Attitude determination of a relative position sensor

TTK4550
Engineering Cybernetics, Specialization Project

Martin Tysseland

June 2021

Summary

This thesis contains insight into how to determine the attitude of a circuit board. For this, measurements from an IMU (inertial measurement unit) combined with an ESKF (error state Kalmen filter) have been used. The IMU provides measurements from three different sensors; accelerometer, gyroscope, and magnetometer. The final attitude for the circuit board is displayed in the Euler angles roll, pitch and yaw. Estimating roll and pitch with gyroscope measurement and correction from the accelerometer measurement, as shown in the result, was mainly accurate. However, I did not succeed in accurate and reliable estimate yaw with gyroscope measurement and correction from the magnetometer measurement in my implementation. Possible reasons for this are discussed in Section 5.

Preface

I want to thank my supervisor, Torleiv H. Bryne from NTNU, for helping me with this thesis. Throughout the project, I have gained new insight into some aspects of the usage of sensor fusion. It has been really challenging to write a thesis, but much has been learned and overall made me more robust for upcoming challenges. I also want to thank Johannes Alming Daleng from Squarehead Technology for providing the hardware used in this thesis and for assisting with valuable knowledge and insight.

NTNU
Norwegian University of
Science and Technology

Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

PROJECT THESIS DESCRIPTION SHEET

Name:	Martin Tysseland
Department:	Engineering Cybernetics
Thesis title (Norwegian):	Orienteringsbestemmelse av en relativ posisjoneringssensor
Thesis title (English):	Attitude determination of a relative position sensor

Thesis Description: In object detection applications, e.g., using electro optical sensors (cameras) or microphone arrays, object position or direction (line of bearing) is only known relative to the sensor. To use such information for absolute positioning depends on knowing the sensors location as well as position and orientation versus some reference coordinate system. With the availability of low-cost combined accelerometer / gyroscope / magnetometer chips, so called 9-DOF inertial measurement units (IMUs) a compact and low footprint sensor has the potential to be integrated with many other types of sensor system.

The following tasks should be considered:

1. Perform a short literature study on extended-Kalman-filter-based techniques for attitude determination using inertial sensor and magnetometer measurements.
2. Implement an attitude and heading reference system (AHRS) with error-state extended-Kalman-filter-based corrections using
    a. Accelerometer levering for roll and pitch determination
    b. Heading corrections for a magnetometer heading determination.
3. Test the algorithm collecting gyro, accelerometer, and magnetometer measurement data from a lab prototype provided by the industry partner.
4. If time permits review techniques for magnetometer calibration, including identifying possible "field friendly" techniques minimizing dependency on calibration rigs or special training.
5. Present your results and discuss these.
6. Conclude your results and suggest further work.

Start date:	2021-02-22
Due date:	2021-06-28
Thesis performed at:	Department of Engineering Cybernetics, NTNU
Supervisor:	Associate professor Torleiv H. Bryne, Dept. of Eng. Cybernetics, NTNU

Contents

1 Introduction

1.1 Project

This thesis aims to compute the attitude of a circuit board based on sensor readings from an accelerometer, gyroscope, and magnetometer. All of these measurements come from an IMU. The gyroscope outputs from the IMU will be integrated through a kinematic model to provide an estimate of the attitude represented as a unit quaternion. The unit quaternion will provide a 4-dimensional attitude representation that will avoid singularities. When describing the attitude in the results, the quaternion will be converted to the triple Euler angles roll, pitch, and yaw for better readability.

Further, an ESKF (error state Kalman filter) will estimate and compensate for bias and filter the noisy measurements. Acceleration and magnetometer measurements will, among other things, be used for this correction. The singularity-free unit quaternion fits well in the state vector in the ESKF. The state vector also contains the gyroscope bias.

1.2 IMU

IMU stands for inertial measurement unit and is sensor assembly of inertial sensors. All of the sampled sensor data in this thesis is gathered from a so-called 9 degree of freedom. The chosen IMU is called LSM9DS1 and is featured with a 3D digital linear acceleration sensor, a 3D digital angular rate sensor, and a 3D digital magnetic sensor (STMicroelectronics, 2015). An IMU is never perfect, and the main errors are measurement noise and measurement bias. Even with calibration these errors still has to be taken into account when performing calculation with measured values. In addition, it is not given that the mounted IMU is perfectly aligned with the body frame coordinates. All of these problems give room for possible errors when estimating roll, pitch, and yaw.

An essential aspect of an IMU is the sampling rate that is used. This is most often described in Hz and tells us how many samples are measured in one second. The IMU does typically provide measurements much more frequently than other navigation sensors, like GNSS (Global Navigation Satellite Systems). The sampling time of the IMU measurements used when testing is 100 Hz and was present in the hardware delivered by Squarehead Technology. The hardware is further described in Section 1.3 and comes with software to run and extract data from the IMU.

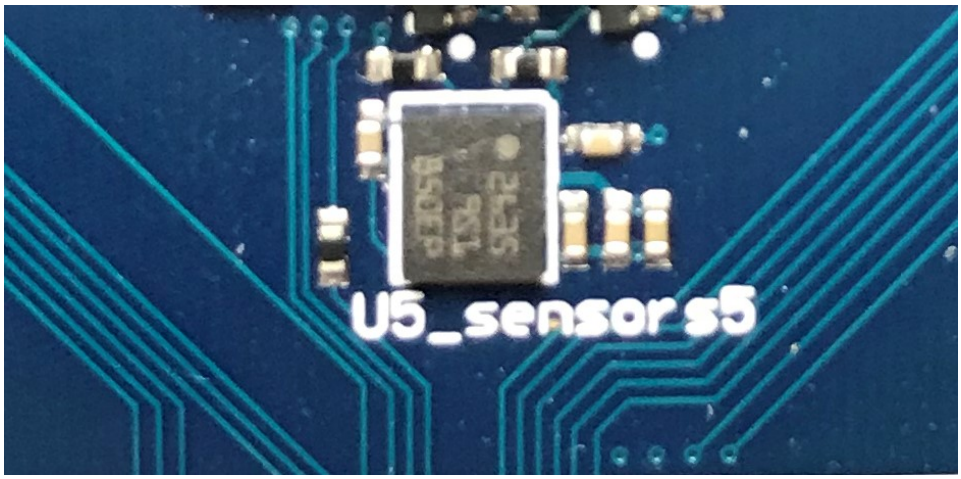

Figure 1.1: A close-up look of the middle IMU that is mounted on the circuit board.

1.3 Circuit board

Squarehead Technology provides hardware for testing that is the same as what they use in their products. The hardware is a circuit board composed of different audio and video processing features and access to mounted IMUs. The only feature that is of interest by now is the mounted IMUs. The circuit board is shown in Figure 1.2. Because the true look of the complete circuit board is classified, the figure shows only a drawing. There are five mounted IMUs, and their location is highlighted in the figure as a black square with white filling and a black dot in the left lower corner. Four IMUs are located in all corners of the circuit board, and the fifth is located in the middle. All IMUs are operative and gather data, but only the data from the IMU in the middle is used when estimating attitude for the circuit board. For power and data transfer, a 1-meter long USB-C to USB-A cable is used. This cable is connected to a stationary computer running on Ubuntu 18.04 LTS operative system.

The orientation shown in Figure 1.2 represents the idle orientation of the circuit board. In this orientation roll, pitch and yaw are expected to be zero. The coordinate system for the circuit board is shown in Figure 1.3. Movement about the x-axis will represent changes in roll, movement about the y-axis will represent changes in pitch, and movement about the z-axis will represent changes in yaw. The local coordinate system for the IMUs was not originally aligned with the chosen coordinate system shown in Figure 1.3. This was adjusted by rotating the IMUs measurements vector with a rotation matrix. The method used is further explained in Section 3.2.

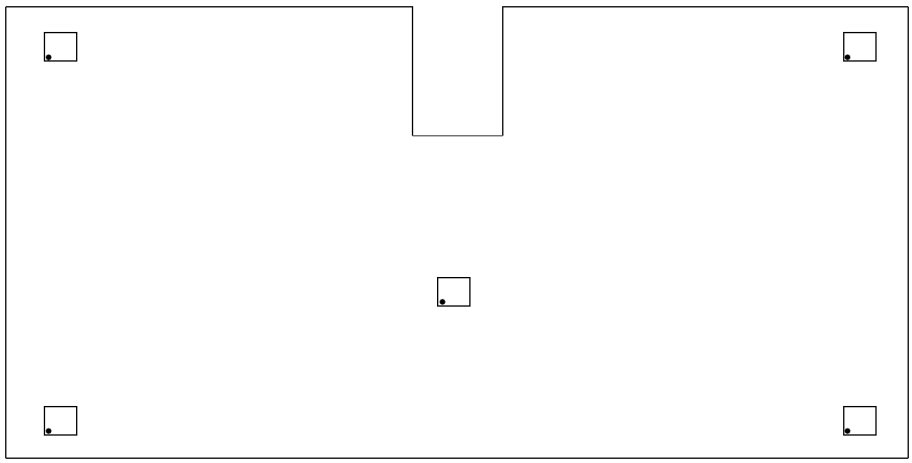

Figure 1.2: A complete drawing of the circuit board provided by Squarehead Technology. Only the IMU located in the middle was used when estimating the results. The dot shown on the IMUs indicates the mounted rotation of the IMU relative to the circuit board.

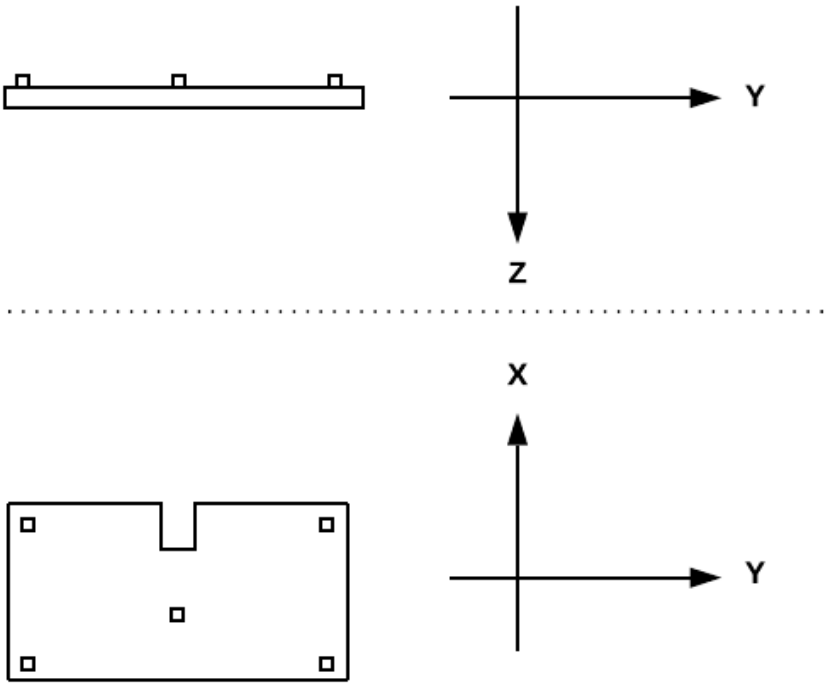

Figure 1.3: The coordinate system that is used for the circuit board. It shows the direction for the x-axis, the y-axis, and the z-axis.

2 Theory

This chapter will introduce the necessary mathematics and theory to better understand calculation with quaternion, converting between quaternion, rotation matrix, Euler angles, and step by step implementation of the ESKF with explanatory equations.

2.1 Norm and Skew-symmetric matrix

In further calculation the term Euclidean norm will be used on vectors when need of scaling. Euclidean norm (Farrell, 2008, B.2) of a n-dimensional vector $\mathbf{v} = \{v_1, ..., v_n\}$ is defined as the square root of the scalar product of a vector with itself:

$$||\mathbf{v}||_2 = \sqrt{\mathbf{v}^T \cdot \mathbf{v}} \tag{2.1}$$

$$||\mathbf{v}||_2 = \sqrt{v_1^2 + ... + v_n^2} \tag{2.2}$$

The Euclidean norm is also the length of the vector. This property can further be used to normalize a vector with

$$\mathbf{v} = \frac{\mathbf{v}}{||\mathbf{v}||_2} \tag{2.3}$$

that can be used when only the relation between the values in the vector is of interest. This will be done with the measurements from the accelerometer and magnetometer. For these measurements, only the direction of the vector is necessary. Normalizing a vector is a way to scale the vector, so the vector's length is equal to 1.

For some calculation in the ESKF it is needed to take the cross product between two vectors. To simplify this the Skew-symmetric matrix operator will be used. The definition for this is such that $\mathbf{v} \times \mathbf{a} = \boldsymbol{S}(\mathbf{v})\mathbf{a}$, or equivalently (Brekke, 2020, eq. 10.5):

$$\boldsymbol{S}(\mathbf{v}) = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{2.4}$$

2.2 Quaternion

The quaternion will be used as the main way to represent the attitude state. This is a 4-dimensional attitude notation and will not suffer from singularities, like any 3-dimensional representation, where an infinite number of possible Euler angles

are possible (Brekke, 2020, ch. 10). The mathematical notation for a quaternion used in this thesis is (Brekke, 2020, eq. 10.19):

$$\boldsymbol{q} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} \tag{2.5}$$

where

$$\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \epsilon_3]^T \tag{2.6}$$

Taking the norm of a quaternion is done the same way as in (2.2) and yields (Brekke, 2020, eq. 10.25):

$$||\mathbf{q}|| = \sqrt{\eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2} \tag{2.7}$$

Now it is very straightforward to normalize the quaternion with

$$\mathbf{q} = \frac{\mathbf{q}}{||\mathbf{q}||} \tag{2.8}$$

and is similar as done for a normal vector in (2.3).

Another good property for the quaternion is when you want to rotate the quaternion attitude representation. Suppose you have one quaternion representing the attitude and another quaternion representing the change. You can take the product between these two for obtaining the new attitude. The calculation of the quaternion product (Brekke, 2020, eq. 10.34) is:

$$\mathbf{q}_a \otimes \mathbf{q}_b = \left(\eta_a \mathbf{I} + \begin{bmatrix} 0 & -\boldsymbol{\epsilon}_a^T \\ \boldsymbol{\epsilon}_a & -\boldsymbol{S}(\boldsymbol{\epsilon}_a) \end{bmatrix} \right) \begin{bmatrix} \eta_b \\ \boldsymbol{\epsilon}_b \end{bmatrix} \tag{2.9}$$

2.3 Conversion between quaternion, rotation matrix and Euler angles

It is helpful to convert between different attitude representations. While the quaternion is used as the attitude state, a mix of quaternion and rotation matrix will be used in the ESKF calculations. When plotting the estimated attitude for the results, the Euler angles roll, pitch, and yaw will be used. This makes a more familiar and readable representation of the attitude.

Firstly we have the equation for converting from quaternion to rotation matrix (Brekke, 2020, eq. 10.37):

$$\mathbf{R} = \mathbf{I} + 2\eta \boldsymbol{S}(\boldsymbol{\epsilon}) + 2\boldsymbol{S}(\boldsymbol{\epsilon})\boldsymbol{S}(\boldsymbol{\epsilon}) \tag{2.10}$$

$$= \begin{bmatrix} \eta^2 + \epsilon_1^2 - \epsilon_2^2 - \epsilon_3^2 & 2\left(\epsilon_1\epsilon_2 - \eta\epsilon_3\right) & 2\left(\epsilon_1\epsilon_3 + \eta\epsilon_2\right) \\ 2\left(\epsilon_1\epsilon_2 + \eta\epsilon_3\right) & \eta^2 - \epsilon_1^2 + \epsilon_2^2 - \epsilon_3^2 & 2\left(\epsilon_2\epsilon_3 - \eta\epsilon_1\right) \\ 2\left(\epsilon_1\epsilon_3 - \eta\epsilon_2\right) & 2\left(\epsilon_2\epsilon_3 + \eta\epsilon_1\right) & \eta^2 - \epsilon_1^2 - \epsilon_2^2 + \epsilon_3^2 \end{bmatrix} \tag{2.11}$$

The rotation matrix is an attitude representation of 9 dimensions and will be used for intermediate calculations in the ESKF method.

Secondly, we have the equation for converting from quaternion to Euler angles (Brekke, 2020, eq. 10.38):

$$\phi = \text{arctan2}\left(2(\epsilon_3\epsilon_2 + \eta\epsilon_1), \eta^2 - \epsilon_1^2 - \epsilon_2^2 + \epsilon_3^2\right) \tag{2.12}$$

$$\theta = \arcsin\left(2(\eta\epsilon_2 - \epsilon_1\epsilon_3)\right) \tag{2.13}$$

$$\psi = \text{arctan2}\left(2(\epsilon_1\epsilon_2 + \eta\epsilon_3), \eta^2 + \epsilon_1^2 - \epsilon_2^2 - \epsilon_3^2\right) \tag{2.14}$$

This will be used when plotting the attitude state in the results. Then we have to go from quaternion to a more readable representation of the attitude.

The last conversion is the equation for converting from Euler angle to quaternion (Brekke, 2020, eq. 10.39):

$$\mathbf{q} = \begin{bmatrix} \cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \sin\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} - \cos\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} - \sin\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} \end{bmatrix} \tag{2.15}$$

It is more intuitive to represent the initial attitude of the circuit board as Euler angles in the Python code. This is then immediately converted to quaternion for the initial attitude state.

2.4 The error state Kalman filter (ESKF)

2.4.1 States

Will step by step show the mathematics and method behind the ESKF. We have three different state kinematics for the system, the nominal state, error state, and true state. The nominal and true state will represent the attitude and gyro bias, and the error state will represent the error between these state vectors. The notation for these will be the same as in (Brekke, 2020, eq. 10.52), but with fewer states. Since it is only of interest to estimate the attitude of the circuit board, the states containing the position and velocity are removed. The accelerometer is only used

for correction and compared with Earth's gravity, so acceleration bias is removed to simplifying the state. A bias in acceleration will only yield a slight offset in the attitude representation and is durable. If the position and velocity were also predicted, this would lead to a constant drift and make the position more inaccurate over time.

Since the error state will represent the error in attitude and gyro bias, it is no need to use a 4-dimensional representation of the attitude. Here a 3-dimensional Euler angles representation is more suited. The ESKF will always try to make all of the entities in this state go towards zero. The representation is:

$$\delta \mathbf{x} = \begin{bmatrix} \delta \boldsymbol{\theta} & \delta \boldsymbol{\omega}_b \end{bmatrix}^T \tag{2.16}$$

The nominal state will be the predicted representation of the attitude and gyro bias. It is from this state that we extract the results shown in the plots in Section 4. The representation is:

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} & \boldsymbol{\omega}_b \end{bmatrix}^T \tag{2.17}$$

The true state is completely unknown, given that this is a real experiment. If all of the data was generated through simulations, we could have used the true state of compassion when evaluating the performance of the implemented ESKF. This is not done in this thesis, but the true state is mentioned for possible later usage in further work. The representation is:

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{q}_t & \boldsymbol{\omega}_{bt} \end{bmatrix}^T \tag{2.18}$$

So when trying to estimate the error state, we use the measurements from the accelerometer and magnetometer to assist. Extracting the quaternion from the nominal state, converting it to a rotation matrix, and multiply it with the earth's gravity vector yields an acceleration vector close to what the accelerometer measurement from the IMU should be. Looking at the difference here gives us information on possible errors in the attitude and gyro bias. This is explained in more detail when updating the Kalman filter. The same principle is used for the magnetometer measurements and the magnetic field around the circuit board. Combining these measurements, we can validate the roll and pitch with an accelerometer and yaw with a magnetometer.

Further, we have the error state covariance matrix $\mathbf{P}$. When estimating the attitude and gyroscope bias for the circuit board, we follow these steps:

1. Set $\mathbf{x}_0^-$, $\mathbf{P}_0^-$ and $k = 0$

2. With new specific force and magnetometer measurements we calculate the error state $\delta\mathbf{x}_k$, correct the nominal state $\mathbf{x}_k^+$ and update the error state matrix $\mathbf{P}_k^+$. Otherwise we set $\mathbf{x}_k^+ = \mathbf{x}_k^-$ and $\mathbf{P}_k^+ = \mathbf{P}_k^-$.
3. Predict $\mathbf{x}_{k+1}^-$ and $\mathbf{P}_{k+1}^-$ with gyroscope measurement.
4. Set $k = k + 1$ and repeat from 2.

2.4.2 Update Kalman filter

All of the measurements received from the IMU have the true magnitude of the measured values. In the case of the acceleration and magnetometer, only the direction of the measurements is of interest. With this in mind, we can scale the measurement vectors and the corresponding reference vectors with normalizing (2.2):

$$\mathbf{g} = \frac{\mathbf{g}}{||\mathbf{g}||_2} \tag{2.19}$$

$$\mathbf{z}_{acc} = \frac{\mathbf{z}_{acc}}{||\mathbf{z}_{acc}||_2} \tag{2.20}$$

$$\mathbf{m}_{ref} = \frac{\mathbf{m}_{ref}}{||\mathbf{m}_{ref}||_2} \tag{2.21}$$

$$\mathbf{z}_{mag} = \frac{\mathbf{z}_{mag}}{||\mathbf{z}_{mag}||_2} \tag{2.22}$$

As mentioned in Section 2.4.1 we further extract $\mathbf{q}_b^n$ from the nominal state and convert this to the rotation matrix $\mathbf{R}_b^n$. Than estimate the acceleration measurement in this orientation given Earth's gravity as the only force (Farrell, 2008, eq. 11.143):

$$\hat{\mathbf{y}}_{acc} = (\mathbf{R}_b^n)^T \cdot -\mathbf{g} \tag{2.23}$$

This will be used for correction in roll and pitch. For correction in yaw, we calculate the cross-product between the accelerometer and magnetic field measurement for better stability. We do the same for Earth's gravity and magnetic reference.

$$\mathbf{z}_{mag} = \mathbf{z}_{acc} \times \mathbf{z}_{mag} \tag{2.24}$$

$$\hat{\mathbf{y}}_{mag} = (\mathbf{R}_b^n)^T \cdot (-\mathbf{g} \times \mathbf{m}_{ref}) \tag{2.25}$$

Here $\mathbf{g} = [0, 0, 9.81]^T$ and $\mathbf{m}_{ref}$ is set equal to the magnetometer measurement when the sensor is in an idle position. In this position, the headings direction is straight to synthetic north as described in Section 3.5.

For correction of the nominal state, we need to calculate the measurement matrix:

$$\mathbf{H}_{6x6} = \begin{bmatrix} \boldsymbol{S}(\hat{\mathbf{y}}_{acc}) & \mathbf{0} \\ \boldsymbol{S}(\hat{\mathbf{y}}_{mag}) & \mathbf{0} \end{bmatrix} \tag{2.26}$$

The basis for the measurement matrix comes from:

$$\hat{\mathbf{y}} = -\hat{\mathbf{R}}^T \mathbf{g} \tag{2.27}$$
$$\mathbf{y} = -\mathbf{R}^T \mathbf{g} \tag{2.28}$$
$$= (\mathbf{I} + \boldsymbol{S}(\delta\boldsymbol{\theta})^T \hat{\mathbf{R}}^T \mathbf{g} \tag{2.29}$$
$$= -\hat{\mathbf{R}}^T \mathbf{g} - \boldsymbol{S}(\delta\boldsymbol{\theta})^T \hat{\mathbf{R}}^T \mathbf{g} \tag{2.30}$$
$$= -\hat{\mathbf{R}}^T \mathbf{g} + \boldsymbol{S}(\delta\boldsymbol{\theta}) \hat{\mathbf{R}}^T \mathbf{g} \tag{2.31}$$
$$= -\hat{\mathbf{R}}^T \mathbf{g} - \boldsymbol{S}(\hat{\mathbf{R}}^T \mathbf{g}) \delta\boldsymbol{\theta} \tag{2.32}$$
$$= \hat{\mathbf{y}} + \boldsymbol{S}(\hat{\mathbf{y}}) \delta\boldsymbol{\theta} \tag{2.33}$$

where we extract $\boldsymbol{S}(\hat{\mathbf{y}})$ and use in $\mathbf{H}$. Further, we calculate the Kalman gain matrix from (Solà, 2017, eq. 274) and (Brekke, 2020, eq. 10.75):

$$\mathbf{K} = \mathbf{P}_k^- \mathbf{H}^T \left(\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_{acc} \right)^{-1} \tag{2.34}$$

The Kalman gain matrix is then used to calculate the estimated error state (Solà, 2017, eq. 275):

$$\delta\mathbf{x}_k = \mathbf{K} \begin{bmatrix} \mathbf{z}_{acc} - \hat{\mathbf{y}}_{acc} \\ \mathbf{z}_{mag} - \hat{\mathbf{y}}_{mag} \end{bmatrix} \tag{2.35}$$

Now we have to update the error state covariance matrix $\mathbf{P}_k^+$ (Solà, 2017, eq. 276) and is done with the symmetric and positive Joseph form:

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}\mathbf{H}) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}\mathbf{H})^T + \mathbf{K} \mathbf{R}_{acc} \mathbf{K}^T \tag{2.36}$$
$$\mathbf{P}_k^+ = \frac{\mathbf{P}_k^+ + (\mathbf{P}_k^+)^T}{2} \tag{2.37}$$

After estimating the error state, it is time to update the nominal state $\mathbf{x}_k^+ = \mathbf{x}_k^- \otimes \delta\mathbf{x}_k$ with corrections from the error state (Brekke, 2020, eq. 10.72) and (Solà, 2017, eq. 283c, 283e):

$$\delta\mathbf{q} = \begin{bmatrix} 1 \\ \delta\boldsymbol{\theta}/2 \end{bmatrix} \tag{2.38}$$
$$\mathbf{q} = \mathbf{q}_b^n \otimes \delta\mathbf{q} \tag{2.39}$$

$$\boldsymbol{\omega}_b = \boldsymbol{\omega}_b + \delta\boldsymbol{\omega}_b \tag{2.40}$$

We want to keep the quaternion as a unit quaternion. This means that the length of the quaternion is equal to 1 and will be achieved by normalizing it, as done in (2.8).

Finally, we also correct the error states covariance matrix $\mathbf{P}_k^+$ after correction of the nominal states (Brekke, 2020, eq. 10.86):

$$\mathbf{P}_k^+ = \mathbf{G}\mathbf{P}_k^+\mathbf{G}^T \qquad \mathbf{G}_{6x6} = \begin{bmatrix} \mathbf{I} - S(\delta\boldsymbol{\theta}/2) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{2.41}$$

2.4.3 Predict the next nominal state

Now the next nominal state has to be predicted. Starting with obtaining the IMU measurements assuming ω is constant over the sampling time period:

$$\boldsymbol{\omega} \approx \mathbf{z}_{gyro} - \boldsymbol{\omega}_b \tag{2.42}$$

Further extracting $\mathbf{q}$ from the nominal states and predict the next $\mathbf{q}$ with respect to $\boldsymbol{\omega}$ (Solà, 2017, eq. 214):

$$\boldsymbol{\kappa} = T_s \cdot \boldsymbol{\omega} \tag{2.43}$$

$$\Delta\mathbf{q} = e^{\frac{\boldsymbol{\kappa}}{2}} = \begin{bmatrix} \cos(\frac{||\boldsymbol{\kappa}||_2}{2}) \\ \sin(\frac{||\boldsymbol{\kappa}||_2}{2})\frac{\boldsymbol{\kappa}^T}{||\boldsymbol{\kappa}||_2} \end{bmatrix} \tag{2.44}$$

$$\mathbf{q} = \mathbf{q} \otimes \Delta\mathbf{q} \tag{2.45}$$

Same as for after updating the nominal state, the quaternion also has to be normalized in this case.

To predict the complete nominal state, also the gyroscope bias is predicted with (Brekke, 2020, eq. 10.58):

$$\boldsymbol{\omega}_b \approx \boldsymbol{\omega}_b + Ts * \dot{\boldsymbol{\omega}}_b \tag{2.46}$$

$$\boldsymbol{\omega}_b = \boldsymbol{\omega}_b - p_{\omega b} \cdot \mathbf{I}_3 \cdot \boldsymbol{\omega}_b \tag{2.47}$$

were the bias is modeled as a Gauss-Markov process (Brekke, 2020, eq. 10.50).

2.4.4 Predict the error state covariance matrix

After predicting the next nominal state, our last error state covariance matrix is no longer up to date. To overcome this, a new error state covariance matrix is predicted. Starting with extracting the $\mathbf{q}_b^n$ from the nominal state and convert this to the rotation matrix $\mathbf{R}_b^n$. Further, it will be shown a way to calculate the needed **F** and **G** matrices. In this project, the given method is used because the correction of the quaternion is done with the local angular error (Solà, 2017, table 4). This decision is seen in (2.39) when updating the nominal states with the corrections from the error states and is repeated here:

$$\mathbf{q} = \mathbf{q} \otimes \delta\mathbf{q} \tag{2.48}$$

and yields (Brekke, 2020, eq. 10.68):

$$\mathbf{F}_{6x6} = \begin{bmatrix} -\boldsymbol{S}(\boldsymbol{\omega}) & -\mathbf{I} \\ \mathbf{0} & -p_{\omega b}\mathbf{I} \end{bmatrix} \qquad \mathbf{G}_{6x6} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{2.49}$$

Now, the next error state covariance prediction can be obtained with Van Loan's formula (Brekke, 2020, eq. 4.63). First, construct the Van Loan matrix:x:

$$\mathbf{Q}_{6x6} = \begin{bmatrix} \sigma_\omega^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sigma_{\omega b}^2\mathbf{I} \end{bmatrix} \tag{2.50}$$

$$\boldsymbol{V}_{12x12} = \begin{bmatrix} -\boldsymbol{F} & \boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^T \\ \mathbf{0} & \boldsymbol{F}^T \end{bmatrix} \tag{2.51}$$

With this, we use Van Loan's formula:

$$\exp\left(\boldsymbol{V}T_s\right) = \exp\left(\begin{bmatrix} -\boldsymbol{F} & \boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^T \\ \mathbf{0} & \boldsymbol{F}^T \end{bmatrix} T_s\right) = \begin{bmatrix} \times & \boldsymbol{V}_2 \\ \mathbf{0} & \boldsymbol{V}_1 \end{bmatrix} \tag{2.52}$$

and further calculate:

$$\boldsymbol{F}_d = \boldsymbol{V}_1^T \tag{2.53}$$

$$\boldsymbol{Q}_d = \boldsymbol{V}_1^T\boldsymbol{V}_2 \tag{2.54}$$

Taking exp of the Van Loan matrix can be time-consuming. A faster approach for this is to use a 2. order Taylor approximation:

$$\exp\left(\boldsymbol{V}T_s\right) = \boldsymbol{I} + \boldsymbol{V}T_s + \frac{(\boldsymbol{V}T_s)^2}{2} \tag{2.55}$$

Finally, the prediction of the next error states covariance matrix can be done with:

$$\mathbf{P}_k^- = \boldsymbol{F}_d \mathbf{P}_{k-1}^+ \boldsymbol{F}_d^T + \mathbf{Q}_d \tag{2.56}$$

3 Implementation

3.1 Setup in Python

All of the sensor data extracted from the IMU in Figure 1.1 is stored in a CSV file. This file is generated when running the sensor with a Python program inside a Docker container. Different data set is labeled based on the given test case (type of movement in pitch, roll and yaw direction for a set time duration). The central part of the program works in the numbered stages:

1. Load sensor data from labeled data set.
2. Allocate memory for all the results.
3. Set parameters values as shown in Table 3.1.
4. Set initialization values for the nominal state and the error state covariance matrix as shown in (3.3).
5. Run a loop for all the iterations:
   a) Update the Kalman filter as shown in Section 2.4.2.
   b) Predict the next nominal state and error state covariance matrix as shown in Sections 2.4.3 – 2.4.4.
6. Plot the estimated nominal state for each iteration. The quaternion is converted to Euler angles with (2.12 – 2.14), and the gyro bias is taken directly from the state.

3.2 Rotation of sensors axes

From Figure 3.1 we see the original rotation for the IMU in Figure 1.1. The white dot shown on the IMU in Figure 3.1 corresponds to the dots on the mounted IMUs shown on the circuit board in Figure 1.2. For the axes displayed in Figure 1.3 to be correct, all of the measurements received were rotated before further use in the implemented system. The accelerometer and the gyroscope were rotated with the rotation matrix:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{3.1}$$

The magnetometer was rotated with the rotation matrix:

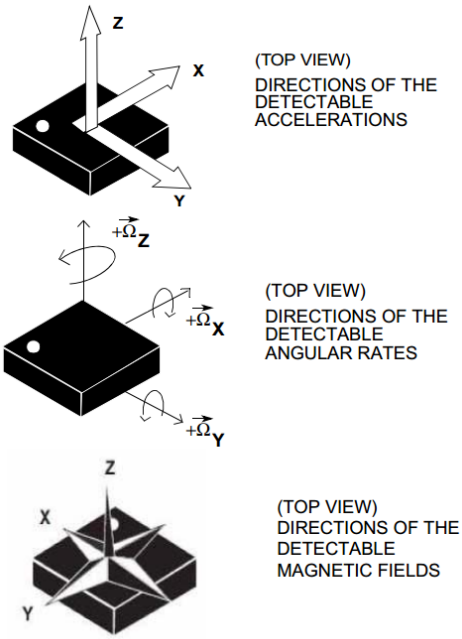

Figure 3.1: Sensor axes from the datasheet STMicroelectronics (2015, Figure 1.)

$$R = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{3.2}$$

Multiplying all of the measurements with its corresponding rotation matrix yields new orientations set in the same frame. With this, we get the wanted x-, y-, and z-direction for accelerometer, gyroscope, and magnetometer as shown in Figure 1.3. Now roll, pitch, and yaw are zero when the circuit board is in idle position. The new IMU rotation is used for all the measurement data that is gathered to estimate the attitude for the circuit board.

3.3 Tuning

The parameters were set by testing the system and adjusting for a better result. From the last table shown in Zinn (2018), focusing on the values for the LSM9DS1 model. This model is the same IMU that is mounted on the circuit board used in this project. Here we gather the standard deviation for the gyroscope measurement as 1.38 deg/$\sqrt{\text{h}}$ and the gyro bias stability as 61.2 deg/h. For usage in the model implemented in python, these values are converted to rad/s as shown in Table 3.1.

3.4 Parameters and initial states

Accelerator and magnetometer measurements are used in the Kalman filter to correct the estimated attitude based on gyroscope measurements. The constant noise variance matrix values for acceleration and magnetometer used in the Kalman filter are shown in Table 3.1 as $\mathbf{R}_{acc}$ and $\mathbf{R}_{mag}$.

Parameter	Value	Unit
$\mathbf{R}_{acc}$	$0.5^2 \cdot \mathbf{I}_3$	
$\mathbf{R}_{mag}$	$0.05^2 \cdot \mathbf{I}_3$	
σ_ω	$\frac{61.2}{3600} \cdot \frac{\pi}{180}$	rad/s
$\sigma_{\omega b}$	$\frac{1.38}{60} \cdot \frac{\pi}{180}$	rad/s
$p_{\omega b}$	$\frac{1}{3600}$	
fs	100	1/s
g	9.81	$\mathrm{m/s^2}$

Table 3.1: Parameters used for IMU data set.

$\mathbf{x}_0$ is the initial nominal state used when the sensor is in an assumed idle starting position. Here the identity quaternion is used and corresponds to roll, pitch, and yaw equal to zero. The gyro bias is also set to zero. This is fitting to the initial idle state that is wanted. $\mathbf{P}_0$ is used for the initial error state covariance matrix. These initial states are shown in (3.3).

$$\mathbf{x}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{P}_0 = \begin{bmatrix} \mathbf{I}_3(6 \cdot \frac{\pi}{180})^2 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3(1 \cdot \frac{\pi}{180})^2 \end{bmatrix} \tag{3.3}$$

3.5 Synthetic north for magnetic reference

For simplifying this thesis, the natural magnetic field in the office where all the testing was conducted was not calculated. Instead, a more extended test with the circuit board in an idle position was completed. For 1000 seconds with no interference, the IMU gathered measurement data. For this period, the mean of the magnetometer measurement was calculated and used as $\mathbf{m}_{ref}$. This represents magnetometer measurements when yaw equals zero, and the circuit board is facing straight synthetic north. This is our local north and the base for the idle position.

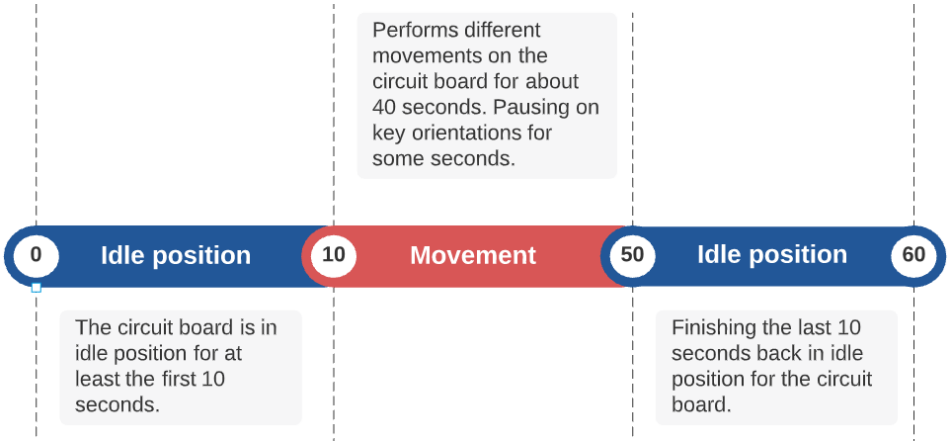

Figure 3.2: Timeline for actions done when the measurement data was gathered from the IMU.

3.6 Conducting testing

This circuit board shown in Figure 1.2 with the mounted IMU shown in Figure 1.1 was used in every test for this thesis. The sensor data is extracted with a program written in Python using a library supported by Squarehead Technology. The circuit board is lying leveled on a table and shows the base for the idle and starting position. For testing, only data from the middle IMU is used. It is not important that the attitude is accurately estimated when the circuit board is in large motion, but when it is set in a fixed orientation, the estimate of roll, pitch, and yaw needs to be as correct as possible.

The idle test was done by letting the sensor stay in an idle position for 5 minutes. For testing with movement, Figure 3.2 shows a timeline for how most of the testing was done. All of the movement was done by hand and contained errors like not moving correctly to wanted attitude, small changes in the wrong directions, movement away from the original location, and non-modeled linear acceleration. From (2.23), it is clear that only acceleration from Earth's gravity is modeled for the estimated acceleration measurement vector. So acceleration coming from the movements itself will be a source of error. Hopefully, slow movements will make the possible self-inflicted acceleration negligible. The sum of all these possible errors will impact the result and are some of the reasons behind minor spikes in gyro bias and attitude.

4 Results

4.1 Idle

The first result is based on the whole system in an idle position. For this test, no movement was carried out for the entire sampling period. The initial values for roll, pitch, and yaw were set to zero, and the test duration was 300 seconds. The blue line represents roll (ϕ), the red line pitch (θ), and the green line yaw (ψ).

As seen in Figure 4.1 the gyro bias converts toward the correct values. Based on the observation that the Euler angles stay around zero with no movement, we should expect an idle system. We notice that attitude is a bit off from idle, and this comes the acceleration measurement used in the measurement matrix does not account for accelerometer bias. Further, we observe that yaw is not as accurate as roll and pitch. This is clearly demonstrated with much higher variance in yaw and gyro bias around the z-axis than the other states shown in Figure 4.1.

The system is also tested for initial values not equal to zero. In Figure 4.2 initial roll is set to -30 degrees, pitch to 20 degrees and yaw to -40 degrees. As shown in the figure, the system is still able to convert to the correct values. This yields a larger transient bias estimate but progresses towards the same bias values as in Figure 4.1.

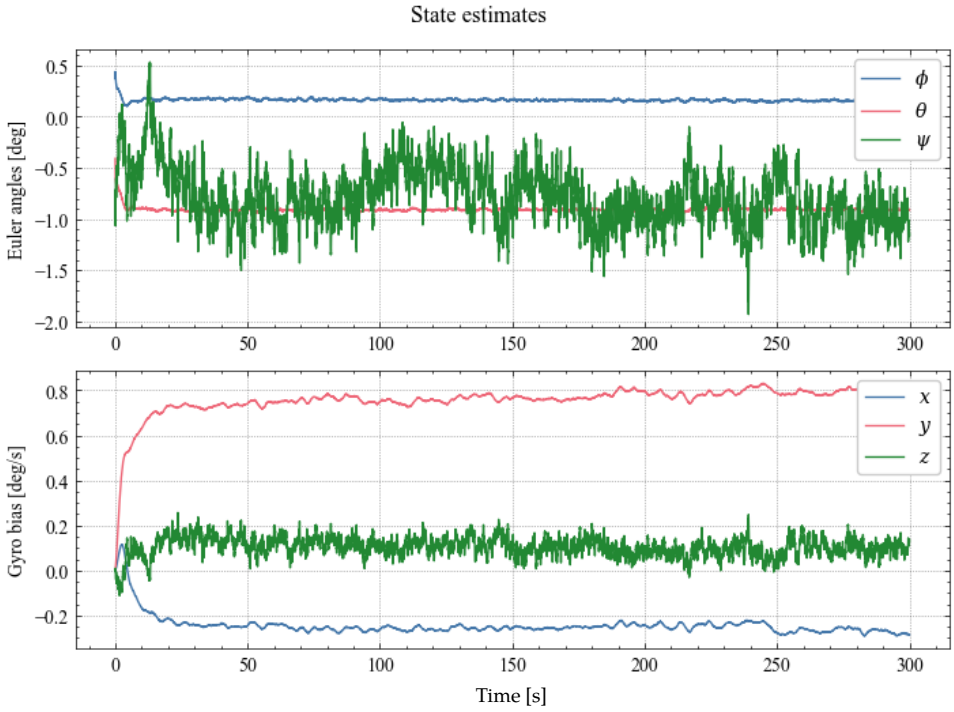

Figure 4.1: State estimates showcasing attitude and gyro bias in idle position for 300 seconds

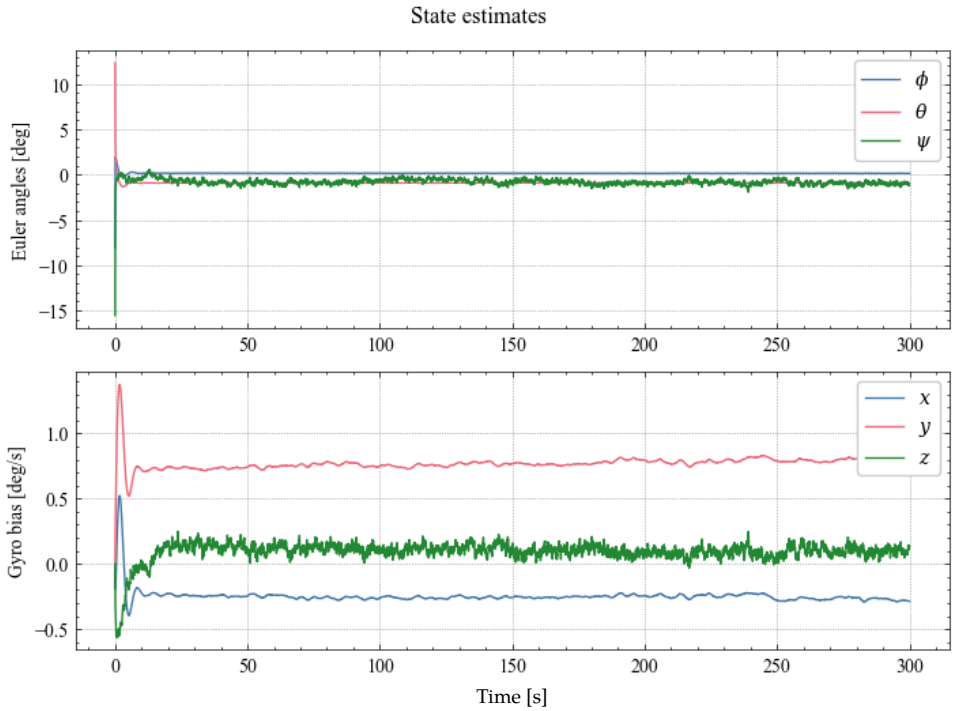

Figure 4.2: State estimates showcasing attitude and gyro bias in idle position for 300 seconds. The initial roll was set to -30 degrees, pitch to 20 degrees, and yaw to -40 degrees.

4.2 45, 90 and 180 degree roll

The system also needs to work for different angular movements. Testing of different magnitude of movement in only roll direction is shown in Figures 4.3 – 4.5. Here we see the estimated angular movement in roll following the movement conducted in the test. When larger movement in roll is done, the error in yaw increases. This comes from the not optimal correction of attitude with the help of magnetometer measurement that is a repeating problem for many testing cases in this project.

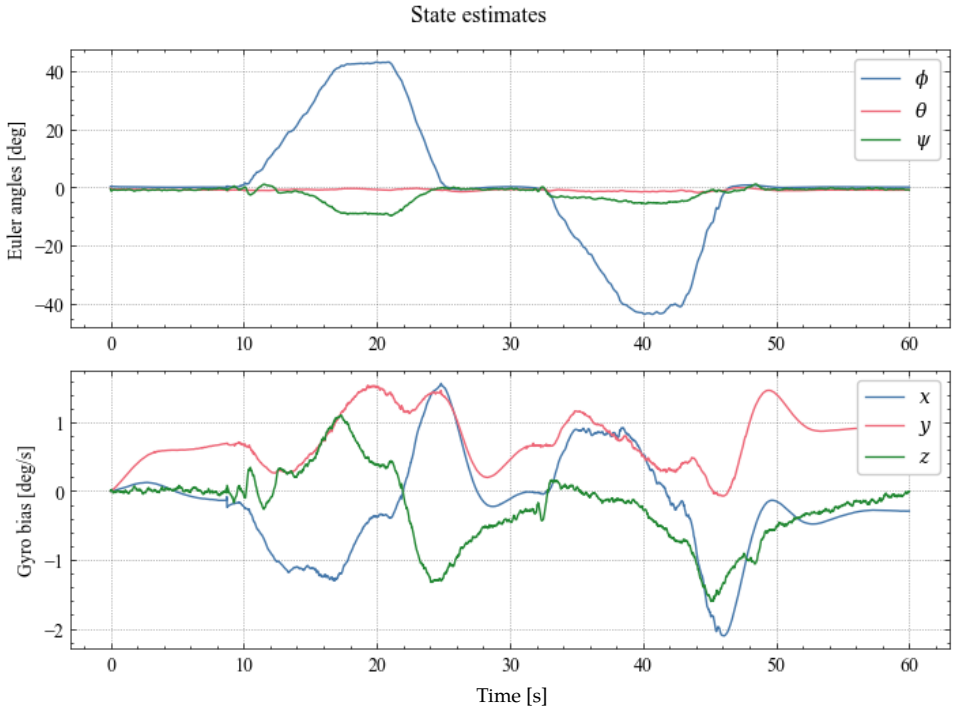

Figure 4.3: State estimates showcasing attitude and gyro bias for 45 degrees movement in positive and negative roll direction and back to idle.

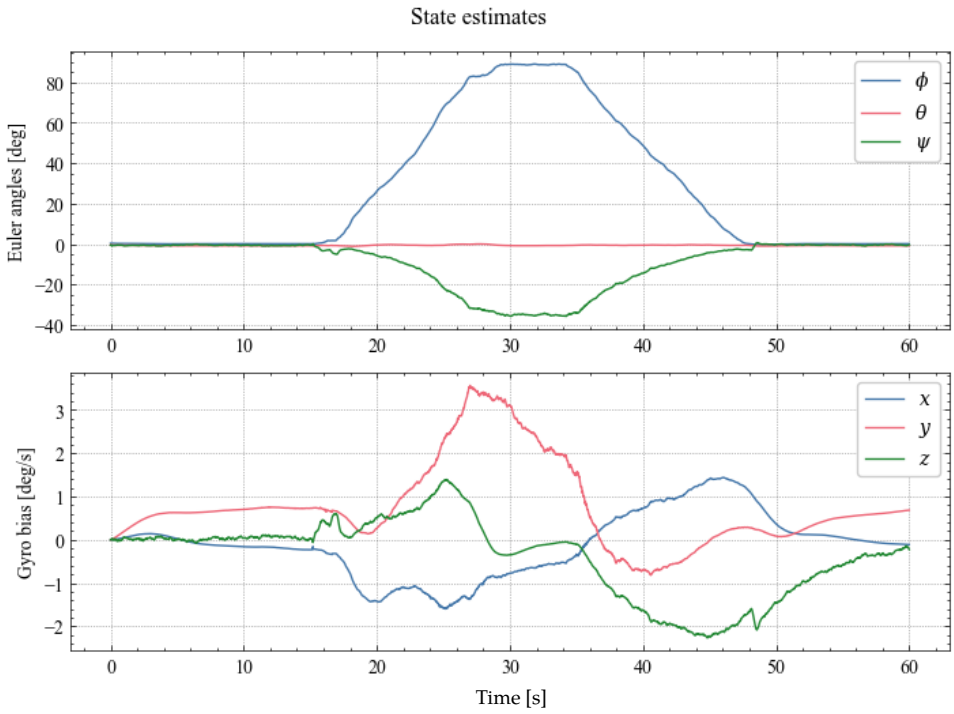

Figure 4.4: State estimates showcasing attitude and gyro bias for 90 degrees movement in positive roll direction and back to idle.

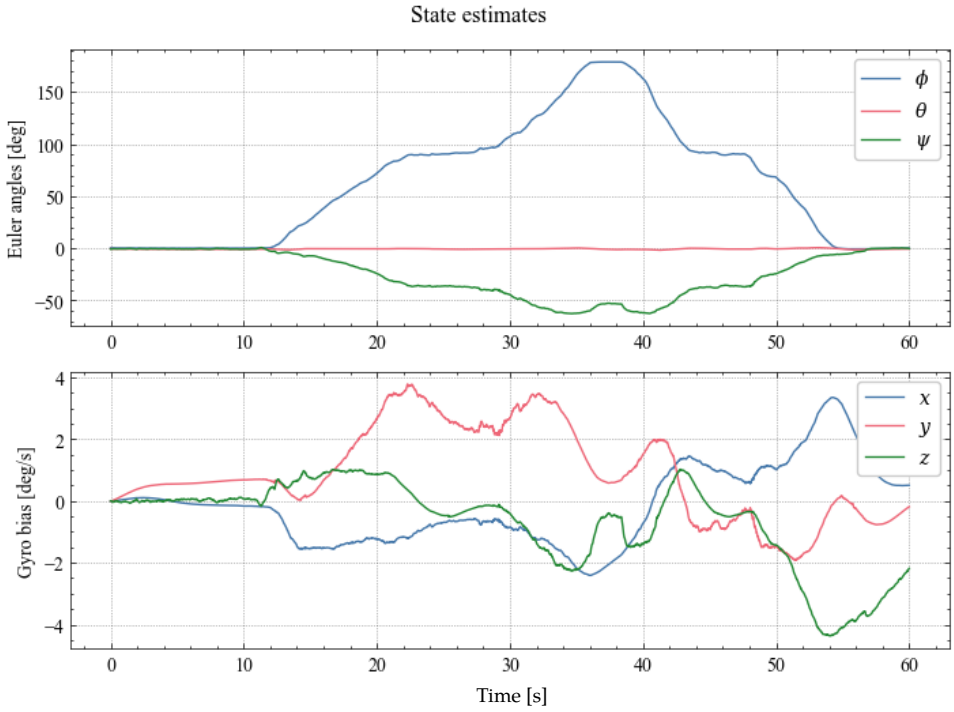

Figure 4.5: State estimates showcasing attitude and gyro bias for 180 degrees movement in positive roll direction and back to idle.

4.3 45, 90 and 180 degree pitch

The same procedure as for roll is done to demonstrate pitch. Here we would expect no movement in roll and yaw for Figure 4.6 and 4.7. This is not the case, and we get the same problem as for roll movement that the yaw angle yields the incorrect estimate. This error is larger for the pitch than for the roll. In Figure 4.8 we see that roll and yaw flip 180 degrees as expected when pitch move further than 90 degrees. The range for pitch is [-90, 90] and the range for roll and yaw is [-180, 180]. We still see an error in the yaw estimate, and this is correlated to the same problems as before.

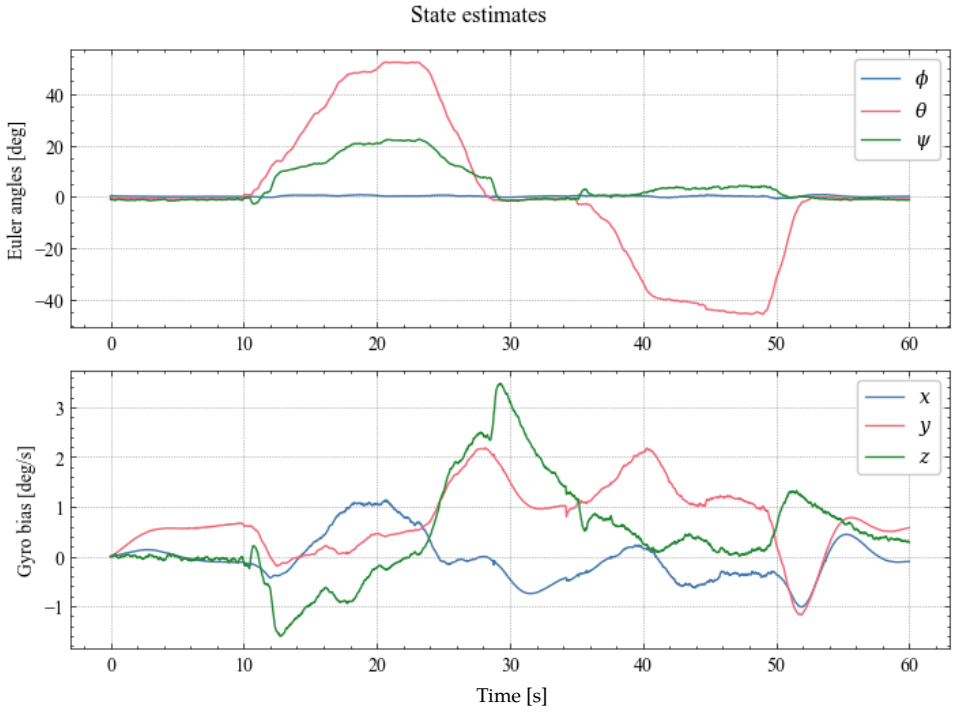

Figure 4.6: State estimates showcasing attitude and gyro bias for 45 degrees movement in positive and negative pitch direction and back to idle.

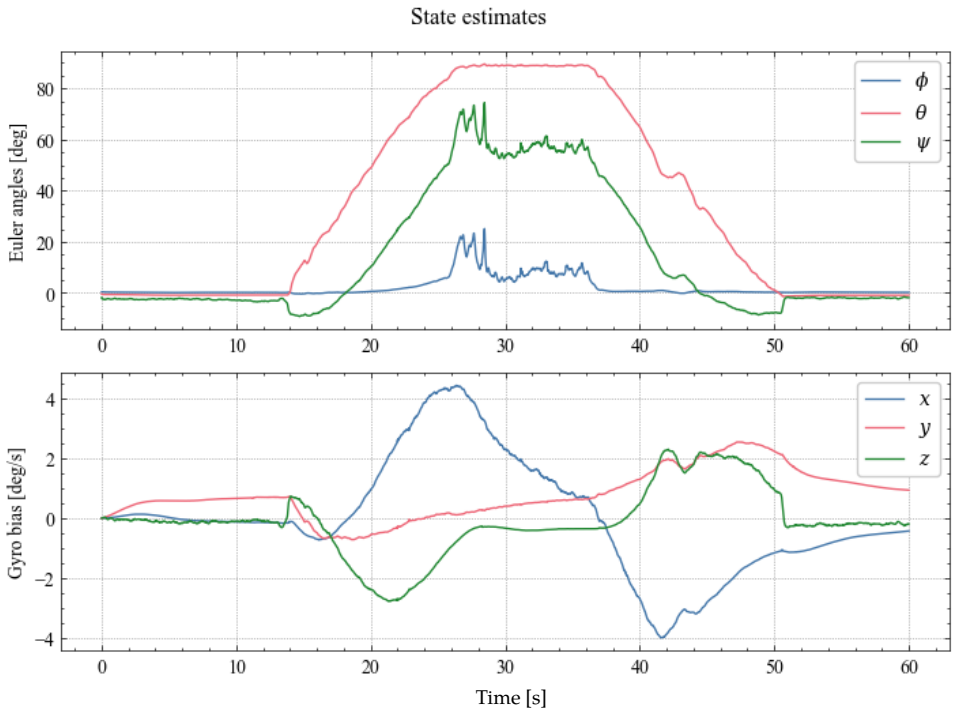

Figure 4.7: State estimates showcasing attitude and gyro bias for 90 degrees movement in positive pitch direction and back to idle.

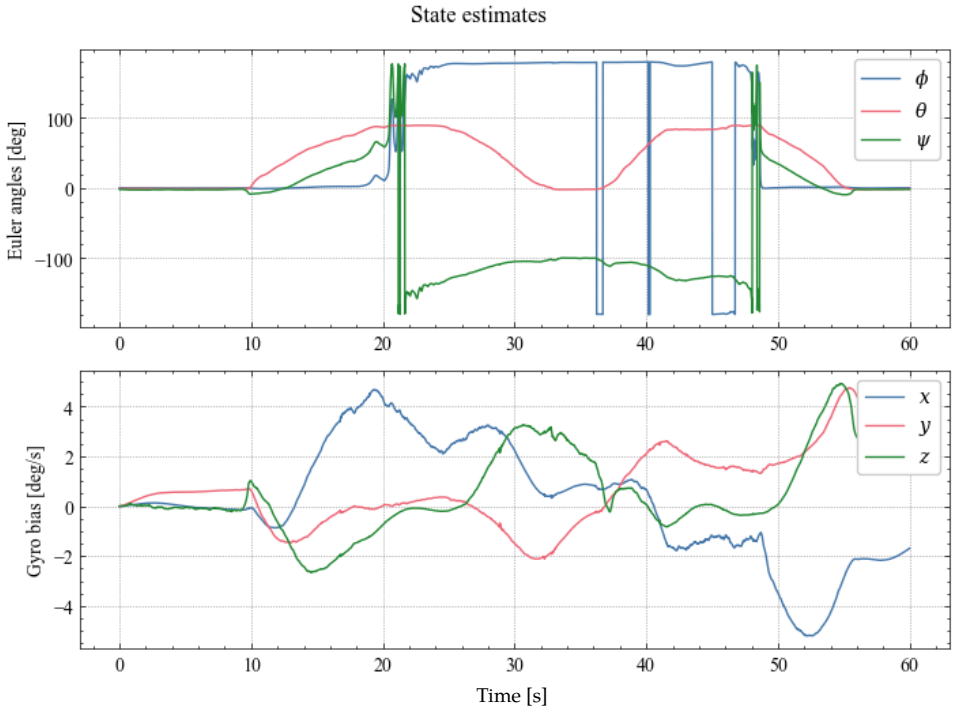

Figure 4.8: State estimates showcasing attitude and gyro bias for 180 degrees movement in positive pitch direction and back to idle.

4.4 45, 90, 180 degree yaw

When only conduction movement in yaw direction, the system struggles to follow the test movement. The yaw movement stops at 15 degrees or drops to negative value as shown in Figures 4.9 – 4.11. This is incorrect and is based on the same problem as described for idle, roll, and pitch.

When simulating the test shown in Figure 4.9, but with correction of yaw turned off after 10 seconds, we can observe a more accurate estimate of the movement. The result of this can be observed in Figure 4.12. Removing the error that the magnetometer measurement introduces when moving away from the idle position. On the bad side, it also introduces a new bias error in yaw that grows through the testing case.

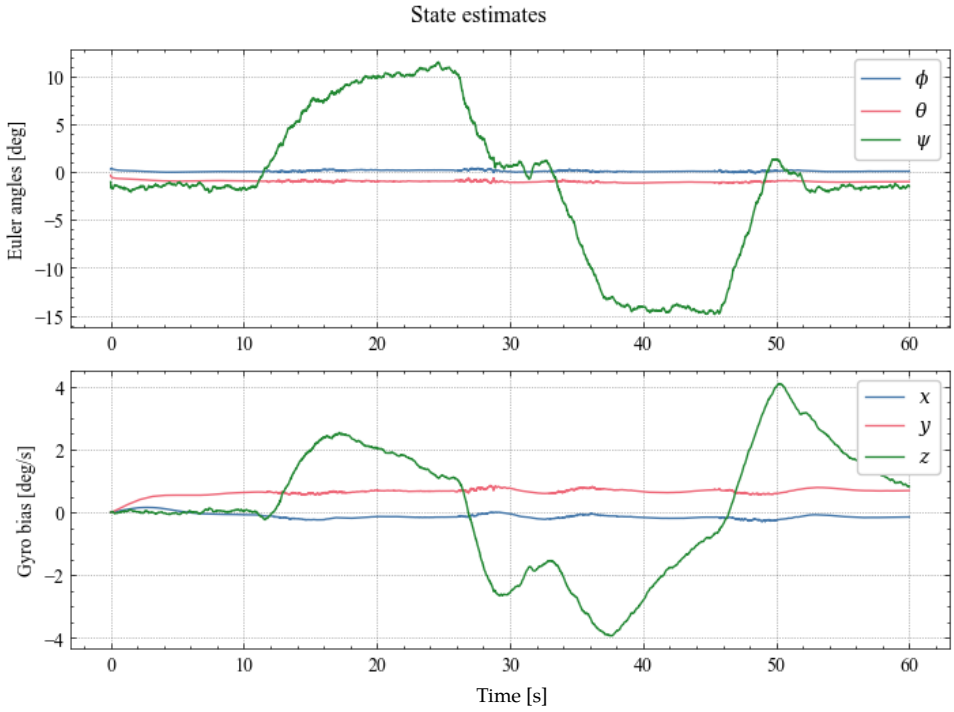

Figure 4.9: State estimates showcasing attitude and gyro bias for 45 degrees movement in positive and negative yaw direction and back to idle.

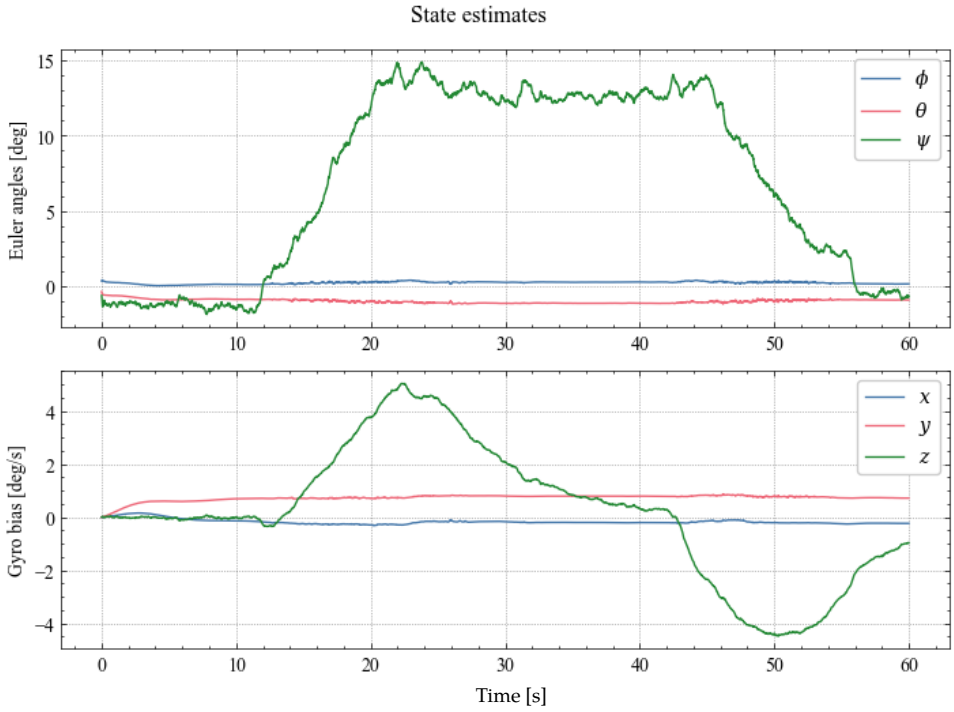

Figure 4.10: State estimates showcasing attitude and gyro bias for 90 degrees movement in positive yaw direction and back to idle.

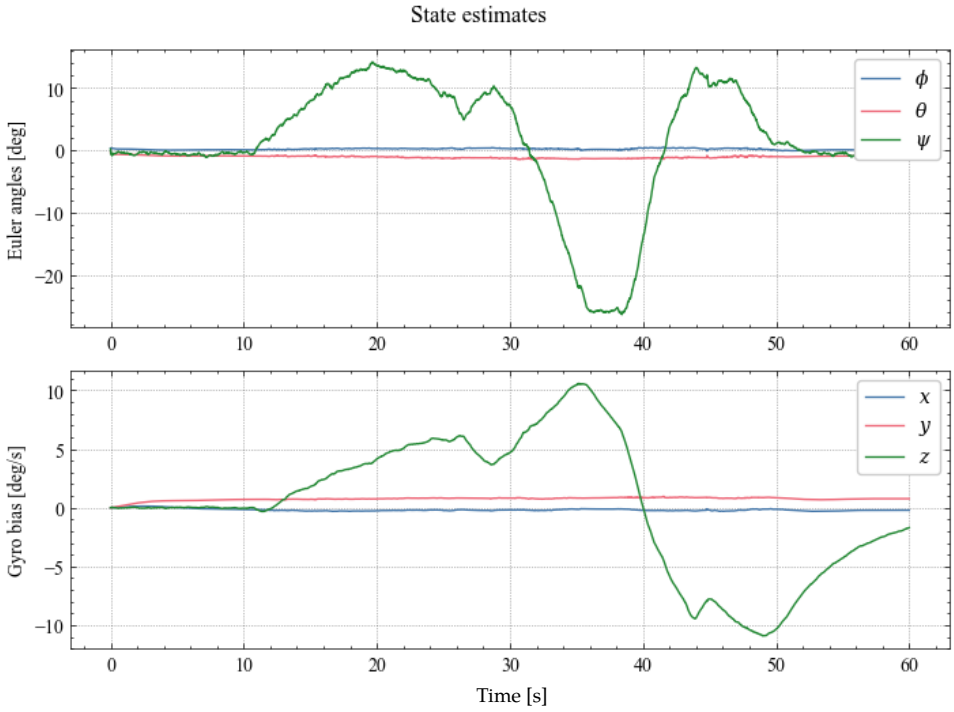

Figure 4.11: State estimates showcasing attitude and gyro bias for 180 degrees movement in positive yaw direction and back to idle.

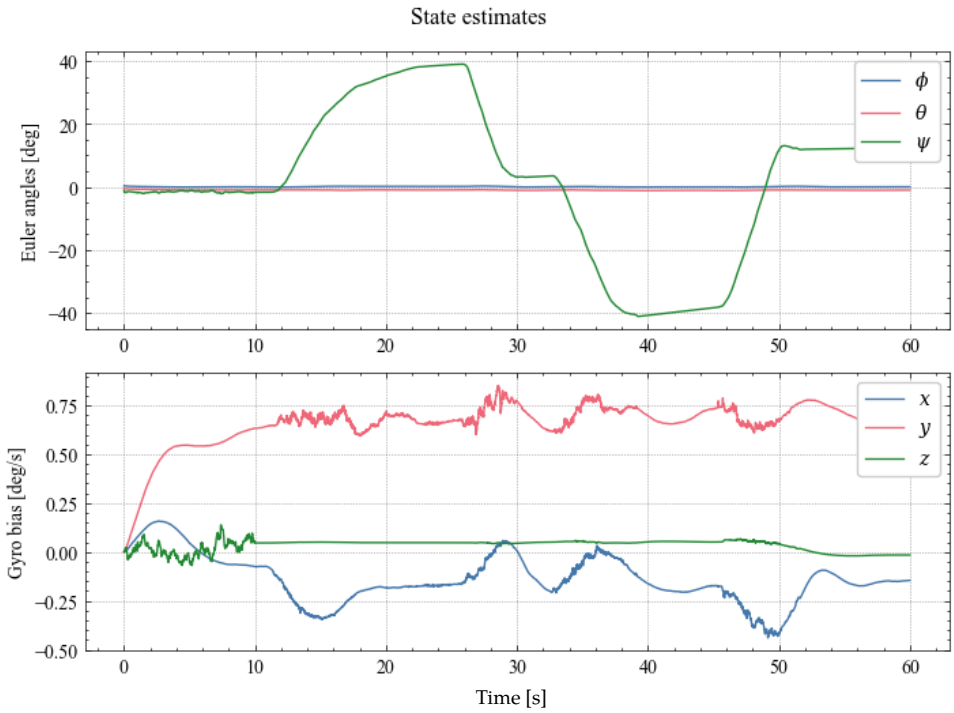

Figure 4.12: State estimates showcasing attitude and gyro bias for 45 degrees movement in the positive and negative yaw direction and back to idle. For this test, the magnetometer measurements were only used for correction in the first 10 seconds.

4.5 Mixing roll and pitch in same test

In this result, we combine multiple movements in the roll and yaw direction. In Figure 4.13 we first orientate 45 degrees in positive roll direction and then 45 degrees in positive pitch direction. This is illustrated with changes in the in roll and pitch for the estimated attitude. The same movement is also done in reverse back to idle. The result for pitch and yaw is correct, but we still get the same problem with changes in yaw. This is a repeating problem for all the results. In Figure 4.14 we have a negative orientation in the roll direction and a positive orientation in the pitch direction. For this scenario, the unwanted orientation in yaw is smaller than in Figure 4.13, but still not zero as wanted.

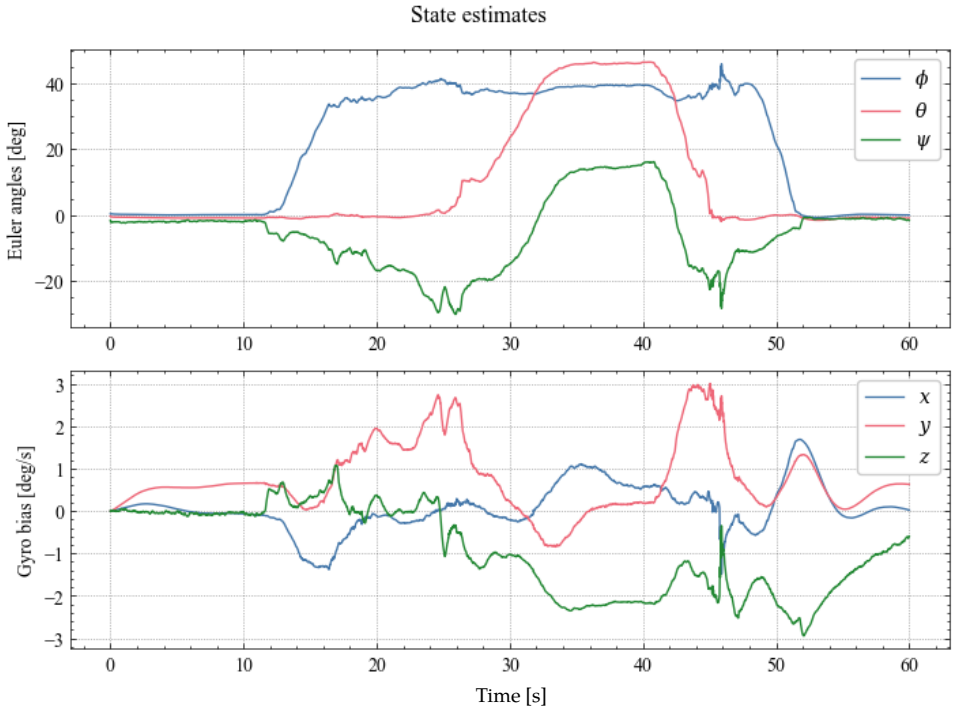

Figure 4.13: State estimates showcasing attitude and gyro bias for 45 degrees movement in positive roll and pitch direction and back to idle.

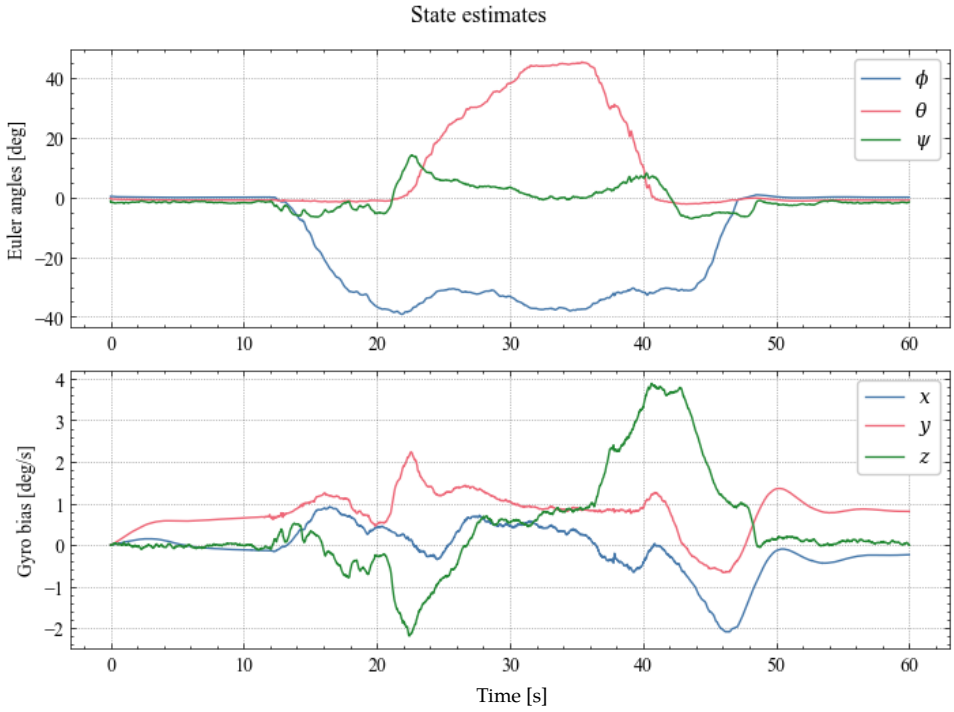

Figure 4.14: State estimates showcasing attitude and gyro bias for 45 degrees movement in negative roll and positive pitch direction and back to idle.

5 Discussion and conclusion

5.1 Discussion

A clear observation from the results is the less accurate estimate of yaw than for roll and pitch. Possible reasons behind this could be that the correction of yaw is based on the difference between the measured magnetic field vector and the estimated magnetic field vector (2.25). This method is not as accurate as for roll and pitch, where the Earth's gravity is used as a reference when estimating the error states. Another reason and most likely more influential are noise and bias in the magnetometer measurements that are not considered.

The errors in the yaw are clearly shown when we move away from the initial zero yaw value. If we turn off the correction based on the magnetometer, as done in Figure 4.12, we see that the estimate of yaw is closer to correct, but we also get a constant deviation in yaw. This indicates that the implementation struggles to stabilize the correct gyroscope bias about the z-axis when in an idle position. This conclusion is reinforced when looking at the variance for gyroscope about the z-axis when in idle position as shown in Figure 4.1. This variance bias is much larger than the variance of the gyroscope bias about the x- and y-axis. The estimate of yaw is, as mentioned before, a repeating problem for almost all of the results.

5.2 Conclusion

Based on the results, the estimation of roll and pitch is close to the actual movement conducted in the test, and yaw has much instability. This shows that the correction based on the accelerometer is far better than the correction based on the magnetic field as currently implemented. A lot of this comes from weaknesses in simple tuning and a possibility of not choosing the best initial values and parameters. My implementation of the error state Kalman filter also has room for improvements and optimization. Different sources of possible errors that primarily impact heading are discussed in Section 5.1.

The estimate for roll and pitch is close to the actual movements for almost all the results. Using the Earth's gravity as a reference yields a good comparison for the correct orientation for roll and pitch. We have the measured acceleration from the IMU's accelerometer and the expected acceleration based on vector $\mathbf{g}$ rotated with the given attitude of the system. This attitude is extracted from the quaternion in the nominal state as a rotation matrix. This works nicely because the magnitude of the Earth's gravity is dominant and makes minor deviations and errors negligible.

5.3 Further work

Further work for master thesis that will improve the estimate of the correct orientation. Many improvements can increase the accuracy of the estimated orientation for the circuit board. Some of these improvements will briefly be explained.

5.3.1 Allan variance based tuning

The datasheet for the LSM9DS1 IMU provided by STMicroelectronics (2015) supports only fundamental specifications. Information about bias stability, random walk, and noise are limited. This information can be estimated by collecting sample data in an idle position over a time period of about 12-24 hours. Then calculate the Allan variance (Farrell, 2008, ch. 4.9.2) based on the sampled data.

5.3.2 Estimate magnetometer bias

The IMU used in this thesis consist of an accelerometer, gyroscope, and magnetometer. For the states in the ESKF described in Section 2.4.1, only the gyroscope biases are estimated as states. By adding magnetometer bias in the x-, y-, and z-direction, we do not need to set a fixed synthetic north as a reference. This is important if we want to estimate the orientation based on real north. Adding this will also increase the states of the ESKF by three.

5.3.3 Estimate accelerometer bias

As seen in the result for an idle position in Figure 4.1, the estimated roll and pitch is not equal to zero. This could be improved by adding accelerometer bias in x-, y-, and z-directions as states in the ESKF. It should be said that a possible scenario for errors is that the table the circuit board was laying on when conducting the idle test was not completely leveled.

5.3.4 Calibration of magnetometer

The magnetic field is not equal for every position around the globe. The measurements from the magnetometer are also affected by the environment close to the IMU. Considering this, we need a procedure to calibrate the magnetometer on-site, where it will be used for a more accurate estimate of orientation in yaw. This should be a simple method that can be done in the initial phase every time the product will be used in a new location. Without this, it will not be possible to find the true north without providing a fixed reference.

5.3.5 Detailed calculation of every tuning parameter

To better adjust the parameters used in the error state Kalman filter, there should be more depth behind the choice of the current values. Trial and error can give good results, but complete calculation behind every value yields a better starting point for possible improvements and more correct attitude estimates.

5.3.6 Use multiple IMUs

As shown in Figure 1.2 we have five IMUs at our disposal. Accuracy for estimating correct orientation can be improved by collecting data from every sensor and compere this before final calculations. Gathering the sensor data from every IMU is already implemented in Python, but only data from the middle sensor is used in this thesis. This gives opportunities for further improvements.

5.3.7 Use GNSS with two antennas to correct heading

The magnetic field for an area is vulnerable to disturbances and will change based on location. These factors make the magnetometer in a cheap IMU potentially not the most reliable source for correction of heading in ESKF. Another solution for this could be to use a GNSS module with heading capability. An example for this is the mosaic-H from Septentrio (2021). With a dual-antenna input, this module can provide an estimated heading that can be used for correction in the ESKF. Optimal these two antennas should be mounted with 1-meter separation, but it is not enough space on the intended product from Squarehead Technology for that. Here it is only room for 30 cm of separation between the antennas. Further work could investigate how precise and reliable this approach is, given the short separation between the antennas.

References

Brekke, E. (2020). *Fundamentals of Sensor Fusion: Target tracking, navigation and SLAM*. Norwegian University of Science and Technology, third edition.

Farrell, J. A. (2008). *Aided Navigation: GPS with High Rate Sensors*. The McGraw-Hill Companies, first edition. doi: 10.1036/0071493298.

Septentrio (2021). mosaic-h – gnss module with heading capability. (Accessed: 21.06.2021).
URL: *https://www.septentrio.com/en/products/gnss-receivers/rover-base-receivers/receivers-modules/mosaic-h*

Solà, J. (2017). *Quaternion kinematics for the error-state Kalman filter*. Cornell University, v1 edition. (Accessed: 25.02.2021).
URL: *https://arxiv.org/pdf/1711.02508.pdf*

STMicroelectronics (2015). Inemo inertial module: 3d accelerometer, 3d gyroscope, 3d magnetometer. LSM9DS1 datasheet revision 3. (Accessed: 01.03.2021).
URL: *https://www.st.com/resource/en/datasheet/lsm9ds1.pdf*

Zinn, N. (2018). Characterizing an imu for a raspberry pi. (Accessed: 18.03.2021).
URL: *https://www.linkedin.com/pulse/characterizing-imu-raspberry-pi-noel-zinn/*