Pitch Angle Estimation of Two Wheel Platform Using EKF and Madgwick Filter

Joakim Lilja

January 6, 2016

Abstract

Having a two wheel platform being able to balance requires an accurate estimate of it's pitch angle, w.r.t. the vertical axis. This report describes the Madgwick and extended Kalman filter algorithm and their implementations as estimators for this problem using data from an Inertial Measurement Unit. The Madgwick filter uses a quaternion representation of the orientation and a gradient descent method to estimate the angle in a computationally efficient way. The extended Kalman filter exploit the assumption that the noise is gaussian and using Bayes filter to estimate the mean of the state. The performance of the filters are then analysed based on the parameter settings and a final comparison shows that both are almost equally accurate(both static RMS error $< 0.2°$) and where the Madgwick filter is 25% faster.

Contents

1 Introduction

Estimating the orientation is an important task in many fields such as aerospace, robotics or human motion analysis. In this report two approaches are discussed and compared in estimating the pitch of a two wheel platform, namely the Madgwick filter and the extended Kalman filter using the gyroscope and accelerometer of an inertial measurement unit, *IMU*.

To balance a two wheel platform an accurate estimation of the pitch is needed for the controller. An IMU provides a gyroscope, an accelerometer and sometimes a compass - or magnetometer. The gyroscope measures angular velocity, the accelerometer measures acceleration and the magnetometer measures the magnetic field, all along three axes. To get the angles, the gyroscope measurements can be integrated assuming the initial condition is known. However, since the measurements are subject to noise, the estimated angle might not be accurate enough and the estimate might drift away due to accumulated error in the integration. This is a well known problem in estimates based on gyroscope measurements. To overcome this, the measurements of the accelerometer, and sometimes the magnetometer, are incorporated to get a better result. When incorporating the accelerometer, the effect of the gravity is exploited. If the platform's(or vehicle's) own acceleration can be neglected in comparison to the gravitational pull, the orientation can partly be obtained(since the gravity only has components in the z-direction the yaw cannot be obtained). To get the full orientation the magnetometer measurements are needed as well, and should be compared with the known magnetic field of the earth. Since we, in this project, don't need the yaw of the platform, the magnetometer measurements will be discarded.

The Madgwick filter has, since published in 2010, become a popular[1] choice in attitude estimation due to it's high accuracy and low computational load. We will use this filter to fuse the estimations from the gyroscope and the accelerometer.

The Kalman filter[6] and it's extended form where the motion model and measurement model is linearized has been a popular choice when estimating the orientation, see for example [4]. Here the state will be composed of the pitch angle and it's angular velocity.

This report intends to be partly a literature study where the basics of both algorithms are described, and partly a study in the behavior and accuracy of the filters based on their parameter settings. The platform is tilted to a known angle and the filters estimations are calculated. These are then compared to the ground truth. Both filters are shown to perform well with a static RMS$< 0.2°$, as expected, and the Madgwick filter is shown to be 25% faster.

The report is structured as follows. Section 2 presents the Madgwick filter along with a brief description of quaternions. Section 3 describes the extended Kalman filter and derives the model for this specific case. These two sections lays out the fundament for the method section, 4, where the experimental setup and implementation is presented followed by the results in section 5. Finally, conclusions are drawn in section 6.

[1] 293 citations according to Google Scholar, `https://scholar.google.se`.

2 Madgwick Filter

In [7] an approach for fusing the gyroscope, accelerometer and magnetometer data was presented which will be described in this section. Since the filter uses a quaternion representation the first subsection is dedicated to this. Then the filter algorithm is presented together with a brief description of each step.

2.1 Quaternion

A quaternion is a four dimensional complex number and can be seen to consist of a real part and an imaginary part, such that $\boldsymbol{q} = \boldsymbol{q}(a, \boldsymbol{v})$, where $a \in \mathbb{R}$, $\boldsymbol{v} = b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k} \in \mathbb{R}^3$ and $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$ [3]. To understand their application, consider first a complex number

$$z \in \mathbb{C}.$$

Using Euler's formula we can write this in polar form

$$z = re^{i\theta}.$$

Multiplying two complex numbers $z_1 = r_1 e^{i\theta}$, $z_2 = r_2 e^{i\phi}$ yields

$$z_1 z_2 = r_1 r_2 e^{i(\theta+\phi)}$$

That is, multiply their respective length but add their angles. Having a complex number of unit length thus correspond to a pure rotation. Complex numbers are hence good for representing rotations and orientations in a two dimensional space. Quaternions of unit length are, in analogy, suitable for representing orientations in three dimensional space. A unit quaternion can be written as

$$\boldsymbol{q} = e^{\boldsymbol{u}\phi} = \cos\phi + \boldsymbol{u}\sin\phi,$$

where $\boldsymbol{u} = u_i\boldsymbol{i} + u_j\boldsymbol{j} + u_k\boldsymbol{k}$ and $|\boldsymbol{u}| = 1$. The following theorem[2] is of interest.

Theorem 1. *If $\boldsymbol{u} \in \mathbb{R}^3$ is a vector of unit length and $\boldsymbol{v}$ is any vector $\in \mathbb{R}^3$ then*

$$e^{\boldsymbol{u}\phi}\boldsymbol{v}e^{-\boldsymbol{u}\phi} \tag{1}$$

results in rotating $\boldsymbol{v}$ around the axis with the direction of $\boldsymbol{u}$, by 2ϕ.

Following the notation of [5] used in[7] we denote the quaternion describing the orientation of frame B relative to frame A as ${}^A_B\hat{\boldsymbol{q}}$. Noting that an arbitrary orientation of B relative A can be described by a rotation around an axis ${}^A\hat{\boldsymbol{r}}$, defined in frame A, by θ. This is illustrated in Fig. 1. Note that, since we want to express a pure rotation, and no scaling, we want the quaternion to be of unit length. Letting r_x, r_y and r_z denote the components of the unit vector ${}^A\hat{\boldsymbol{r}}$ yields, by theorem 1,

$${}^A_B\hat{\boldsymbol{q}} = \begin{bmatrix}\cos\frac{\theta}{2} & -r_x\sin\frac{\theta}{2} & -r_y\sin\frac{\theta}{2} & -r_z\sin\frac{\theta}{2}\end{bmatrix}.$$

The quaternion conjugate, denoted by $\boldsymbol{q}^*$ is useful to swap frame of reference. Thus

$${}^A_B\hat{\boldsymbol{q}}^* = {}^B_A\hat{\boldsymbol{q}} = \begin{bmatrix}\cos\frac{\theta}{2} & r_x\sin\frac{\theta}{2} & r_y\sin\frac{\theta}{2} & r_z\sin\frac{\theta}{2}\end{bmatrix}.$$

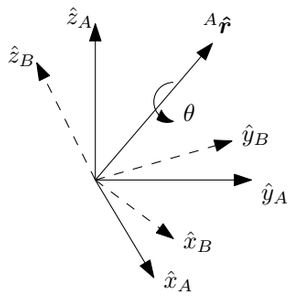

Figure 1: Orientation of frame B relative to frame A expressed as a rotation around an axis ${}^A\hat{\boldsymbol{r}}$.

The quaternion product is defined as

$$\begin{aligned}\boldsymbol{a}_1 \otimes \hat{\boldsymbol{b}}_2 &= [a_1 \quad a_2 \quad a_3 \quad a_4] \otimes [b_1 \quad b_2 \quad b_3 \quad b_4] \\ &= \begin{bmatrix} a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4 \\ a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3 \\ a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2 \\ a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1 \end{bmatrix}. \end{aligned} \tag{2}$$

Having two orientations ${}^A_B\hat{\boldsymbol{q}}$ and ${}^B_C\hat{\boldsymbol{q}}$ we can compute the compound orientation as

$${}^A_C\hat{\boldsymbol{q}} = {}^A_B\hat{\boldsymbol{q}} \otimes {}^B_C\hat{\boldsymbol{q}}.$$

By using the quaternion product the rotation of a vector in frame A, ${}^A\boldsymbol{v}$ to frame B, ${}^B\boldsymbol{v}$(that is, the vector is the same with respect to it's frame) can be expressed as

$${}^B\boldsymbol{v} = {}^A_B\hat{\boldsymbol{q}} \otimes {}^A\boldsymbol{v} \otimes {}^A_B\hat{\boldsymbol{q}}^*.$$

Note that, the vectors need to be four dimensional, thus the first index is set to 0, such that $\boldsymbol{v} = [0 \quad v_x \quad v_y \quad v_z]$.

2.2 Filter Algorithm

The update step of the filter can be seen in Table 1. First the orientation is estimated given the gyroscope measurements, ω_t. That's line 1-2. Then the orientation is estimated using the measurements from the accelerometer. This is line 3. Line 4 fuses both estimations into the final orientation estimate. Each step will be described below.

Algorithm Madgwick Filter
update_madgwick($\boldsymbol{q}_{est}[t-1]$, $\boldsymbol{\omega}[t]$, $\boldsymbol{a}[t]$), β, Δt)
1. $\dot{\boldsymbol{q}}[t] = \frac{1}{2}\hat{\boldsymbol{q}}_{est}[t-1] \otimes \boldsymbol{\omega}[t]$
2. $\boldsymbol{q}_\omega[t] = \hat{\boldsymbol{q}}_{est}[t-1] + \dot{\boldsymbol{q}}[t]\Delta t$
3. $\boldsymbol{q}_\nabla[t] = -\frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|}$
4. $\boldsymbol{q}_{est}[t] = \beta \Delta t \boldsymbol{q}_\nabla[t] + \boldsymbol{q}_\omega[t]$
return $\frac{\boldsymbol{q}_{est}[t]}{\|\boldsymbol{q}_{est}[t]\|}$

Table 1: Madgwick Filter Algorithm for orientation estimation of a body in 3-D. The super- and subscripts are omitted for clarity. The quaternions are always in sensor frame S relative to earth frame E and the measurements $\boldsymbol{\omega}[t]$ of the gyroscope and $\boldsymbol{a}[t]$ of the accelerometer in the sensor frame S.

2.2.1 Orientation Estimation From Gyroscope

Let

$${}^S\boldsymbol{\omega} = [0 \quad \omega_x \quad \omega_y \quad \omega_z]$$

be the quaternion representing the angular velocities around the axis in the frame of the sensor, S. Then the quaternion derivative in sensor frame S relative to the earth frame E is

$${}^S_E\dot{\boldsymbol{q}} = \tfrac{1}{2}\,{}^S_E\hat{\boldsymbol{q}} \otimes {}^S\boldsymbol{\omega} \quad [7] \tag{3}$$

Assuming the previous quaternion estimate , ${}^S_E\hat{\boldsymbol{q}}_{est}[t-1]$ is known, the quaternion estimate of current time t is given by numerically integrating Eq. 3

$$\begin{aligned}{}^S_E\dot{\boldsymbol{q}}[t] &= \tfrac{1}{2}\,{}^S_E\hat{\boldsymbol{q}}_{est}[t-1] \otimes {}^S\boldsymbol{\omega}[t] \\ {}^S_E\boldsymbol{q}_\omega[t] &= {}^S_E\hat{\boldsymbol{q}}_{est}[t-1] + {}^S_E\dot{\boldsymbol{q}}[t]\Delta t.\end{aligned}$$

Following the notation in [7], the subscript ω stands for the estimation due to the gyroscope measurement. This corresponds to line 1 and 2 in Table 1.

2.2.2 Orientation Estimation from Accelerometer

Assuming a field direction(such as gravity or magnetic), ${}^E\hat{\boldsymbol{d}}$, is known in the earth reference frame, the measured direction in the sensor frame, ${}^S\hat{\boldsymbol{s}}$, allows one to calculate the orientation. However, the solution will not be unique, as all solutions rotated around an axis parallel to the field are valid. To overcome this, the measurements from the magnetometer and accelerometer are combined to yield a unique solution [7]. Since the pitch angle is the only orientation of interest in this report, this is not a problem here and the magnetometer is not used.

To calculate the orientation an optimization problem is formulated. The task is to find ${}^S_E\hat{\boldsymbol{q}}$ such that

$${}^S_E\hat{\boldsymbol{q}}_\nabla = \min_{{}^S_E\hat{\boldsymbol{q}} \in \mathbb{R}^4} \boldsymbol{f}\left({}^S_E\hat{\boldsymbol{q}}, {}^E\hat{\boldsymbol{d}}, {}^S\hat{\boldsymbol{s}}\right) \tag{4}$$

$$f\left({}^S_E\hat{\boldsymbol{q}}, {}^E\hat{\boldsymbol{d}}, {}^S\hat{\boldsymbol{s}}\right) = {}^S_E\hat{\boldsymbol{q}}^* \otimes {}^E\hat{\boldsymbol{d}} \otimes {}^S_E\hat{\boldsymbol{q}} - {}^S\hat{\boldsymbol{s}}, \tag{5}$$

where

$$\begin{aligned}{}^S_E\hat{\boldsymbol{q}} &= [q_1 \quad q_2 \quad q_3 \quad q_4] \\ {}^E\hat{\boldsymbol{d}} &= [0 \quad d_x \quad d_y \quad d_z] \\ {}^E\hat{\boldsymbol{s}} &= [0 \quad s_x \quad s_y \quad s_z].\end{aligned}$$

This is numerically solved using the gradient descent method, see for example [1]. Given an initial guess ${}^S_E\hat{\boldsymbol{q}}[0] = \hat{\boldsymbol{q}}_0$, and step size μ the $(k+1)$th guess is calculated as

$${}^S_E\boldsymbol{q}_{k+1} = {}^S_E\,\hat{\boldsymbol{q}}_k - \mu \frac{\nabla \boldsymbol{f}\left({}^S_E\hat{\boldsymbol{q}}_k, {}^E\hat{\boldsymbol{d}}, {}^S\hat{\boldsymbol{s}}\right)}{\|\nabla \boldsymbol{f}\left({}^S_E\hat{\boldsymbol{q}}_k, {}^E\hat{\boldsymbol{d}}, {}^S\hat{\boldsymbol{s}}\right)\|}, \qquad k = 0, 1, 2, ..., n.$$

Let ${}^E\hat{\boldsymbol{d}} = {}^E\hat{\boldsymbol{g}} = [0 \quad 0 \quad 0 \quad 1]$ and ${}^S\hat{\boldsymbol{s}} = {}^S\hat{\boldsymbol{a}} = [0 \quad a_x \quad a_y \quad a_z]$ denote the normalized acceleration vector due to gravity in the earth frame and the normalized measured acceleration in the sensor frame respectively. Then, using Eq. (2) and (5),

$$\boldsymbol{f}\left({}^S_E\hat{\boldsymbol{q}}, \boldsymbol{g}, \boldsymbol{a}\right) = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - a_x \\ 2(q_1q_2 + q_3q_4) - a_y \\ 2(\frac{1}{2} - q_2^2 - q_3^2) - a_z \end{bmatrix}, \tag{6}$$

with corresponding Jacobian

$$\boldsymbol{J}\left({}^S_E\hat{\boldsymbol{q}}\right) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix}. \tag{7}$$

The gradient of $\boldsymbol{f}$ can then be calculated as

$$\nabla \boldsymbol{f}({}^S_E\hat{\boldsymbol{q}}, \boldsymbol{a}) = \boldsymbol{J}\left({}^S_E\hat{\boldsymbol{q}}\right)^T \boldsymbol{f}\left({}^S_E\hat{\boldsymbol{q}}, \boldsymbol{a}\right).$$

Furthermore, assuming the convergence rate of the gradient descent algorithm is greater than the physical rate of change, it's acceptable[7] letting the filter

only compute one iteration each time step. This is to reduce the computational load which would greatly increase if multiple iterations were carried out each time step. Thus

$$ {}^{S}_{E}\boldsymbol{q}_{\nabla}[t] = {}^{S}_{E}\hat{\boldsymbol{q}}_{est}[t-1] - \mu[t]\frac{\nabla \boldsymbol{f}}{||\nabla \boldsymbol{f}||} \approx -\mu[t]\frac{\nabla \boldsymbol{f}}{||\nabla \boldsymbol{f}||}. \quad (8) $$

This corresponds to line 3 in Table 1. The last step follows from the fact that the term ${}^{S}_{E}\hat{\boldsymbol{q}}_{est}[t-1]$ becomes negligible as described in section 2.2.3 below. Note also that the step size $\mu[t]$ has been incorporated in the weight of the accelerometer estimation in the fusion step as will be described below.

2.2.3 Fusing Estimations

The idea is now to fuse the estimations from the gyroscope and the accelerometer.

$$ \boldsymbol{q}_{est}[t] = \gamma[t]\boldsymbol{q}_{\nabla}[t] + (1-\gamma[t])\boldsymbol{q}_{\omega}[t] \quad (9) $$

where $0 < \gamma[t] < 1$. As discussed in [7], an optimal value of γ is that which ensures the weighted convergence rate of ${}^{S}_{E}\boldsymbol{q}_{\nabla}[t]$ is equal to the weighted divergence rate of $\boldsymbol{q}_{\omega}[t]$. With $\mu[t]/\Delta t$ being the convergence rate of ${}^{S}_{E}\boldsymbol{q}_{\nabla}[t]$ and β being the divergence rate of $\boldsymbol{q}_{\omega}[t]$ representing the noise in the gyroscope sensor, we have

$$ (1-\gamma[t])\beta = \gamma[t]\frac{\mu[t]}{\Delta t} \Longrightarrow \quad (10) $$

$$ \gamma[t] = \frac{\beta}{\frac{\mu[t]}{\Delta t} + \beta} = \frac{\beta \Delta t}{\mu[t] + \beta \Delta t}. \quad (11) $$

Since $0 < \gamma[t] < 1$ and we've assumed that the step size $\mu[t]$ is chosen so that the convergence rate $\mu[t]/\Delta t$ is larger than the physical rate of change, which is $\propto \beta$, there is no upper bound on $\mu[t]$ and Eq. (11) simplifies to, assuming $\mu[t] >> \beta\Delta t$,

$$ \gamma[t] \approx \frac{\beta \Delta t}{\mu[t]} \quad (12) $$

and

$$ (1-\gamma[t]) \approx 1. $$

The step size $\mu[t]$ cancels out when combining Eq. (8) and (12), and Eq. (9) simplifies to

$$ \boldsymbol{q}_{est}[t] = \beta \Delta t \boldsymbol{q}_{\nabla}[t] + \boldsymbol{q}_{\omega}[t], $$

which corresponds to line 4 in Table 1.

The parameter β is expressed as a quaternion derivative representing the mean zero noise of the gyroscope measurements, $\tilde{\omega}$. It can thus be defined as[7]

$$ \left\| \frac{1}{2}\hat{\boldsymbol{q}} \otimes [0 \quad \tilde{\omega} \quad \tilde{\omega} \quad \tilde{\omega}] \right\| = \sqrt{\frac{3}{4}}\tilde{\omega}, $$

where $\hat{\boldsymbol{q}}$ is any unit quaternion.

3 Extended Kalman Filter

In this section the Extended Kalman Filter algorithm is presented and described. First, some concepts are introduced and explained.

3.1 Terminology

The Kalman Filter is part of the family of recursive state estimators, where the probability distribution of the true state, the belief, is represented as gaussians. In this application the state is the vertical angle and corresponding angular velocity.

The filter implements the Bayes filter algorithm where the state, x_t evolves over time influenced by *control* and *previous state*, x_{t-1}. The belief is then updated by incorporating measurements, z_t, using the probability distribution of the measurement z_t *given* the state x_t. A graphical representation of this temporal process can be seen in Fig. 2. For more details and derivation of Bayes filter see for example [8].

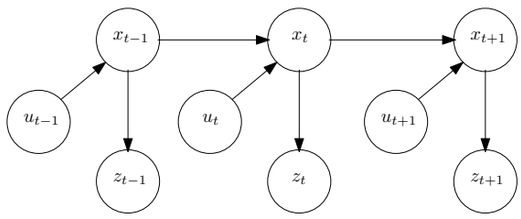

Figure 2: Graphical representation of the temporal model underlying the Baye's filter algorithm. Such model is also known as a *hidden Markov model* or *dynamic Bayes network*.

3.2 EKF Algorithm

One iteration of the extended Kalman filter is presented in Table 2. The filter takes the previous state mean $\mu[t-1]$, previous covariance matrix $\Sigma[t-1]$, current control $u[t]$ and current measurements $z[t]$ as inputs and return the new estimated mean and covariance at time t. Line 1-2 is referred to as the *prediction step*, where the state is predicted using the state transition function. Line 3 calculates the *Kalman gain* used for incorporating the measurements. Line 4-5 is referred to as the *measurement update step* where the belief is updated after incorporating the measurements.

Algorithm Extended Kalman Filter
update_ekf$(\mu[t-1], \Sigma[t-1], u[t], z[t])$
1. $\bar{\mu}[t] = g(u[t], \mu[t-1])$
2. $\bar{\Sigma}[t] = G[t]\Sigma[t-1]G[t]^T + R[t]$
3. $K[t] = \bar{\Sigma}[t]H^T[t](H[t]\bar{\Sigma}[t]H^T[t] + Q[t])^{-1}$
4. $\mu[t] = \bar{\mu}[t] + K[t](z[t] - h(\bar{\mu}[t]))$
5. $\Sigma[t] = (I - K[t]H[t])\bar{\Sigma}[t]$
return $\mu[t],\ \Sigma[t]$

Table 2: The extended Kalman filter algorithm[8].

The filter assumes that the state transition and measurements can be described by *nonlinear* functions g and h respectively with added gaussian noise, ε and δ,

$$ x[t] = g(u[t], x[t-1]) + \varepsilon[t] $$
$$ z[t] = h(x[t]) + \delta[t]. $$

These nonlinear functions are then linearized around their respective mean which is represented in the filter as the jacobians $G[t]$ and $H[t]$ of g and h respectively. By doing this the calculated belief remains gaussian which is the key point of the Kalman filter. For more details and derivation of the Kalman and Extended Kalman filter see for example [8].

3.3 EKF for Vertical Angle Estimation

The state is given by

$$\boldsymbol{x}[t] = \begin{bmatrix} \theta[t] \\ \omega[t] \end{bmatrix}$$

where θ is the angle from the vertical axis and $\omega = \dot{\theta}$. The transition of θ is governed by

$$\theta[t] = \theta[t-1] + \omega[t-1]\Delta t + \varepsilon_\theta[t].$$

Here ε_θ is gaussian zero mean noise accounting for sampling errors etc.

The control is set to zero, $u[t] = 0$, and we can model the change in angular velocity as only gaussian noise, ε_ω, since the platform is tilted by hand

$$\omega[t] = \omega[t-1] + \varepsilon_\omega[t].$$

This can be written in matrix form as

$$\begin{aligned} \boldsymbol{x}[t] &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \boldsymbol{x}[t-1] + \begin{bmatrix} \varepsilon_\theta[t] \\ \varepsilon_\omega[t] \end{bmatrix} \\ &= A\boldsymbol{x}[t-1] + \boldsymbol{\varepsilon}[t]. \end{aligned}$$

Since the state transition is linear the filter relaxes to a linear Kalman filter in the prediction step where $g(\boldsymbol{x}) = A\boldsymbol{x}$ and $G = A$. The measurement model is

$$\boldsymbol{h}[t] = \begin{bmatrix} \omega[t] \\ a_y[t] \\ a_z[t] \end{bmatrix} = \begin{bmatrix} \omega[t] \\ g\sin\theta \\ g\cos\theta \end{bmatrix},$$

see Fig. 3. The jacobian of $\boldsymbol{h}$ evaluated at $\bar{\boldsymbol{\mu}} = [\bar{\theta},\ \bar{\omega}]$ is

$$H[t] = \begin{bmatrix} 0 & 1 \\ g\cos\bar{\theta}[t] & 0 \\ -g\sin\bar{\theta}[t] & 0 \end{bmatrix}.$$

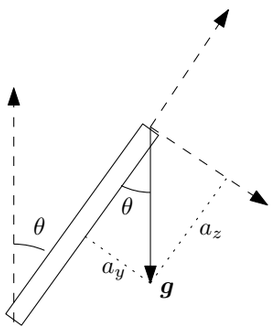

Figure 3: Schematic of the balancing platform. Assuming the acceleration due to gravity is much larger than the acceleration of the platform, the measurements $a_y = g\sin\theta$ and $a_z = g\cos\theta$.

4 Filter Evaluation

4.1 Experimental Setup

To get a better understanding of the filter characteristics the parameters of the filters are varied. The optimal parameters are then estimated using Matlab's *fminsearch*. In Tab. 3 the parameters can be seen. β corresponds to the weighting between the gyroscope and accelerometer measurements in the Madgwick filter, see section 2.2.3.

A large β corresponds to a large weight on the accelerometer measurements, but is also related to large step size in the gradient descent method. A too large β would therefore lead to numerical instability and a large variance in the estimation. A too small β would however weight the gyroscope measurements too high, making the filter prone to accumulated error in the integrated measurement. Based on this, we vary the β between 2 and 0.1.

The EKF parameters relate to R, the process noise covariance matrix, and to Q, the measurement noise covariance matrix, as

$$R = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}$$
$$Q = \begin{bmatrix} q & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & q \end{bmatrix}.$$

In the extended Kalman filter more options are available. The noise covariance matrices of the motion model and measurements model make up to a total of five different parameters (assuming the noise in different directions and over different sensors is independent). We simplify this to two parameters for the process noise, r_1 and r_2. We still wan't these separated as we have chosen a complete random motion model for the angular velocity. It's assumed constant, but with added noise. The noise in the motion model for the angle is assumed to be of lesser magnitude. In the measurement model we however assume the same variance for all measurements, q.

To compare the performance of the Madgwick filter and the extended Kalman filter a ground truth is needed. With limited time and budget we couldn't use optical tracking or similar so instead we choose too see how well the filter estimates the final angle after a rotation. The angle which is easiest to get an accurate ground truth of is 90° degrees, that is lying down horizontally, since a spirit level of high accuracy can be used. The platform then starts initially at 0° degrees and is then rotated down to 90° degrees and the final angle is calculated using both filters, see Fig. 4.

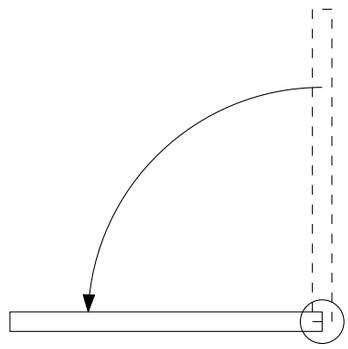

Figure 4: The platform starts initially at 0° degrees and is then rotated down to 90° degrees and the final angle is estimated using both filters.

4.2 Implementation

The platform can be seen in Fig. 4.The model of the IMU is Adafruit 9-DOF IMU Breakout - L3GD20H + LSM303. The measurement data is then processed in Matlab on a Macbook Pro Mid-2010 4GB RAM, 2,4 GHz Intel Core 2 Duo. The sensors were calibrated at 90° and the bias removed. The algorithms were then implemented in Matlab®.

Table 3: Filter Parameters

	Madgwick	EKF		
#	β	r_1	r_2	q
1	2.00	0.01	1.00	0.01
2	1.00	0.01	1.00	2.00
3	0.10	0.01	0.01	100.0
Optimal	0.0275	0.0015	1.265	1.550

5 Results

The results of different parameters are shown in Fig. 5 and in Fig. 6 for the Madgwick filter and in Fig. 7 and in Fig. 8 for the extended Kalman filter. In each plot the ground truth, $90° \pm 0.02°$ is shown as a black line. The error in the ground truth was calculated from the accuracy of the spirit level. Note that it's only the final angle that is accurately known.

To compare the filters the RMS error of 200 samples around the final angle of 6 different datasets was evaluated. The results are presented in Tab. 4 together with the mean calculation time per dataset.

Table 4: Comparison of Madgwick and EKF filter

	RMS [°]	Mean Time
Madgwick	0.160	0.114
EKF	0.178	0.153

6 Conclusions

6.1 Filter Parameters

In the Madgwick filter the variance in the estimated angle depend greatly on the parameter β. A too large β corresponds to a large step size μ leading to numerical instability, furthermore the filter will weight the gyroscope measurements less which might be a problem when the IMU is accelerating since we are neglecting acceleration of the IMU in the calculations, only taking into account the acceleration due to gravity. A smaller β yields better results, but a too small β would weight the gyroscope measurements too high making the estimation more prone to the gyroscope estimate "drifting" away due to accumulated noise in the integration. This is a common problem discussed in [7]. This can't be seen here as the filter operates over a quite short time span.

In the extended Kalman filter, the results of the different parameter settings are expected. In parameter settings #1 a too small variance in the measurement model leads to a noisy result as the Kalman gain gets large and each update step causes the mean to be corrected by a large amount, see line 4 in Tab. 2. However, a too certain motion model together with a large uncertainty in the measurement model yields a small Kalman gain and hence a slow convergence to the true mean by the same reasoning. This can be seen with parameter setting #3.

6.2 Filter Comparison

The final comparison test shows, in accordance with the result of [7], that the Madgwick filter performs well. Even though the number of states is small, the Madgwick filter is 25% faster than the EKF with equal accuracy. It converges fast and with correct parameter settings, it's estimate is smooth. The goal of this project was to get hands on with an implementation of the Madgwick and the extended Kalman filter in a real-world problem, understand the algorithms and how changing the parameter affects the estimation and all of this was captured within the scope of this report. The parameters of the Kalman filter were a bit simplified, the covariance matrix of the measurement noise can be changed to capture different measurement noise of the accelerometer and the gyroscope. The measurements were, however, so accurate so it was deemed unnecessary in this application. It is also worth mentioning that the state vector in the EKF implementation was simple and can be extended to also hold the bias of the sensors as well, as was done in [4].

However, the setup is simple. For a more thorough comparison an experiment similar to the one conducted in [7] has to be carried out. Since the goal was to understand and implement the Madgwick filter, such a setup was far beyond the scope of this project. Furthermore, only one angle is estimated. The full orientation requires the magnetometer measurements to be incorporated as well, showing the full potential of the Madgwick filter.

References

[1] Gradient descent, wikipedia. `https://en.wikipedia.org/wiki/Gradient_descent`. Accessed: 2015-12-09.

[2] Introducing the quaternions. `http://math.ucr.edu/~huerta/introquaternions.pdf`. Accessed: 2015-12-07.

[3] Wikipedia: Quaternion. `https://en.wikipedia.org/wiki/Quaternion`. Accessed: 2016-01-06.

[4] Billur Barshan and Hugh F Durrant-Whyte. Inertial navigation systems for mobile robots. *Robotics and Automation, IEEE Transactions on*, 11(3):328–342, 1995.

[5] J.J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley series in electrical and computer engineering: control engineering. Pearson/Prentice Hall, 2005.

[6] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[7] Sebastian OH Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 2010.

[8] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent robotics and autonomous agents. MIT Press, 2005.

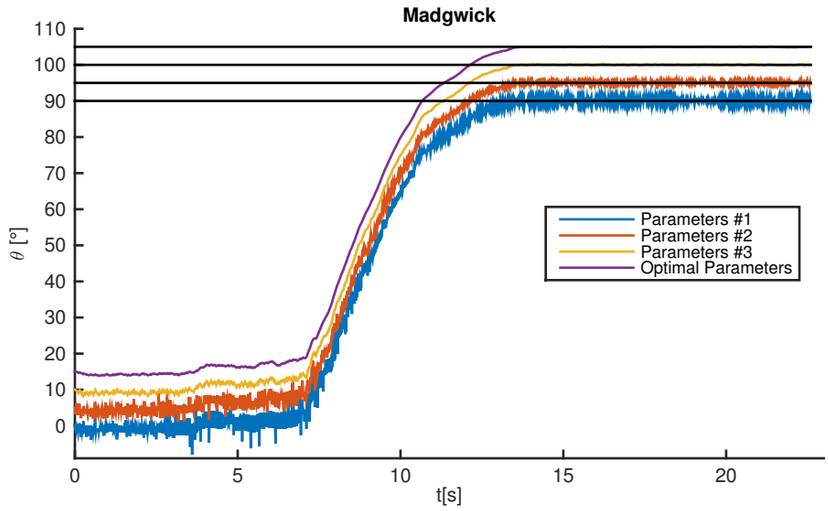

Figure 5: Results of the Madgwick filter for different parameters Tab. 3. The results are shifted 5° degrees each for illustrative purposes.

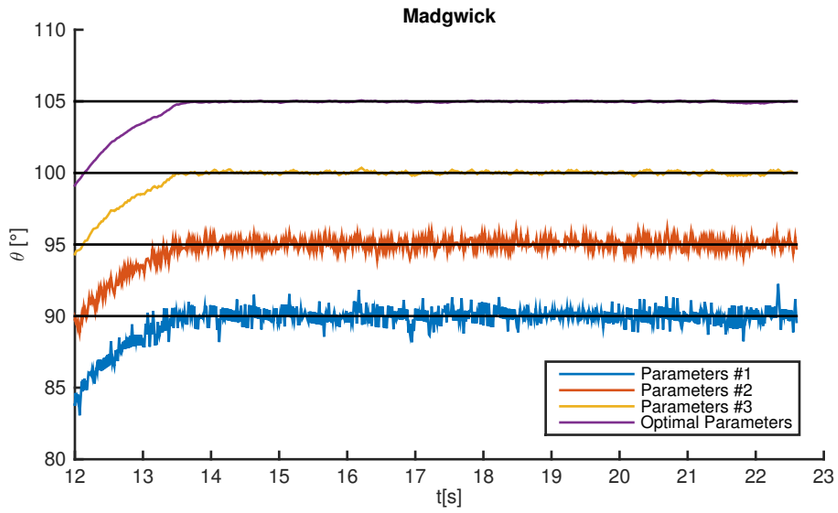

Figure 6: Results of the Madgwick filter for different parameters zoomed around the final angle.

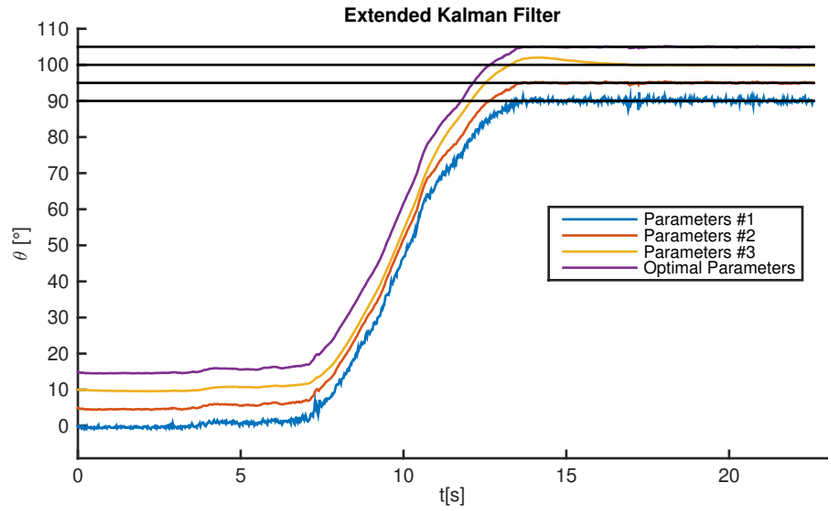

Figure 7: Results of the extended Kalman filter for different parameters as defined in Tab. 3. The results are shifted 5° degrees each for illustrative purposes.

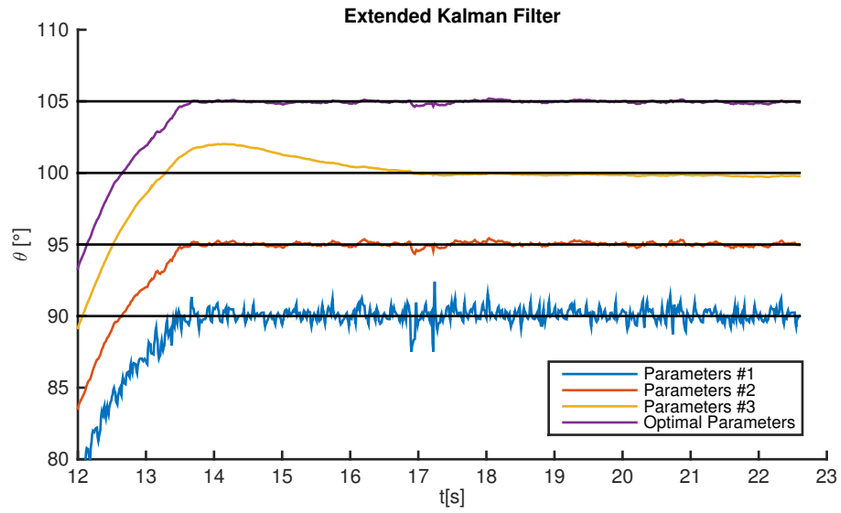

Figure 8: Results of the extended Kalman filter for different parameters zoomed around the final angle.