering are shown.

Before the actual filter design is described, a way to calculate attitude based on acceleration and magnetic field data is shown. Also, different approaches on averaging attitudes are presented.

2.2.1 Attitude Calculation

Absolute Attitude Calculation

When the system is not accelerated, $\vec{a}^b$ points into the direction of gravity.

The unit vectors $\vec{x}_n^b$, $\vec{y}_n^b$ and $\vec{z}_n^b$ that define the navigation coordinate system in the body coordinate system can be calculated as follows:

$$\vec{z}_n^b = \frac{\vec{a}^b}{\|\vec{a}^b\|} \tag{2.2}$$

The $\vec{z}_n^b$ part of $\vec{m}^b$ has to be eliminated to obtain $\vec{x}_n^b$. Thus, $\vec{m}^b$ is projected onto the plane normal to $\vec{z}_n^b$ and the resulting vector is normalized:

$$\vec{x}_n^b = \frac{\vec{m}^b + \vec{z}_n^b * ((-\vec{z}_n^b \vec{m}^b)(\vec{z}_n^b \vec{z}_n^b)^{-1})}{\|\vec{m}^b + \vec{z}_n^b * ((-\vec{z}_n^b \vec{m}^b)(\vec{z}_n^b \vec{z}_n^b)^{-1})\|} \tag{2.3}$$

As system n is a right-hand coordinate system, $\vec{y}_n^b$ can be obtained as cross product

$$\vec{y}_n^b = \vec{z}_n^b \times \vec{x}_n^b. \tag{2.4}$$

The attitude of system n relative to system b is then defined by the directional cosine matrix

$$\mathbf{C}_n^b = \begin{pmatrix} \vec{x}_n^b & \vec{y}_n^b & \vec{z}_n^b \end{pmatrix} \tag{2.5}$$

or its quaternion equivalent.

In operation, the quadrocopter will experience accelerations. $\vec{a}^b$ will rarely point directly into the direction of gravity.

We however assume that the quadrocopter's accelerations do not mark the dominant part of $\vec{a}^b$. The quadrocopter can only be accelerated into a certain direction for a limited amount of time due to physical limitations. These periods of acceleration into a certain direction (e.g. for position control) usually last for only a few seconds. Thus, on average, $\vec{a}^b$ will be collinear with $\vec{\gamma}^b$. During normal flight operation, the assumption $\vec{a}^b \approx \vec{\gamma}^b$ should not pose a serious issue.

Anyhow, we need to be aware of a problem with equation (2.3) when $\vec{a}^b$ and $\vec{m}^b$ become collinear. $\vec{x}_n^b$ is undefined in that case. As, in practice, this situation should only arise rarely and for very short times, an error can be caught easily. To solve the problem, $\vec{x}_n^b$ could be chosen arbitrarily or set to the last value that was calculated correctly.

It should be noted that the value of $\vec{x}_n^b$ (and consequently $\vec{y}_n^b$) becomes much more sensitive to noise in $\vec{a}^b$ and $\vec{m}^b$ as the angle between both vectors decreases.

Similarly, equation (2.2) suffers from noise in $\vec{a}^b$ more drastically as the vector approaches $\begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$ and cannot be solved finally (free fall). In this case, the previously calculated

value will be used.

Relative Attitude Calculation

Equipping the gyroscope data $\vec{\omega}_k^b$, sample time τ and an attitude measured at sample k, the following attitude at sample $k+1$ can be calculated. $\vec{\omega}_k^b\tau$ is the orientation vector defining the attitude between $(\vec{q}_b^n)_{k+1}$ and $(\vec{q}_b^n)_k$, thus

$$
\begin{aligned}
(\vec{q}_b^n)_{k+1} &= f((\vec{q}_b^n)_k, \vec{\omega}_k^b\tau) && (2.6)\\
&= (\vec{q}_b^n)_k \bullet \begin{pmatrix} \cos\frac{|\vec{\omega}_k^b|\tau}{2} \\ \frac{\vec{\omega}_k^b}{|\vec{\omega}_k^b|}\sin\frac{|\vec{\omega}_k^b|\tau}{2} \end{pmatrix} && (2.7)
\end{aligned}
$$

For small angular rates $|\vec{\omega}_k^b|$, this can be linearized, yielding

$$
(\vec{q}_b^n)_{k+1} \approx (\vec{q}_b^n)_k \bullet \begin{pmatrix} 1 \\ 1/2\vec{\omega}_k^b\Delta t \end{pmatrix}. \tag{2.8}
$$

2.2.2 Averaging Attitudes

When developing filters for attitude estimation, the need for finding an average of attitudes is encountered. The first question is whether there is any meaningful definition for the average of attitudes and if so, how it can be calculated.

Between two attitudes, there are always two angles around a fixed axis that allows transforming one attitude to the other with one rotation. There is one shortest rotation θ except for $\theta = \pi$, in which case both rotations have the same magnitude. An average attitude is expected to lie somewhere on this shortest rotational path.

The average of two attitudes is be determined, where each attitude is assigned a weighting factor w. Let $\theta_{\overline{1}}$ be the shortest rotation angle between attitude 1 and the average and $\theta_{\overline{2}}$ be the shortest rotation angle between attitude 2 and the average. A reasonable average attitude would minimize the weighted sum

$$
\epsilon = w_1\theta_{\overline{1}} + w_2\theta_{\overline{2}}. \tag{2.9}
$$

Using the orientation vector attitude representation, a naive way to find the weighted average $\vec{\overline{\theta}}$ of $\vec{\theta}_1$ and $\vec{\theta}_2$ would be

$$\overline{\vec{\theta}} = \frac{w_1\vec{\theta}_1 + w_2\vec{\theta}_2}{w_1 + w_2}. \tag{2.10}$$

While this works fine for collinear orientation vectors, the method does not work for general orientation vectors as rotations in $\Re^3$ generally are not additive. For small directional deviations, equation (2.10) may be a sufficient approximation.

To find an average of two attitudes in unit quaternion representation, the same approach as above leads to

$$\overline{\vec{q}^*} = \frac{w_1\vec{q}_1 + w_2\vec{q}_2}{w_1 + w_2}. \tag{2.11}$$

First of all, this does not generally result in a unit quaternion. Second, as known from section 1.3, $\vec{q}_1$ and $-\vec{q}_1$ represent the same attitude, while (2.11) generally results in two different average quaternions representing two different attitudes.

By renormalization, $\overline{\vec{q}^*}$ can be returned to the unit quaternion space. The resulting unit quaternion

$$\overline{\vec{q}} = \frac{\overline{\vec{q}^*}}{|\overline{\vec{q}^*}|} \tag{2.12}$$

still suffers from the second issue. Thus, it cannot be the quaternion we expect for representation of the average attitude. Anyhow, it is shown in [1] that $\overline{\vec{q}}$ actually is quite a good approximation of such an average attitude quaternion.

Averaging quaternions that represent attitudes and averaging attitudes is not the same. Turning back to (2.9), an attitude averaging function is constructed. Let

$$\vec{q}_2 = \vec{q}_1 \bullet \vec{q}_\Delta, \tag{2.13}$$

thus

$$\vec{q}_\Delta = \vec{q}_1^{\,-1} \bullet \vec{q}_2 \tag{2.14}$$

represents the attitude difference between $\vec{q}_1$ and $\vec{q}_2$. The shortest orientation vector representing $\vec{q}_\Delta$ shall be $\vec{\theta}_\Delta$. $\vec{\theta}_\Delta$ can now be scaled by multiplication with a scalar $0 \leq k \leq 1$. The equivalent quaternion representation $\vec{q}(k\theta_\Delta)$ yields an attitude on the shortest rotational path between the identity quaternion and $\vec{q}_\Delta$. By choosing

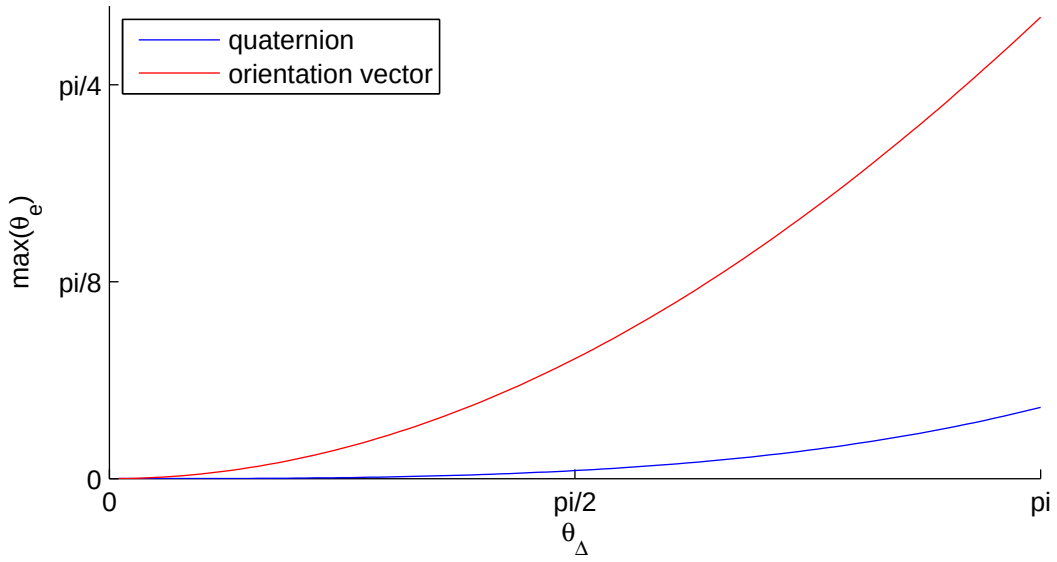

Figure 2.2: Maximum angular error $max(\theta_e)$ using vector space averaging of orientation vector and quaternion attitude representations when averaging attitudes with difference angle of θ_Δ

$$k = \frac{w_2}{w_1 + w_2} \tag{2.15}$$

it is clear that

$$\bar{\vec{q}} = \vec{q}_1 \bullet \vec{q}(k\theta_\Delta) \tag{2.16}$$

is the average attitude that minimizes (2.9).

The resulting errors of different approaches for attitude averaging are shown in figure 2.2.

More information, also regarding averaging of multiple quaternions, can be found in [2].

2.2.3 Attitude Estimator Designs

As this estimator is supposed to estimate the attitude, the system model state vector $\vec{x}$ should certainly somehow represent this attitude.

The attitude changes over time, which can be described via an angular rate. One obvious approach to model this is to include angular rate information into the system state vector. When using some quaternion $\vec{q}$ for attitude representation, the state vector would be

$$\vec{x} = \begin{pmatrix} \vec{q} \\ \vec{\omega} \end{pmatrix} \tag{2.17}$$

and the system state transition would be described by an nonlinear function $\vec{f}$:

$$\vec{x}_{k+1} = \vec{f}(\vec{x}_k) \tag{2.18}$$

In this model, the angular rate information would be updated via the measurement model.

Considering what was discussed before, there are some drawbacks with this concept. It was stated that angular rate measurements can be expected to be afflicted with the least significant systematic measurement errors. Also, as depicted in figure 2.1, angular rate information is already filtered when passed to the attitude filter. When introduced as a system state, angular rate data would be filtered once more and this time, acceleration and magnetic field data would influence the angular rate state estimate.

To avoid this, the following model is proposed. Angular rate information shall be treated as a system input, and the measurement uncertainty (in form of measurement noise) shall be treated as system noise. This gives the system state and system input vectors

$$\vec{x} = \left(\vec{q}\right), \ \vec{u} = \left(\vec{\omega}\right) \tag{2.19}$$

and the system state transition would be described by an nonlinear function $\vec{f}$:

$$\vec{x}_{k+1} = \vec{f}(\vec{x}_k, \vec{u}_k). \tag{2.20}$$

For this system model, a measurement model does not contain the angular rate measurement.

It is assumed that τ_ω is the shortest sample time of the measured signals. It shall also equal the sample time τ_q at which the attitude is to be estimated:

$$\tau_\omega = \tau_{\vec{q}} \tag{2.21}$$

$$\tau_a \geq \tau_{\vec{q}} \tag{2.22}$$

$$\tau_m \geq \tau_{\vec{q}} \tag{2.23}$$

Thus, the rate at which input for equation 2.20 is available equals the state prediction rate, which eliminates a potential need for angular rate interpolation between samples.

Two different measurement models are considered for the system model in equation 2.20. The linear model

$$\left(\tilde{\vec{q}}\right) = \mathbf{H}\left(\vec{q}\right) + \vec{\mu}_q \tag{2.24}$$

requires a preceding step in which the measurement data is transformed: The "measured" attitude $\tilde{\vec{q}}$ is calculated from measured acceleration and magnetic field data as described in section 2.2.1. $\vec{\mu}_q$ denotes the resulting (transformed) noise.

The second measurement model is nonlinear:

$$\begin{pmatrix} \tilde{\vec{a}}^b \\ \tilde{\vec{m}}^b \end{pmatrix} = \vec{h}\left(\vec{q}\right) + \begin{pmatrix} \vec{\mu}_a \\ \vec{\mu}_m \end{pmatrix} \tag{2.25}$$

In order to calculate the estimated measurements $\hat{\vec{a}}^b$ and $\hat{\vec{m}}^b$, information about the expected values of $\vec{a}^n$ and $\vec{m}^n$ is needed inside the measurement function $\vec{h}$.

In the following, five different filters are developed. Three filters using measurement equation (2.24) are called "truncated" because the transformation between the actual measurements and the resulting attitude is not part of the filter. The other two filters's measurement model is described by equation (2.25). The kind of attitude representation is part of a filter's name as well, it is distinguished between an orientation vector filter and quaternion filters. The third distinction relevant for denotation concerns the filter equations being used: Kalman filters are standard extended Kalman filters. When "Kalman" is not part of the filter's name, modifications of the Kalman filter concept are involved.

The combinations Truncated Orientation Vector Kalman Filter (TOVKF), Truncated Quaternion Kalman Filter (TQKF) and Truncated Quaternion Filter (TQF) are examined first. These mainly differ in the way the average attitude is calculated in the (Kalman) update step and are closely related to what was shown in section 2.2.2.

A Quaternion Kalman Filter (QKF) is then derived, which is probably the most straightforward application of the extended Kalman filter concept to the attitude estimation problem as encountered in this thesis.

At last, a Quaternion Filer (QF) is developed. This filter is the least similar to a standard Kalman filter.

Truncated Orientation Vector Kalman Filter

As a first filter, a concept involving a very simple Kalman filter is designed. An orientation vector $\vec{\theta}_b^n = \vec{\theta}(\vec{q}_b^n)$ is used for attitude representation. The nonlinear prediction and measurement steps are performed using quaternions, while the transform between orientation vector and quaternions and vice versa is not explicitly mentioned in the following.

Based on acceleration and magnetic field data, an absolute attitude calculation is performed as described in section 2.2.1. The resulting orientation vector is used as the measurement vector $\tilde{\vec{z}}_k$ of the Kalman filter. The Kalman filter's state vector $\vec{x}_k$ shall approximate the same orientation vector as $\tilde{\vec{z}}_k$. Thus, the Kalman filter's measurement matrix is constant and

$$\mathbf{H} = \mathbf{I_3}. \tag{2.26}$$

The nonlinear prediction step as described in section 2.2.1 is also performed outside the Kalman filter. $\hat{\vec{x}}$ is updated. Due to the fact that no dynamics are modeled in the system equation $\vec{f}$, linearizing it with respect to the state vector yields

$$\begin{aligned} \mathbf{\Phi} &= \frac{\partial \vec{f}}{\partial \vec{x}} \\ &= \mathbf{I_3}. \end{aligned} \tag{2.27}$$

The Kalman filtering results now depend upon the selection of the noise-related elements.

First, the system noise is examined. Noise couples into the system from erroneous angular rate data. As this data is given in the body coordinate system, it needs to be transformed. The noise is transformed by $\mathbf{G}$. In this case, this transformation is simply an orthonormal coordinate transform. Due to that, the noise transition matrix has to be a directional cosine matrix.

In the Kalman equations, $\mathbf{G}$ only appears for updating $\mathbf{P}$ as given by equation (1.17). As $\mathbf{G}$ is a directional cosine matrix, and as our angular data vector has equal noise distributions in all vector components,

$$\mathbf{G}^T = \mathbf{G}^{-1} \tag{2.28}$$

$$\mathbf{Q} = \mathbf{I}_3(\sigma_\omega \tau). \tag{2.29}$$

Due to this,

$$\begin{aligned}\mathbf{GQG}^T &= \mathbf{G}\mathbf{I}_3(\sigma_\omega\tau)\mathbf{G}^{-1} \\ &= (\sigma_\omega\tau)\mathbf{G}\mathbf{G}^{-1} \\ &= (\sigma_\omega\tau)\mathbf{I}_3. \end{aligned} \tag{2.30}$$

Thus, calculating $\mathbf{G}$ is not necessary and we can as well just choose

$$\mathbf{G} = \mathbf{I}_3. \tag{2.31}$$

Second, we turn to the measurement covariance and determine $\mathbf{R}$. The magnetic field data noise and acceleration data noise do not directly represent orientation vector noise. The noise has to be transformed to obtain the angular influence.

Consider a vector $\vec{v}$ that is supposed to define a certain direction. Let $\vec{\Delta v}$ be a (small) deviation of $\vec{v}$.

For $|\vec{\Delta v}| << |\vec{v}|$, and $\vec{\Delta v}$ being orthogonal to $\vec{v}$, the directional error caused by $\vec{\Delta v}$ is

$$\Delta\alpha \approx \frac{|\vec{\Delta v}|}{|\vec{v}|}. \tag{2.32}$$

This also holds for the standard deviation. Thus, the angular influence of the variance of the acceleration and magnetic field data on the measured attitude can be estimated as

$$\sigma^2_{\theta,a} = \frac{\sigma^2_a}{|\vec{a}|^2} \approx \frac{\sigma^2_a}{|\vec{\tilde{a}}|^2} \tag{2.33}$$

$$\sigma^2_{\theta,m} = \frac{\sigma^2_m}{|\vec{m}|^2} \approx \frac{\sigma^2_m}{|\vec{\tilde{m}}|^2}. \tag{2.34}$$

It may be noted that (2.34) is not totally correct due to (2.3).

We will now assume $\sigma_{\tilde{\theta}} = \sigma_{\tilde{\theta},a} = \sigma_{\tilde{\theta},m}$. Thus, all components of the orientation vector are afflicted with the same noise magnitude. Noting Gaussian distribution, this leads to the constant measurement covariance matrix

$$\mathbf{R} = \mathbf{I}_3\sigma_{\tilde{\theta}}. \tag{2.35}$$

Truncated Quaternion Filter

A filter with similar goals as above is designed. The previous concept is altered so that the state vector and the measurement vector are both quaternions.

Similarly as before, we find

$$\begin{aligned} \mathbf{\Phi} &= \mathbf{I}_4 && (2.36) \\ \mathbf{H} &= \mathbf{I}_4. && (2.37) \end{aligned}$$

For $\mathbf{G} = \mathbf{I}_4$, the angular rate data noise μ_ω is injected as system noise via

$$\vec{w} \approx \vec{x} \bullet \begin{pmatrix} 1 \\ 1/2\mu_\omega\tau \end{pmatrix} - \vec{x} \tag{2.38}$$

as seen in (2.7). Similarly, the measurement orientation vector noise $\mu_{\tilde{\theta}}$ can be transformed via

$$\vec{v} \approx \vec{z} \bullet \begin{pmatrix} 1 \\ 1/2\mu_{\tilde{\theta}} \end{pmatrix} - \vec{z}. \tag{2.39}$$

It can now be assumed that the measurement quaternion and the state quaternion are quite similar, thus the noise transformation is similar. In the Kalman equations, $\mathbf{P}$ and $\mathbf{R}$ only play a role for calculating the Kalman gain via equation (1.18). Noting that $\mathbf{\Phi}$, $\mathbf{G}$, $\mathbf{Q}$ and $\mathbf{R}$ are all diagonal matrices, it is obvious that $\mathbf{P}$ and, consequently, $\mathbf{K}$ become diagonal. For k_{ii}, p_{ii} and r_{ii} being diagonal elements of $\mathbf{K}$, $\mathbf{P}$ and $\mathbf{R}$, the Kalman gain can be calculated as

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1} \\ &= \mathbf{P}_k^- \left(\mathbf{P}_k^- + \mathbf{R}_k \right)^{-1} && (2.40) \\ k_{ii} &= p_{ii}/(p_{ii} + r_{ii}). && (2.41) \end{aligned}$$

Any factor coming from the (almost identical) transformations in equations 2.38 and 2.39 would be canceled out here. Thus,

$$\begin{aligned} \mathbf{Q} &= \mathbf{I}_4(\sigma_\omega \tau) && (2.42) \\ \mathbf{R} &= \mathbf{I}_4 \sigma_{\tilde{\theta}} && (2.43) \end{aligned}$$

can be chosen.

Note that after each update step, the state vector needs to be renormalized so that it is a unit quaternion again.

Truncated Quaternion Filter

The goal for designing this filter is to prevent the unit quaternion which represents the attitude to leave its manifold, the unit sphere, because of an update in vector space.

For our matrices from the previous filter, considering the choice of $\mathbf{Q}$ and $\mathbf{R}$, the Kalman gain is

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \\ &= \mathbf{P}_k^- \left(\mathbf{P}_k^- + \mathbf{R}_k\right)^{-1} \\ &= k\mathbf{I}_4 \end{aligned} \tag{2.44}$$

with a scalar k that can be calculated via

$$k = \frac{p_{ii}}{p_{ii} + r_{ii}} \tag{2.45}$$

for any $i = 1, 2, 3, 4$. Noting this, the update equation can be written as

$$\hat{\vec{x}}_k^+ = \hat{\vec{x}}_k^- + k\left(\tilde{\vec{z}}_k - \hat{\vec{x}}_k^-\right) \tag{2.46}$$

The purpose of this equation is to find a weighted average between the measured state and the predicted state. In the standard Kalman filter, this is done by weighted vector averaging. As discussed in section (2.2.2), averaging attitudes and averaging quaternion (vectors) representing attitudes are not the same. By noting that equation (2.46) with additional renormalization actually has the structure of (2.11), it seems legitimate to replace it by an equation with the structure of (2.16). This gives an update equation that actually averages the attitudes represented by the quaternions:

$$\hat{\vec{x}}_k^+ = \hat{\vec{x}}_k^- \bullet \vec{q}\left(k\vec{\theta}_\Delta\right) \tag{2.47}$$

The right-hand function $\vec{q}$ determines the quaternion representing the scaled attitude difference orientation vector. The attitude difference orientation vector is calculated as in equation (2.14):

$$\vec{\theta}_\Delta = \vec{\theta}\left((\hat{\vec{x}}_k^-)^{-1} \bullet \tilde{\vec{z}}_k\right) \tag{2.48}$$

Quaternion Kalman Filter

For the previous filters, we assumed $\sigma_m = \sigma_a$. The prime motivation for designing this filter is the wish to independently assign covariances to magnetic field and acceleration data. An extended Kalman filter is designed to reach this goal.

The same system equation as before is used, giving the same state transition matrix $\mathbf{\Phi} = \mathbf{I}_4$ as before. Anyhow, the nonlinear measurement model from equation (2.25) is now used. Due to this, the observations made in section 2.2.3 concerning noise transmission which lead to simple covariance matrices are useless for this filter.

The system covariance can be approximated as shown in equation (2.38). Noting that quaternion multiplication can be expressed as matrix/vector multiplication, this can be transformed as follows:

$$\begin{aligned}
\mathbf{G}\vec{w} &= \vec{x} \bullet \begin{pmatrix} 1 \\ 1/2\mu_\omega\tau \end{pmatrix} - \vec{x} \\
&= \begin{pmatrix} x_1 & -x_2 & -x_3 & -x_4 \\ x_2 & x_1 & -x_4 & x_3 \\ x_3 & x_4 & x_1 & -x_2 \\ x_4 & -x_3 & x_2 & x_1 \end{pmatrix} \begin{pmatrix} 1 \\ 1/2\mu_\omega\tau \end{pmatrix} - \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} && (2.49) \\
&= \mathbf{G}\left(\mu_\omega\tau\right) && (2.50)
\end{aligned}$$

for a choice of

$$\mathbf{G} = \frac{1}{2} \begin{pmatrix} -x_2 & -x_3 & -x_4 \\ x_1 & -x_4 & x_3 \\ x_4 & x_1 & -x_2 \\ -x_3 & x_2 & x_1 \end{pmatrix}. \tag{2.51}$$

In this case, the system covariance matrix becomes

$$\mathbf{Q} = \mathbf{I}_3(\sigma_\omega^2\tau). \tag{2.52}$$

The measurement equation predicts the measurement vector based on the current state estimate by transforming the known, ideal acceleration and magnetic field vectors

$$\vec{a}^n = \begin{pmatrix} a_1^n \\ a_2^n \\ a_3^n \end{pmatrix}, \ \vec{m}^n = \begin{pmatrix} m_1^n \\ m_2^n \\ m_3^n \end{pmatrix} \tag{2.53}$$

to the body coordinate frame using the state quaternion.

Linearizing the measurement equation considering $a_1^n = a_2^n = m_2^n = 0$ gives

$$\begin{aligned} \mathbf{H} &= \frac{\partial \vec{h}}{\partial \vec{x}} \\ &= 2 \begin{pmatrix} -a_3^n x_3 & a_3^n x_4 & -a_3^n x_1 & a_3^n x_2 \\ a_3^n x_2 & a_3^n x_1 & a_3^n x_4 & a_3^n x_3 \\ a_3^n x_1 & -a_3^n x_2 & -a_3^n x_3 & a_3^n x_4 \\ m_1^n x_1 - m_3^n x_3 & m_1^n x_2 + m_3^n x_4 & -m_1^n x_3 - m_3^n x_1 & m_3^n x_2 - m_1^n x_4 \\ m_3^n x_2 - m_1^n x_4 & m_1^n x_3 + m_3^n x_1 & m_1^n x_2 + m_3^n x_4 & m_3^n x_3 - m_1^n x_1 \\ m_1^n x_3 + m_3^n x_1 & m_1^n x_4 - m_3^n x_2 & m_1^n x_1 - m_3^n x_3 & m_1^n x_2 + m_3^n x_4 \end{pmatrix} \end{aligned} \tag{2.54}$$

As opposed to the previous filters, splitting the measurement is possible, which allows for $\tau_m \neq \tau_a$. By selecting the according rows of $\mathbf{H}$ and $\mathbf{R}$, a Kalman correction step solely based on either magnetometer or accelerometer data is possible.

Note that, for this filter setup, the lengths of the acceleration and magnetic field vectors do have an influence in the update equation. For attitude determination, there is no valuable information to be obtained from the vectors' lengths. Thus, it can be considered useful to adjust the lengths of $\tilde{\vec{a}}^b$ and $\tilde{\vec{m}}^b$ to fit the lengths of $\hat{\vec{a}}^b$ and $\hat{\vec{m}}^b$ before executing the Kalman update step. This is done in the test implementation of this filter.

Furthermore, due to the vector space Kalman update step, state quaternion renormalization is necessary for this filter.

Quaternion Filter

A filter concept is developed that combines features of the TQF and QKF. This filter includes an update equation in unit quaternion space. Also, it allows different sample rates τ_m, τ_a and different variances σ_m^2, σ_a^2 for magnetic field and acceleration data.

Again, the attitude shall be represented by $\vec{q}_b^n$ as state vector.

It was shown previously how two scalar covariances assigned to two quaternions could be used to average these quaternions. In order to allow for different measurement variances, the concept of the TQF needs to be extended. The observation that, when the noise

is propagated as in the case of the TQF and TQKF, one (scalar) variance measure for the (four component) quaternion is sufficient is used. The estimated state quaternion is now assigned two different estimation variances. One shall represent the certainty of the quaternion's z^b-axis estimation, the other one shall represent the certainty of the quaternion's x^b-axis estimation. It is assumed that these are sufficiently independent.

Basically, two different independent system equations and measurement equations are set up. The actual state prediction is performed as for the previous filters. The state transition matrix used for covariance transition is not the linearized state prediction equation, though.

$$\mathbf{\Phi} = \mathbf{I}_2 \tag{2.55}$$

$$\mathbf{G} = \mathbf{I}_2 \tag{2.56}$$

$$\mathbf{Q} = \begin{pmatrix} q_a & 0 \\ 0 & q_m \end{pmatrix} = \begin{pmatrix} \tau\sigma_\omega^2 & 0 \\ 0 & \tau\sigma_\omega^2 \end{pmatrix} \tag{2.57}$$

$$\mathbf{R} = \begin{pmatrix} r_a & 0 \\ 0 & r_m \end{pmatrix} = \begin{pmatrix} \sigma_{\tilde{\vec{\theta}},a}^2 & 0 \\ 0 & \sigma_{\tilde{\vec{\theta}},m}^2 \end{pmatrix} \tag{2.58}$$

$$\mathbf{H} = \mathbf{I}_2 \tag{2.59}$$

$$\mathbf{K} = \begin{pmatrix} k_a & 0 \\ 0 & k_m \end{pmatrix} \tag{2.60}$$

These matrices now allow for variance propagation and gain calculation as usual. Anyhow, the gains k_a and k_m obviously cannot be used as usual. Instead, they serve as gains for two consecutive update steps, featuring a update equation similar to that of the TQF. As opposed to the TQF, this filter does, however, not calculate a complete attitude quaternion from the measurement data. Instead, the orientation vector defined by the angular difference between the estimated and measured acceleration and magnetic field vectors are used for attitude correction.

The first update step, involving correction based on acceleration data, is

$$\vec{\theta}_a = \frac{\tilde{\vec{a}} \times \hat{\vec{a}}}{|\tilde{\vec{a}} \times \hat{\vec{a}}|} \arccos(\tilde{\vec{a}}^T \hat{\vec{a}}) \tag{2.61}$$

$$\hat{\vec{x}}_k^{+a} = \hat{\vec{x}}_k^{-} \bullet \vec{q}(k_a \vec{\theta}_a). \tag{2.62}$$

Following this, correction based on magnetic field data is performed analogically:

$$\vec{\theta}_m = \frac{\tilde{\vec{m}} \times \hat{\vec{m}}}{|\tilde{\vec{m}} \times \hat{\vec{m}}|} \arccos(\tilde{\vec{m}}^T \hat{\vec{m}}) \tag{2.63}$$

$$\hat{\vec{x}}_k^{+m} = \hat{\vec{x}}_k^{+a} \bullet \vec{q}(k_m \vec{\theta}_m). \tag{2.64}$$

The order of both steps is chosen arbitrarily and of course, similar to the QKF, independent updates based on either acceleration or magnetic field data are possible.

Filter Initialization

All filter equations covered are recursive. This poses the question of how to choose the initial values $\vec{x}_0$ and $\mathbf{P}_0$.

One idea is to wait for absolute attitude measurements to be available. The filter would remain inactive until magnetometer and accelerometer information are available. In that case, $\vec{x}_0$ would be set according to the measurements and

$$\mathbf{P}_0 = \mathbf{H}_0^{-1} \mathbf{R} \mathbf{H}_0. \tag{2.65}$$

This does, however, result in the disadvantage an undefined filter output during the period of inactivity.

Another approach is to choose a value which could be expected at startup. For the quaternion filters, a good guess would be

$$\vec{x}_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^T \tag{2.66}$$

as the quadrocopter would probably start with at an attitude with the upside upwards. $\mathbf{P}_0$ would be set to represent a high uncertainty.

Practically, the choice does barely have any influence on the filter performance at all. An estimation difference would only last for a few samples. The latter approach is chosen as filter inactivity is more difficult to handle for the control loop.

2.3 Altitude Estimation

The quadrocopter's flight altitude can be estimated combining sensor information. A rough direct altitude measurement is possible using barometric altimeter data, but an actual barometric altimeter is quite inaccurate. Altitude is calculated based on pressure,

taking into consideration that pressure decreases when moving upwards inside the earth's atmosphere. However, there are major generally unknown parameters that influence the pressure at a certain location.

The barometric altimeter's measurements can be supported with information about the quadrocopter's vertical acceleration. This improves the resolution of the barometric altitude estimation and decreases the effect of pressure disturbances. Besides the (strap down) accelerometer data, an attitude estimation is necessary to allow for coordinate transformation so the vertical acceleration component can be obtained. Using equation (1.11),

$$\begin{pmatrix} 0 \\ a_x^n \\ a_y^n \\ a_z^n \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{a}^n \end{pmatrix} = \vec{q}_b^n \bullet \begin{pmatrix} 0 \\ \vec{a}^b \end{pmatrix} \bullet \vec{q}_n^b \tag{2.67}$$

transforms the data of the strap down accelerometers into the navigation coordinate system. As this transformation is performed before using a Kalman filter, no nonlinear part is needed inside the Kalman filter equation.

Assuming $\tau_h > \tau_a$, a linear Kalman filter will be designed to fuse the barometric altimeter measurement and the vertical acceleration data.

2.4 System Model

We will use the following linear discrete time system model:

$$\begin{pmatrix} s \\ v \\ \tilde{g} \end{pmatrix}_{k+1} = \begin{pmatrix} 1 & \tau_a & -0.5 * \tau_a^2 \\ 0 & 1 & -\tau_a \\ 0 & 0 & 1 - \tau_X \end{pmatrix} \begin{pmatrix} s \\ v \\ \tilde{g} \end{pmatrix}_k + \begin{pmatrix} 0.5 * \tau_a^2 \\ \tau_a \\ \tau_X \end{pmatrix} a_z \tag{2.68}$$

s and v denote the altitude and vertical speed, respectively. $\tilde{g}$ is the estimated gravity constant calculated as a mean value of a_z. $\tilde{g}$ is estimated in order to cope with accelerometer bias and and local gravitational acceleration being different from the value expected due to standard gravity g_0.

The system noise results from noise of the input, thus

$$\mathbf{G}_k = \mathbf{B}_k \tag{2.69}$$
$$\mathbf{Q}_k = \sigma_a^2. \tag{2.70}$$

The linear measurement model is

$$\tilde{h}_k = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \hat{\vec{x}}_k + \mu_{h,k}. \tag{2.71}$$

Hence, he measurement covariance is the scalar

$$\mathbf{R}_k = \left(\sigma_h^2 \right). \tag{2.72}$$

Chapter 3

Controller Design

A controller is now designed that shall allow for altitude and attitude reference tracking. Similarly to the estimator, the controller is divided into separate modules.

As seen in figure 3.1, an altitude controller is responsible for tracking a reference altitude $\breve{s}$ and a reference vertical velocity $\breve{v}$. The altitude controller outputs the overall vertical thrust T_z^n that is necessary for this. The attitude controller processes this quantity and tries to track a reference attitude $\breve{\vec{q}}_b^n$ and reference angular rates $\breve{\vec{\omega}}^b$. It outputs a vector

$$\vec{T} = \begin{pmatrix} T_{front} & T_{right} & T_{rear} & T_{left} \end{pmatrix}^T \tag{3.1}$$

the components of which represent the individual engines' thrust.

Besides better clarity, the modular design also has the advantage that the altitude controller can easily be deactivated to allow for manual altitude control. For example, this may sometimes be desired due to environmental pressure fluctuations leading to altitude estimation error.

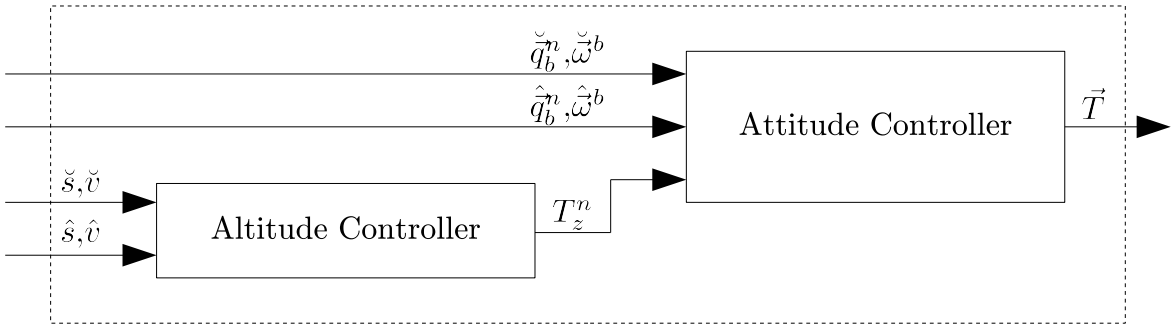

Figure 3.1: Controller division

3.1 Altitude Controller

A PID controller with c_p, c_i and c_d as factors of the proportional, integral and derivative path is proposed.

$$T_z^n = c_p\left(\hat{s} - \breve{s}\right) + c_i \int_0^\tau \left(\hat{s} - \breve{s}\right) dt + c_d\left(\hat{v} - \breve{v}\right) + T_{z,0}^n \tag{3.2}$$

$T_{z,0}^n$ is the vertical thrust required for flight at a constant height.

Expedient values for c_p, c_d and c_i shall now be found. Therefore, the controller is optimized using the Linear Quadratic Regulator (LQR) technique described in [9].

A state e representing the steady state altitude error is introduced. The system is modeled as

$$\dot{\vec{x}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \vec{x} + \begin{pmatrix} 0 \\ 0 \\ 1/m \end{pmatrix} u \tag{3.3}$$

$$\vec{y} = \mathbf{I}_3 \vec{x} \tag{3.4}$$

with the state vector $x = \begin{pmatrix} e & s & v \end{pmatrix}^T$, the input $u = T_z^n - T_{z,0}^n$ and the quadrocopter's mass m.

Choosing the LQR weighting matrices as

$$\mathbf{Q} = \begin{pmatrix} Q_e & 0 & 0 \\ 0 & Q_s & 0 \\ 0 & 0 & Q_v \end{pmatrix} \tag{3.5}$$

$$\mathbf{R} = \mathbf{I}_1 \tag{3.6}$$

with tuning parameters Q_e, Q_s and Q_v results in the optimal feedback vector

$$\vec{F} = \begin{pmatrix} c_i \\ c_p \\ c_d \end{pmatrix} \tag{3.7}$$

that can, for example, be obtained using the Matlab command `lqr`.

3.2 Attitude Controller

The altitude controller determines the necessary vertical thrust. Due to the quadrocopter's underactuation shown in section 1.1, vertical thrust cannot be generated independently. In order to obtain the desired vertical thrust, the four engines actually need to deliver an average thrust of

$$\overline{\vec{T}} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^T \frac{T_z^n}{4\cos(\alpha)} \tag{3.8}$$

which depends upon α, the quadrocopter's z_b axis angular deviation from the vertical position caused by rotation around the x_n and y_n axes. This angle is obtained as the length of the attitude's orientation vector after elimination of the z-component:

$$\alpha = \left| \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \vec{\theta}\left(\hat{\vec{q}}_b^n\right) \right| \tag{3.9}$$

The attitude deviation in form of an orientation vector is

$$\vec{\theta}_\Delta^b = \vec{\theta}\left(\hat{\vec{q}}_b^n \bullet (\breve{\vec{q}}_b^n)^{-1}\right). \tag{3.10}$$

A PID controller is proposed. The engines' thrust shall be obtained by

$$\vec{T} = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & -1 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{pmatrix} \left(\mathbf{C}_p \vec{\theta}_\Delta^b + \mathbf{C}_i \int_0^\tau \vec{\theta}_\Delta^b dt + \mathbf{C}_d \left(\hat{\vec{\omega}}^b - \breve{\vec{\omega}}^b \right) \right) + \overline{\vec{T}} \tag{3.11}$$

where the leftmost matrix is given physically by the position and rotational direction of the engines' rotors (also see figure 1.2). Obviously, due to this matrix, equation (3.11) does not change the overall thrust - the sums of the components of $\overline{\vec{T}}$ and $\vec{T}$ are identical.

The matrices $\mathbf{C}_p$, $\mathbf{C}_i$ and $\mathbf{C}_d$ contain the PID parameters, these can be expected to be diagonal matrices. Different weights for the influence of roll, pitch and yaw deviations can be altered here, too. All controller constants can be optimized using the LQR technique when the quadrocopter's inertia tensor and the relation between the rotors' thrust and moment are known. Anyhow, designing an optimal controller is beyond the scope of this thesis.

Instead, the controller constants will be chosen based on an existing, hand-tuned and well tested roll-pitch-yaw PID controller so that both controllers produce the same output for

small angular deviations.

Chapter 4

Application to Quadrocopter

This chapter deals with the application of the previously developed estimator and controller to a physical quadrocopter. A look is taken at the quadrocopter's hardware, focusing on the sensors. Based on this, the parameters needed for estimator and controller application are determined.

The estimator from chapter 2 and the controller from chapter 3 were integrated into the quadrocopter software framework. The QF attitude filter was chosen for implementation. Implementation details shall not be focused here.

4.1 Hardware overview

The TUHH Quadrocopter of the Institute for Reliability of Systems, developed by Dipl. Ing. Jonas Witt, is used. This is an strongly modified version of the commercially available *Reely QuadroCopter 450 ARF*. All components except for the frame and the motor controllers are replaced.

A *PICO820* ITX board serves as the central processing platform. It is equipped with an *Intel ATOM* processor of the *Z500* Series, which serves enough processing power so that algorithm efficiency is no primary concern.

In the center of the quadrocopter, an *Analog Devices ADIS16405* triaxial inertial sensor with magnetometer is mounted. It provides the accelerometric, gyroscopic and magnetometric data used by the estimator. The device contains an embedded temperature sensor and offers automatic and manual bias correction controls. More information can be found in [7].

A *VTI Technologies SCP1000* barometer described in [6] is the basis for the barometric

Figure 4.1: A photo of the TUHH quadrocopter

altimeter.

The overall weight of the ready-to-fly quadrocopter amounts to

$$m_{Quadrocopter} \approx 1.1\,[\mathrm{kg}] \tag{4.1}$$

4.2 Accelerometer

The *ADIS16405* micro-machined electromechanical (MEMS) accelerometer provides for internal bias calibration. A long-term measurement of the calibrated sensor is shown in figure 4.2. For more information on MEMS sensors, refer to [10].

The accelerometer covariance matrix is calculated as

$$\tilde{\Xi}_a = \begin{pmatrix} 0.0039 & 0.0000 & 0.0001 \\ 0.0000 & 0.0035 & -0.0003 \\ 0.0001 & -0.0003 & 0.0036 \end{pmatrix} \left[\mathrm{m}^2/\mathrm{s}^4\right]. \tag{4.2}$$

An approximation of this matrix is used, by which the accelerometer variance σ_a^2 is defined:

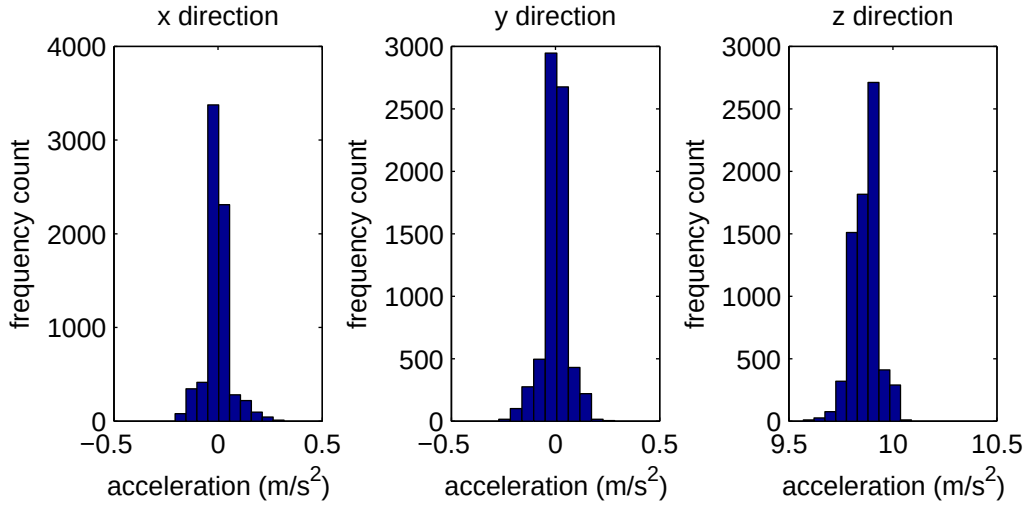

Figure 4.2: Histogram of calibrated accelerometer data

$$\tilde{\boldsymbol{\Xi}}_a \approx \sigma_a^2 \mathbf{I}_3 \tag{4.3}$$

$$\sigma_a^2 = 3.5 \cdot 10^{-3} \left[\mathrm{m}^2/\mathrm{s}^4\right] \tag{4.4}$$

Accelerometer data is available with a sample time of

$$\tau_a = 4[\mathrm{ms}]. \tag{4.5}$$

4.3 Gyroscope

The MEMS gyroscope also offers internal bias correction. Figure 4.3 shows a histogram of the calibrated gyroscope. The sensor is analyzed similar to the accelerometer, giving

$$\tilde{\boldsymbol{\Xi}}_\omega = 10^{-4} \begin{pmatrix} 0.1892 & 0.0009 & -0.0071 \\ 0.0009 & 0.2069 & 0.0097 \\ -0.0071 & 0.0097 & 0.2135 \end{pmatrix} \left[\mathrm{rad}^2/\mathrm{s}^2\right] \tag{4.6}$$

$$\approx \sigma_\omega^2 \mathbf{I}_3 \tag{4.7}$$

$$\sigma_\omega^2 = 2 \cdot 10^{-5} \left[\mathrm{rad}^2/\mathrm{s}^2\right]. \tag{4.8}$$

Gyroscope data is available with a sample time of

$$\tau_\omega = 4[\mathrm{ms}]. \tag{4.9}$$

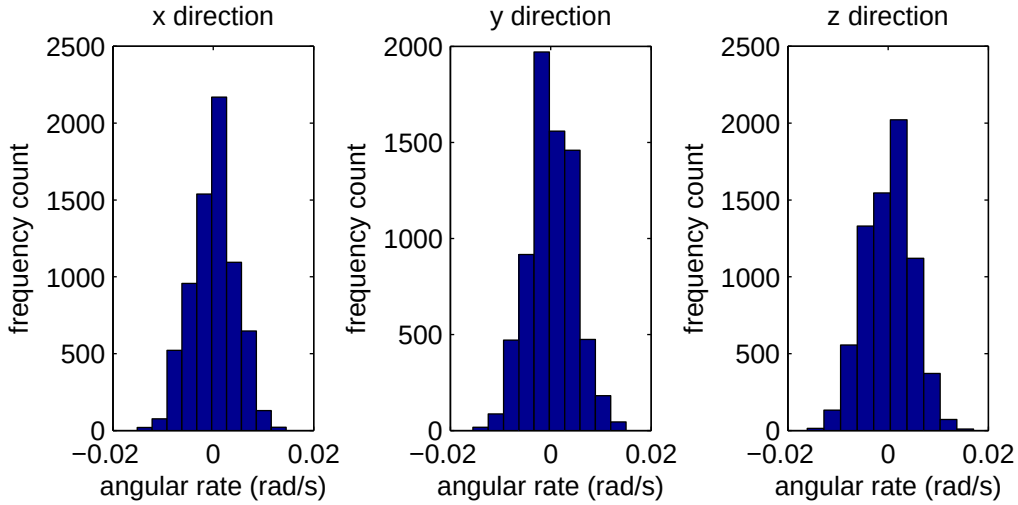

Figure 4.3: Histogram of calibrated gyroscope data

4.4 Magnetometers

The magnetometer is intended to measure the earth's magnetic field, which has a field strength of 0.5 to 0.6 gauss (G). The field vector's direction depends upon the latitude, it is vertical at the earth's poles and horizontal at the equator. Given sufficient distance from the poles, the field vector's horizontal component indicates the north facing direction.

4.4.1 Sensor Calibration

There are two types of interferences often encountered with magnetometers the effect of which can be annihilated by calibration:

- Hard iron interference
- Soft iron interference

Hard iron interference is caused by materials in the sensor's vicinity that act like magnets. Any field caused by magnetic material and the earth's magnetic field are superimposed.

Hard iron interference can be compensated for under the premise that the source of interference has a stationary position relative to the sensor. In that case, the interference simply has the consequence of a constant value being added to each of the three sensor direction components. These constants can easily be measured by sensor rotation and then substracted from the measured values.

Soft iron interference is caused by ferromagnetic materials. Ferromagnetic materials do not induce magnetic fields themselves, but they distort existing ones. Because of this, the distortion caused depends on the sensor orientation.

It is still possible to compensate for soft iron interference, though this is not as simple as for hard iron interference effects.

4.4.2 Utilizability Analysis

On the quadrocopter platform, there are several potential sources of magnetic disturbances, like

- Cables
- Battery
- Computer board
- DC converter
- Motors
- Motor controllers.

Especially the electrical components that naturally produce magnetic fields are matters of concern. First, if the sensor is driven into saturation by strong static magnetic fields, the previously discussed compensation methods can not be applied. Second, there are components that can be expected to produce unpredictably fluctuating fields. This poses the questions whether sensor calibration is possible and whether magnetic data can be employed successfully at all.

Figure 4.4 shows the magnetic data acquired by the on board sensor with all electrical components being enabled. Although the magnetic field is disturbed and the disturbance varies over time, satisfactory directional information can be obtained after simple hard iron calibration.

A magnetometer data histogram for the sensor is shown in figure 4.5. The covariance matrix is

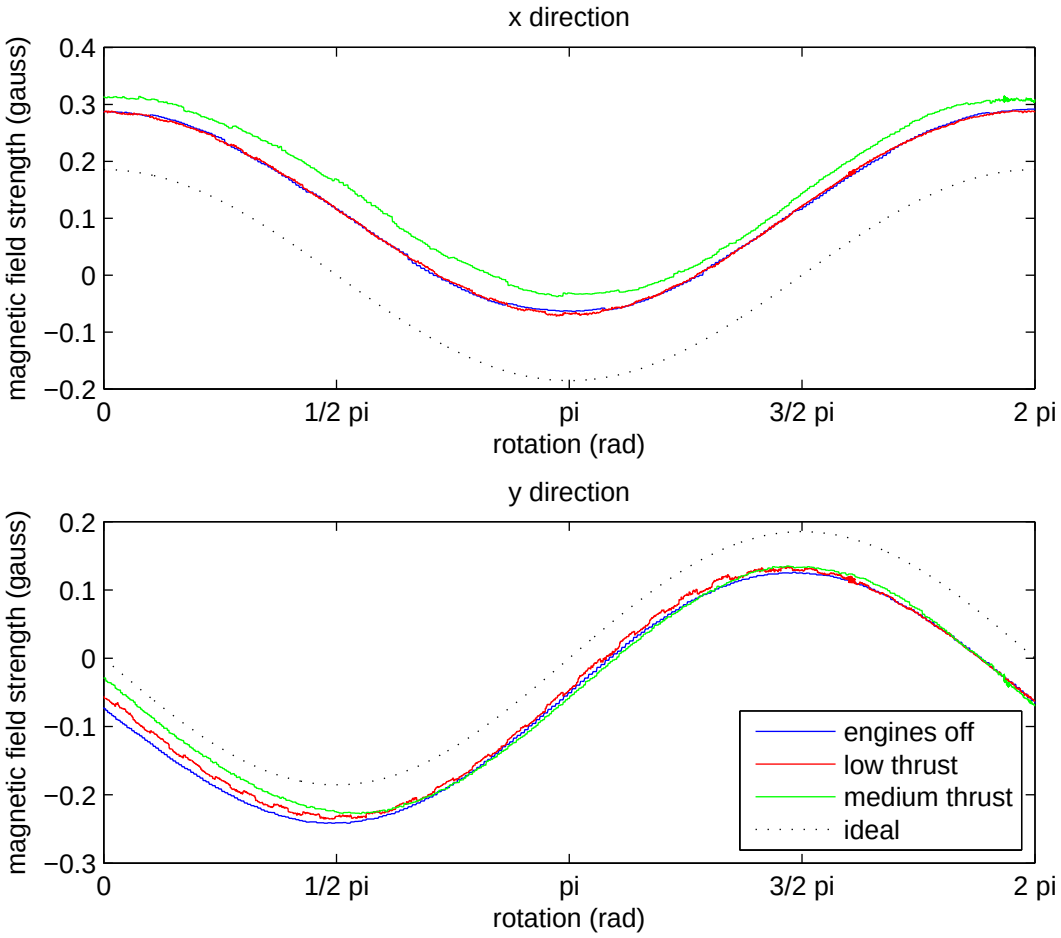

Figure 4.4: Magnetometer data of a rotation around the z-axis

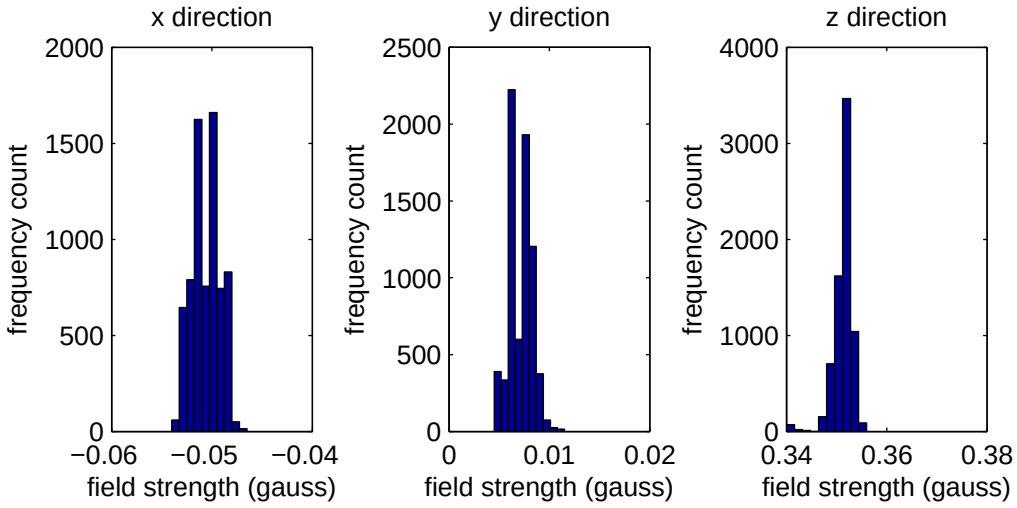

Figure 4.5: Histogram of magnetometer data

$$\begin{aligned}
\tilde{\mathbf{\Xi}}_m &= 10^{-5} \begin{pmatrix} 0.1899 & -0.1164 & -0.0032 \\ -0.1164 & 0.1381 & 0.0017 \\ -0.0032 & 0.0017 & 0.3301 \end{pmatrix} \left[\mathrm{G}^2\right] && (4.10) \\
&\approx \sigma_m^2 \mathbf{I}_3 && (4.11) \\
\sigma_m^2 &= 2 \cdot 10^{-6} \left[\mathrm{G}^2\right]. && (4.12)
\end{aligned}$$

The magnetometer sample time time is

$$\tau_m = 20[\mathrm{ms}]. \tag{4.13}$$

4.5 Barometric Altimeter

The barometer is to be used for measuring the altitude. This is possible as atmospheric pressure p and altitude s are related, as modeled by the barometric formula (for details, see [5]).

The approximation

$$p(s) \approx 1013{,}25 \left(1 - \frac{0{,}0065 \cdot s}{288{,}15[\mathrm{m}]}\right)^{5{,}255} \mathrm{hPa} \tag{4.14}$$

is often used for practical applications.

Hence, the altitude at a given pressure can be approximated as

$$s(p) \approx 44330[\mathrm{m}] \cdot \left(1 - \left(\frac{p}{1013.25[\mathrm{hPa}]}\right)^{0.19}\right). \tag{4.15}$$

Atmospheric pressure at a certain location is not constant but can change quite quickly due to weather conditions and other influences. Figure illustrates this by showing altimeter data recorded over a period of several minutes.

In order to cope with pressure fluctuations over time, a reference ground barometer can be used. With such a reference system, long-term accuracy of the altitude measurement can probably be improved significantly. We will, however, work without a reference measurement station in this paper. Thus, the altitude estimation will not be long-term reliable.

For calculating the barometric altimeter's variance, a short period of seemingly low environmental pressure changes is chosen. The data histogram for this period is shown in figure 4.7. The measured barometric altimeter's variance amounts to

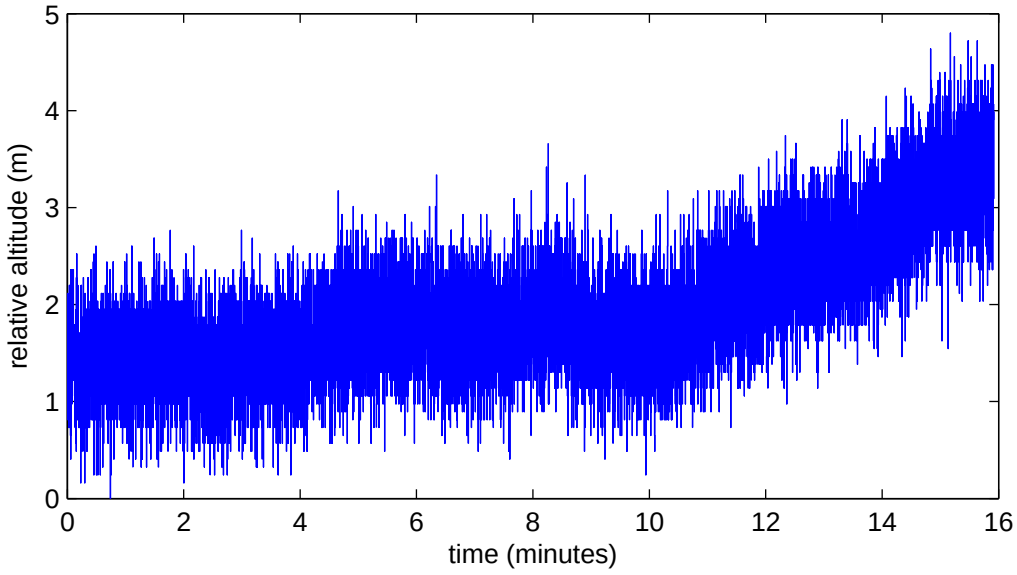

Figure 4.6: Longterm altimeter measurement

$$\sigma_s \approx 0.15[\mathrm{m}^2]. \tag{4.16}$$

The barometric altimeter's sample time is

$$\tau_s = 40[\mathrm{ms}]. \tag{4.17}$$

4.6 Controller Constants

4.6.1 Altitude Controller Constants

The altitude PID controller constants are now determined for the quadrocopter.

The altitude controller's priority shall be to reduce the vertical velocity error, followed by reducing the altitude error. This gives an idea for setting the weighting factors from equation (3.6). They are chosen to be

$$Q_e = 4 \tag{4.18}$$
$$Q_s = 2 \tag{4.19}$$
$$Q_v = 1. \tag{4.20}$$

Considering the quadrocopter's weight from equation (4.1), the LQR-optimized controller

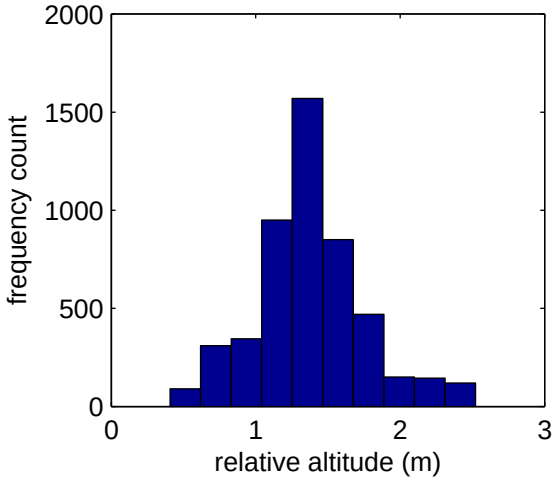

Figure 4.7: Histogram of altimeter data

constants become

$$c_p = -2.9751 \tag{4.21}$$

$$c_i = -1 \tag{4.22}$$

$$c_d = -3.4257. \tag{4.23}$$

4.6.2 Attitude Controller Constants

The already existing RPY attitude controller can be transformed to a form similar to (3.11). As for small angular deviations, RPY angles and the orientation vector's components are identical, the optimal controller constants for both controllers are identical. As the RPY controller does, however, not output physically meaningful thrust values but a relative thrust measure, the linear relation between these derived in [4] is used, giving a factor of 0.03.

In the end, this results in the following PID tuning matrices:

$$\mathbf{C}_p = 0.03 \begin{pmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{pmatrix} \tag{4.24}$$

$$\mathbf{C}_i = 0.03 \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.2 \end{pmatrix} \tag{4.25}$$

$$\mathbf{C}_d = 0.03 \begin{pmatrix} 0.075 & 0 & 0 \\ 0 & 0.075 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.26}$$

Chapter 5

Results

In this chapter, tests are conducted in order to verify the estimators developed in chapter 2 and the controllers developed in chapter 3. A conclusion on the work done is drawn. Finally, an outlook on possible future work related to this thesis is presented.

5.1 Tests

The following tests are numerical simulations of the estimators' and controllers' behavior. For all these tests, the parameters (sample times, variances, controller constants) derived in in the previous chapter are used.

5.1.1 Altitude Estimator

The altitude estimator is tested using measurement data with ideal white-noise disturbance. The original altitude data is defined by the vertical acceleration. This acceleration is a sinus oscillation with decreasing period time.

The simulation results are shown in figure 5.1.

It can be seen that the velocity estimate takes some time to stabilize at the beginning. This is because the initial state estimation variance is high, so the estimate is more sensitive to measurement noise at the beginning. Besides this, nothing remarkable can be seen in the simulation results. The estimate follows the correct state closely and the filter works as expected.

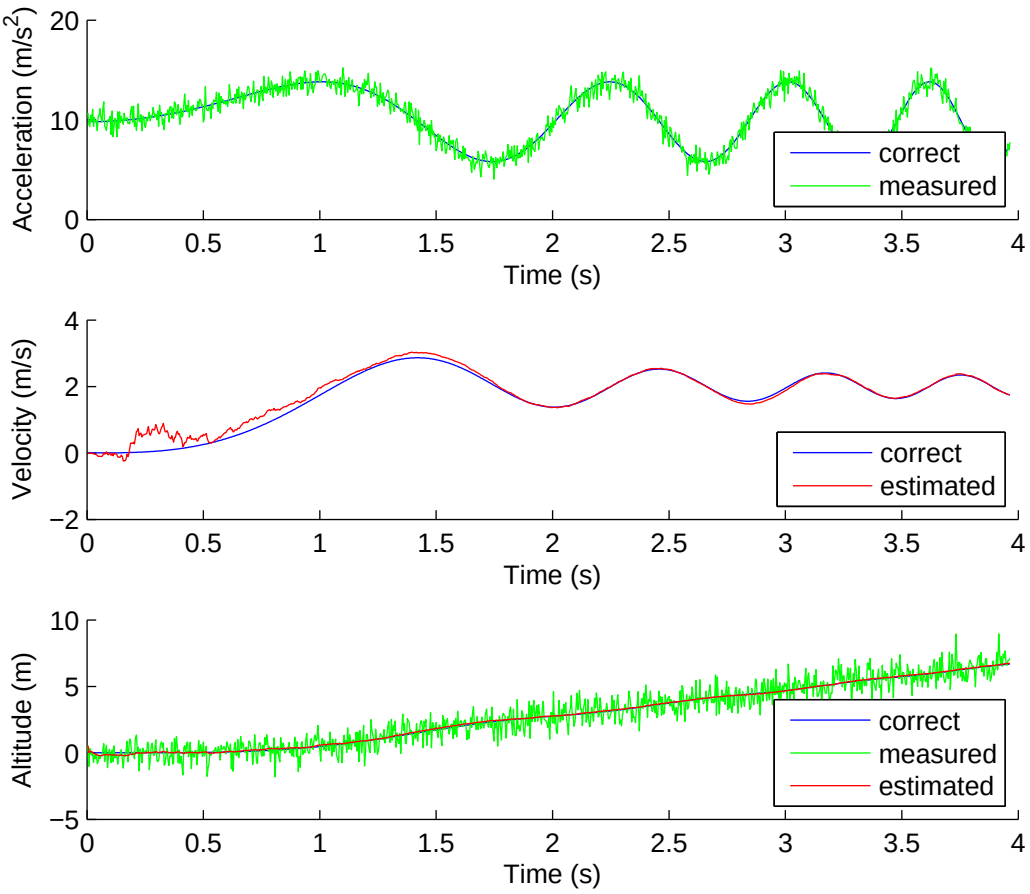

Figure 5.1: Altitude estimator response to simulated measurement data

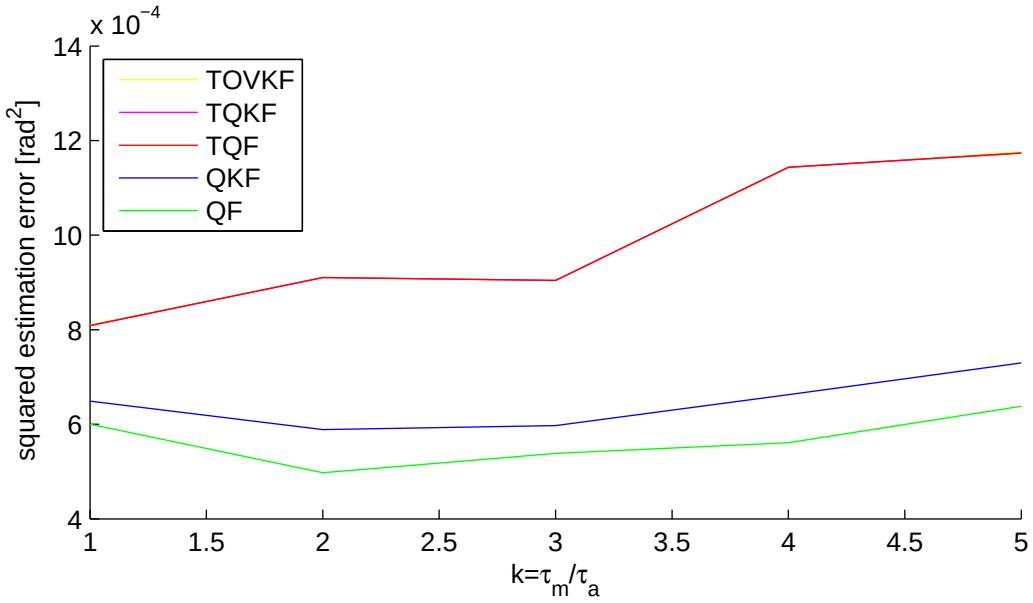

Figure 5.2: Attitude estimation error for different magnetometer sample times

5.1.2 Attitude Estimators

To simulate the attitude estimators' performance, a random attitude rotation trajectory for a period of 2 minutes is generated.

All estimators show similar behavior for the parameters of interest. The estimation standard deviation turns out to be greater than the estimation difference between different filters. A diagram of estimation error over time is not helpful for filter comparison. Instead, figure 5.2 shows the variance (the average sum of squared errors) of the different attitude estimates in terms of magnitude of the difference attitude orientation vector between the original attitudes and the estimated attitudes. A parameter k is altered, by which τ_m is varied: The first simulations are performed for $\tau_m = \tau_a$, while the last (for $k = 5$) are performed using the real value of $\tau_m = 5\tau_a$.

The first notability concerns the results of the truncated filters. The TOVKF, TQKF and TQF create estimates so similar that no difference can be observed in the plot, the curve describing the TQF covers those of the other two truncated filters. The maximum difference in the plotted values amounts to $2 \cdot 10^{-6}$ between the TOVKF and TQKF and $4 \cdot 10^{-9}$ between the TQKF and TQF. This can be explained by considering figure 2.2: The low variances and low sample times result in rather small deviations between predicted and estimated attitudes.

A second observation is that the truncated filters generally have a worse performance than the other two, which is due to the fact that one major assumption - the equality of

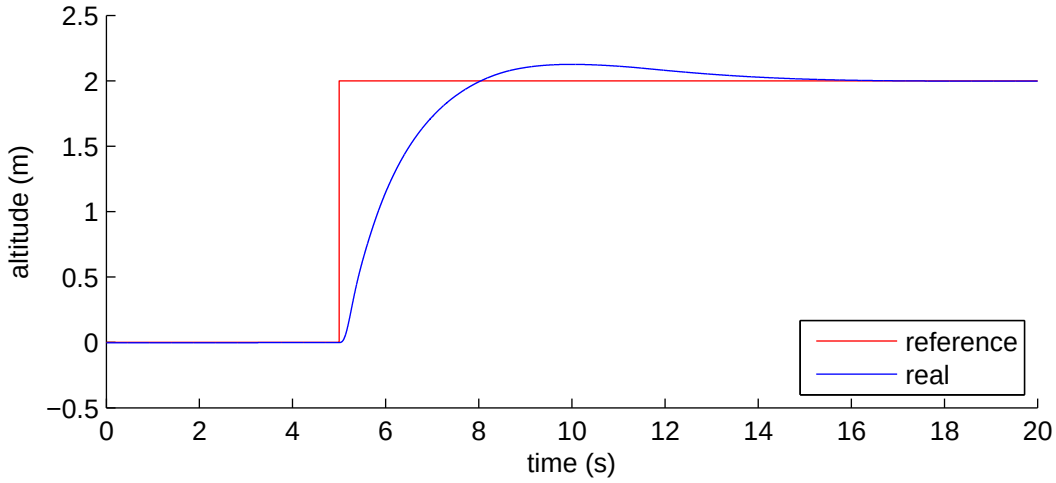

Figure 5.3: Simulated step response of the altitude controller

σ_a^2 and σ_m^2 - is not valid. Thus, the filters work with suboptimal measurement covariance matrices. Also, it is obvious that the truncated filters are negatively influenced by an increase of the magnetometer's sample time most significantly. This is what was expected from the filter design, as equality of τ_a and τ_m was another assumption needed for the truncated measurement equation.

Finally, it is notable that the QF outperforms the other filters in all simulations performed.

5.1.3 Altitude Controller

The altitude controller performance is simulated using a system model. Therefor, the verified nonlinear quadrocopter model developed in [4] is used.

As reference signal, a two meter step after five seconds is given. The simulation result can be seen in figure 5.3.

The quadrocopter model travels to the reference altitude within three seconds. A slight altitude overshoot can be seen. As the simulation model and the system model used for controller optimization are not identical, this is not the best possible response. For example, the altitude controller does not model the engine dynamics. Anyhow, the controller works as expected.

5.1.4 Attitude Controller

For simulating the attitude controller behavior, again, the quadrocopter model developed in [4] is used.

A reference attitude trajectory is supposed to be followed. The attitude trajectory is represented in form of orientation vector components θ_1, θ_2 and θ_3. Derived from this trajectory, reference angular rates ω_1, ω_2, ω_3 are also given.

The simulation results are shown in figure 5.4.

The initial difference in ω_x ω_3 is obviously caused by a reference signal step. After 55 seconds, a sudden change of ω_x has a slight influence on ω_y. Slight overshoot is seen in all orientation vector components. The reference angular rates are tracked less precisely, overshoot and minor oscillations can be seen.

The attitude controller works fine.

5.1.5 Flight Tests

Several flight test with the TUHH quadrocopter were performed. However, as no reference attitude and altitude measurement systems with sufficient accuracy were set up during the work on this thesis, no meaningful evaluation of recorded data is possible. Therefor, only a qualitative description is given here.

The estimator and controller both work well and allow for comfortable handling of the device. Attitude estimation and control work totally reliable. Altitude estimation and control suffer from pressure disturbances, making it difficult to operate at a proximity of less than 1.5 meters to horizontal barriers.

5.2 Summary

In this thesis, a modular state estimator is developed. One module is responsible for attitude determination. Five different filter concepts for this module are designed and tested. Simulation results suggest that the filter denoted QF is the best. This filter is based on the Kalman filter technique but is modified for seamlessly working with unit quaternions. It is chosen for application on the quadrocopter.

The second estimator module is used for altitude determination. The filter inside this module is a simple linear Kalman filter.

At last, an LQR-optimized altitude controller and a quaternion based attitude controller are shown.

The estimators and controllers are tested via simulation and successfully deployed on the TUHH quadrocopter.

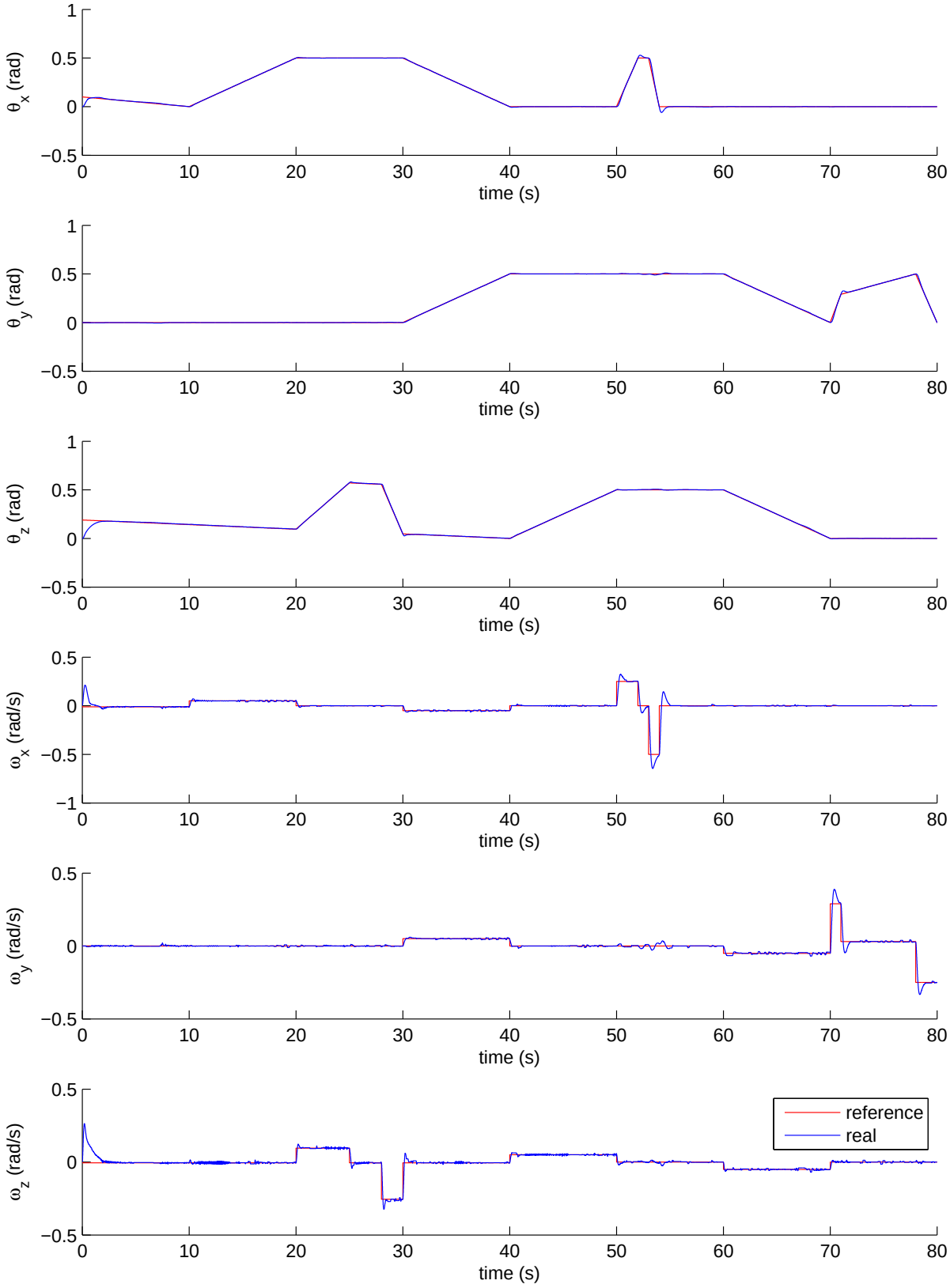

Figure 5.4: Attitude controller reference tracking simulation

5.3 Future Work

The work on this thesis leads to some ideas for future work which shall be pointed out here.

There are some ways to improve the state estimator. A logical step would be to extend estimation to feature state prediction based on the controller output. A very promising concept for improving the altitude estimator would be the introduction of a reference barometer to allow for compensation of environmental pressure fluctuations.

Both estimator parts might profit from a dynamic adaption of measurement covariances. Depending of the quadrocopter's movement and attitude, conclusions are possible concerning the certainty of information coming from the sensors. For example, during periods of rapid movement, the confidence in accelerometer information for attitude determination could be lowered.

A deeper analysis of the quaternion filter equations from section 2.2.3 is necessary. The filter should be classified and compared to other filter concepts used for quaternion filtering like the Sigma Point Kalman filter.

The attitude controller can be optimized based on a quadrocopter model, e.g. using the LQR technique.

Finally, current and future quadrocopter developments need to be evaluated and compared. For this purpose, a precise reference altitude and attitude measurement system has to be found and set up.

Nomenclature

$\vec{a}$	Accelerations vector
$\mathbf{C}$	Directional cosine matrix
$\vec{F}$	Vector of the four motors' forces
g_0	Standard gravity, $g_0 = 9.80665 m/s^2$
$\vec{g}$	Local acceleration due to gravity
$\mathbf{G}$	System noise input matrix
$\vec{h}$	Observation function
$\mathbf{H}$	Observation matrix
$\mathbf{I}$	Identity matrix
k	Time step or component of Kalman gain matrix
$\mathbf{K}$	Kalman gain matrix
$\vec{m}$	Magnetic field vector
$\vec{M}$	Vector of the four motor moments
$\mathbf{P}$	State estimation covariance matrix
$\vec{q}$	Quaternion (usually unit length)
$\mathbf{Q}$	System covariance matrix
$\mathbf{R}$	Measurement covariance matrix
s	Altitude
$\vec{T}$	Vector of the four motors' thrusts
$\vec{u}$	Input vector
v	Vertical velocity
$\vec{x}$	State vector
$\vec{z}$	Measurement vector

Greek letters

$\mathbf{\Phi}$	Discrete time system matrix (state transition matrix)
$\mathbf{\Gamma}$	Discrete time input matrix

σ^2	Covariance (squared standard deviation)
$\vec{\omega}$	Angular rate
$\vec{\theta}$	Orientation vector
θ	Length of an orientation vector

Operators

$\bullet$	Quaternion multiplication
∂	Partial derivative operator

Bibliography

[1] C. Gramkow. On averaging rotations. *Journal of Mathematical Imaging and Vision*, 15(1):7–16, 2001.

[2] F.L. Markley, Y. Cheng, J.L. Crassidis, and Y. Oshman. Averaging Quaternions. *Journal of Guidance Control and Dynamics*, 30(4):1193, 2007.

[3] P.S. Maybeck. *Stochastic models, estimation and control*. Academic Pr, 1982.

[4] Sven Pfeiffer. Identifikation eines nichtlinearen physikalischen quadrokopter-modells. Thesis, August 2009.

[5] VTI Technologies, P.O. Box 27, FI-01621 Vantaa, Finland. *SCP1000 Pressure Sensor as Barometer and Altimeter*, 2009.

[6] VTI Technologies, P.O. Box 27, FI-01621 Vantaa, Finland. *SCP1000 Series Absolute Pressure Sensor Data Sheet*, 2009.

[7] One Technology Way. *Triaxial Inertial Sensor with Magnetometer ADIS16400/ADIS16405*. Analog Devices, Inc., P.O. Box 9106, Norwood, MA 02062-9106, U.S.A., b edition, 2009.

[8] Jan Wendel. *Integrierte Navigationssysteme*. Oldenbourg Wissensch.Vlg, 3 2007.

[9] Herbert Werner. Control systems 2. Lecture notes, 2006.

[10] O.J. Woodman. An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 2007.