



UNIVERSITY OF SOUTHERN DENMARK
FACULTY OF ENGINEERING

Low-cost wearable body tracking

Bachelor's Thesis for the degree of
BSc in Engineering (Robot Systems)

Author:
Mikael Westermann
mikael.westermann@gmail.com

Supervisor:
Asst. Prof. Jimmy A. Jørgensen

June 2015

Author signature:

Date:

Mikael Westermann

DD – MM – YYYY

Abstract

This paper describes the development and experimental results of a system for reliable and robust tracking of human upper arm and forearm orientation using Inertial Measurement Units (IMUs) with magnetometers. The system is developed to provide accurate estimates of the anatomical orientation of the arm segments using sensor measurement error compensation and sensor-to-anatomy orientation calibration techniques. The possibilities of kinematic model calibration using position estimation from the IMU data are investigated. The methods and software implemented in this project have been developed with readability, simplicity and extensibility in mind. The error compensation and orientation techniques are shown to produce excellent results, while being easy to repeat and require minimal user involvement. The length of the upper arm is determined using position estimation through double integration of acceleration during oscillatory movement. The presented method requires moderate user involvement and is neither reliable nor highly repeatable. The developed software and methods, with the exception of the kinematic model calibration, are simple, yet satisfactory, and highly extensible. The system is shown to produce satisfactory estimates of anatomical orientation of the arm segments.

Preface

This thesis is the final work of the Bachelor of Science (BSc) in Engineering (Robot Systems) programme provided by the University of Southern Denmark (SDU). It has been carried out at the Faculty of Engineering (TEK).

I would like to thank my supervisor, Assistant Professor Jimmy A. Jørgensen, for his guidance and explanations. I would also like to thank fellow students Jens-Jakob Bentsen and Kristian H. Terkildsen for many valuable discussions on rotations and velcro. Finally, I would like to thank Assistant Professor Kjeld Jensen and fellow student Mathias Neerup for lending me an IMU, which has been essential to the experiments carried out during this project.

Mikael Westermann
Odense, June 2015

Contents

1	Introduction	1
1.1	Source code	1
2	Problem statement	2
3	Notation and abbreviations	3
4	Coordinate systems convention	4
4.1	The world frame	4
4.2	The sensor frame	5
4.3	The anatomy frame	6
4.4	Quaternions	6
5	Sensor calibration	7
5.1	Motivation	7
5.2	Sensor models	7
5.3	On-board compensation	10
5.4	Sensor calibration methods	10
6	Orientation tracking through sensor fusion	14
6.1	Sensor fusion	14
6.2	IMU Firmware	15
6.3	Data collection	16
6.4	Offline data processing in Matlab	16
6.5	Results	17
6.6	Discussion	17
6.7	Summary	18
7	Orientation calibration	23
7.1	Gravity-based method	23
7.2	PCA-based method	24
7.3	Data used for orientation calibration	24
7.4	Results	26
7.5	Discussion	27
7.6	Summary	27
8	Kinematic model calibration through position estimation	28
8.1	IMU position estimation	28
8.2	Upper arm length estimation	29
9	Discussion	29
10	Summary	32
11	Human body tracking results	33
12	Discussion	35
13	Conclusion	36
14	References	37

1 Introduction

The applications of wearable human body tracking systems are many. One example is the field of rehabilitation. Rehabilitation allows patients with reduced mobility to regain their mobility. For this, continuous monitoring of the patients' movements, achievable by human body tracking systems, is necessary [24]. Another field is that of sports applications. Analyses of the movements of an athlete can be used to improve the athlete's performance, and it has been shown in [10] that wearable sensor systems can be used to measure signals which indicate forces, torques and other features of the human body motion.

The advantages of wearable human body tracking systems over optical tracking systems include that the sensor does not need to be within any line of sight and does not need to be illuminated. Another advantage is that wearable human body tracking systems can be developed at very little costs. For example, 15 Inertial Measurement Units (IMUs) of the type used in this project [14], attached to different body segments, making up the main part of the body tracking system, would cost less than US \$ 1100.

This paper describes the development and experimental results of a low-cost wearable IMU-based system for reliable and robust tracking of human upper arm and forearm orientation. The main focus of this paper is laid on extensible software development, intrinsic sensor calibration and sensor-to-anatomy orientation calibration.

The kinematic model used in this project is defined as follows: The upper arm consists of a three degrees of freedom rotational ball-and-socket joint (the shoulder) and a link (the upper arm link) to the elbow joint. The elbow joint is modeled as having one degree of freedom, and the movement in this dimension is either flexion or extension. The elbow joint is directly connected to the radioulnar joint, which has one degree of freedom. This joint is connected to the wrist through the forearm link. Thus, the arm kinematic model has 5 degrees of freedom and two parameters: Upper arm length and forearm length.

Section 4 describes the relations between the coordinate systems used. Section 5 describes the development of sensor measurement error compensation software and methods, while section 7 describes the sensor-to-anatomy orientation calibration. Section 6 describes how IMU orientation tracking is achieved in this project, and section 8 describes how accelerometer measurements may be used to calibrate the kinematic model. Section 11 presents the human body tracking results of the calibrated system, while sections 12 and 13 are a summarizing discussion and conclusion respectively.

1.1 Source code

Source code and additional files can be found at the following url, [17]:

<https://github.com/madsherlock/Low-cost-wearable-body-tracking/>

2 Problem statement

The purpose of this project is to investigate how to perform reliable and robust tracking of human body parts using wearable inertial measurement units (IMUs). This problem is restricted to tracking of the upper arm and forearm of a human. The tracking will be performed using two IMUs, each equipped with a tri-axial accelerometer and a tri-axial gyroscope. The IMUs also have tri-axial magnetometers, and their usefulness with respect to the human arm tracking problem is investigated.

Desired features of the human body tracking are:

- High accuracy, reliability and repeatability.
- Robustness to sensor errors such as noise and bias.
- Minimal user setup: The person whose body parts are being tracked should be able to perform the data collection alone, and with minimal prior instruction.

Finally, in some applications, realtime orientation estimation could be desirable.

This project analyses the problems involved with human body tracking, their importance and their solutions. Different solutions of different complexity exist, and while complex, state-of-the-art implementations are best for human body tracking, simple solutions are fit better into the scope of this project and the explanations of the problems involved with human body tracking.

The software developed in this project should be easy to read and understand, and most importantly, extensible by others.

The main focus of this project will be on two types of calibration: Intrinsic sensor calibration and sensor-to-anatomy orientation calibration. Possibilities for calibration of the kinematic model using direct sensor position estimation will also be investigated.

3 Notation and abbreviations

IMU Inertial measurement unit

MARG sensor Magnetic, angular rate and gravity sensor

AHRS Attitude and heading reference system

PCA Principal Component Analysis

ω Angular velocity

s Linear position

v Linear velocity

a Linear acceleration

H Magnetic field strength

T Temperature

Note that while the term MARG sensor always covers a sensor which measures magnetic field, angular velocity and linear acceleration, the term IMU can mean a sensor which measures all three or a sensor which measures only angular velocity and linear acceleration. The sensors in this project can act as both IMUs and as MARG sensors, and the main term used when referring to them is IMU, as this is the term used by the manufacturer, [14]. Terms like tri-axial gyroscope are abbreviated to gyroscope or just gyro. Two IMUs were used in this project, and each was given an arbitrary board ID. One sensor has the ID 'bi' and is referred to as the 'bi' IMU, while the other sensor has the ID 'kj' and is referred to as the 'kj' IMU.

4 Coordinate systems convention

In this paper, different right-handed coordinate systems are used for representation of orientations and positions, and an overview of their uses will be given in this section. Many different coordinate and rotation conventions exist, and it is easy to get confused when converting between them, especially when trying to merge different works within the field of orientation tracking.

Rotations about the axes of a fixed coordinate system (where the axes do not change during rotation) are referred to as extrinsic rotations. Rotations where the coordinate system axes change after each elemental rotation are referred to as intrinsic rotations.

Three coordinate frames, explained below, are used in the representation of orientations in this project. These are the world frame, C_W , the sensor frame, C_S , and the anatomy frame, C_A , denoted as in equations 4.1, 4.2 and 4.3.

$$C_W : [X_W, Y_W, Z_W]^T \quad (4.1)$$

$$C_S : [X_S, Y_S, Z_S]^T \quad (4.2)$$

$$C_A : [X_A, Y_A, Z_A]^T \quad (4.3)$$

The relation between the different coordinate frames in the context of forearm orientation tracking can be seen on figure 4.1. Note that the world frame origin does not correspond to the one chosen for orientation visualization, which is chosen as the center of the shoulder joint. Note also that the Z_A and Z_W axes are not necessarily aligned as they are in figure 4.1.

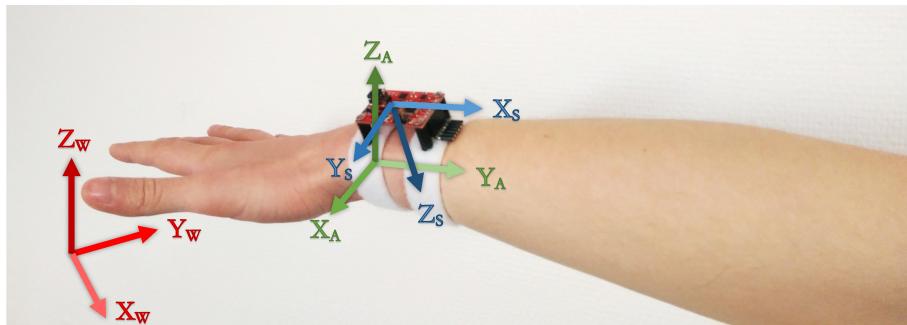


Figure 4.1: Three different coordinate frames in the context of forearm orientation tracking. The pictured origins of the anatomy frame and sensor frame correspond to the actual axes tracked, while the origin of the world frame does not correspond to the actual origin used for tracking (the shoulder joint center).

4.1 The world frame

The world frame, here also called the static frame, fixed frame or Earth frame is defined according to the Earth: The axis referred to as the vertical axis is aligned with the direction of gravity. The plane perpendicular to this axis is the horizontal plane. Commonly used right-handed coordinate systems in this frame are North East Down (NED), also known as local tangent plane (LTP), and East North Up (ENU). Rotations expressed as elemental rotations around the axes of these coordinate systems are typically extrinsic.

Angles in the world frame that happen in three distinct axes are often referred to as pitch, roll and yaw (Tait-Bryan angles). A rotation expressed in terms of pitch, roll and yaw angles can be

intrinsic or extrinsic. A convention sometimes used in IMU context is z-y'-x'' (intrinsic), where the z-axis is the vertical axis.

In the convention used in this paper and the main part of the literature referred to in this paper, yaw is the angle around the vertical axis, that is, the angle typically measured by a compass. This is also known as the azimuth. Pitch and roll angles measure tilt relative to the horizontal plane, and one example of a Tait-Bryan convention that is typically used in the context of an IMU is the aircraft principal axes, see [18].

If the aircraft principal axes are defined as following the z-y'-x'' convention mentioned above, the yaw axis is the z-axis aligned with the vertical axis of the world frame, and is pointing downwards. The x'' axis points forward (from the back of the plane to its nose) along the center of the airplane, and a rotation about this axis is called roll. The "up and down" tilting of the nose of the plane (a rotation about the y'-axis) changes the pitch.

The yaw reference angle, defining the fixed frame roll and pitch axes directions, can be chosen arbitrarily or it can be chosen as magnetic north, which is typically done when using a magnetometer for measuring the direction of Earth's magnetic field (as it requires fewer conversions between coordinate systems). In this paper, that is exactly how the yaw reference angle is chosen: When a magnetic reference is available, the yaw angle is defined according to this reference. When the magnetometer is unused and no magnetic reference is available, the initial yaw angle reference is chosen arbitrarily. In this project, the choice of reference yaw angle does not affect the human body orientation tracking, except when it comes to the issue of three-dimensional (3D) visualization, where the "camera angle" should be chosen according to the reference yaw angle.

4.2 The sensor frame

The sensors used in this project each have their own coordinate frame, denoted the sensor frame. This frame is relative to the sensor. The angular rate, acceleration and magnetic flux data collected from the sensors are all represented in the sensor frame, see figure 4.2. Thus, if the sensor is placed horizontally so that the sensor z-axis is parallel with the gravitational acceleration, the accelerometer is expected to output 1 g in the z direction and 0 g in the x and y directions.

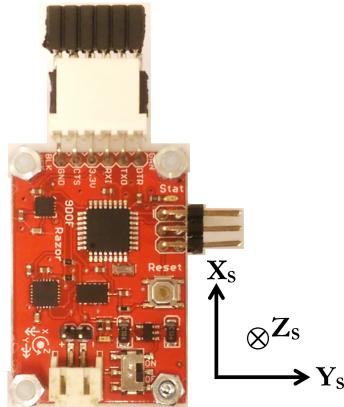


Figure 4.2: SparkFun 9 Degrees of Freedom Razor IMU. The coordinate system used in the sensor frame is shown with black arrows (right), not to be confused with the unused white arrows printed on the circuit board which are not used.

The three-angle coordinate representations (in some cases called Euler angles, in other cases

called Tait-Bryan angles) in Earth's frame have singular properties which can lead to the problem known as "Gimbal lock", [19]. However, the intuitiveness of the Euler angle and yaw-pitch-roll (a Tait-Bryan variant) representations means that they will be used for some parts of this project. For the main part, however, rotations will be represented by quaternions, explained below.

4.3 The anatomy frame

While the sensors measure sensor orientation, human body motion is tracked in the anatomy frame. In this project, the anatomy frame is defined from the radioulnar joint axis and the rotational axis of the upper arm. Ideally, the absolute position of the tracked body parts during a rotation about the anatomical axes (approximately aligned with the bones), should remain the same. The anatomy frames of the forearm and upper arm have their Y-axes aligned with the rotational axes. The X-axes are chosen to lay in the horizontal plane when the arm is held horizontally with the palm facing down. This leaves the choice of Z-axes as the cross-products of the respective pairs of X- and Y-axes, and this means it will point upwards when the arm is held horizontally with the palm facing down. See figure 4.1 for a visualization of the forearm anatomy frame axes.

4.4 Quaternions

Quaternions are hyper-complex numbers capable of describing rotations in three-dimensional space [20]. As each quaternion is defined by four elements, it is a more compact rotation representation than rotation matrices (which have nine elements), and they don't suffer from the problem of Gimbal lock. Quaternions are arguably less intuitive than other rotation representations, but an attempt on an intuitive explanation from a practical point of view is given below.

A unit quaternion \vec{q} of length 1 representing a rotation at an angle α about an axis $\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$, is calculated as in equations 4.4 and 4.5.

$$\vec{q}_1 = \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\alpha/2) \\ x_1 \sin(\alpha/2) \\ y_1 \sin(\alpha/2) \\ z_1 \sin(\alpha/2) \end{bmatrix} \quad (4.4)$$

$$\vec{q} = \frac{\vec{q}_1}{\|\vec{q}_1\|} \quad (4.5)$$

As an example, a $+180^\circ$ rotation about the x-axis is quickly found as the quaternion $q = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$.

5 Sensor calibration

For this project, two Sparkfun 9 Degrees of Freedom Razor IMUs [14] were used. These sensors each consist of an ADXL345 tri-axial accelerometer [1], an ITG-3200 MEMS tri-axial gyroscope with internal thermometer [8] and a HMC5883L tri-axial magnetometer [7]. Note that these sensors provide nine degrees of measurement (and a temperature measurement) within the six degrees of freedom. The accelerometer provides translation measurements while the gyroscope and magnetometer provide rotation measurements.

Each of these sensors have sensor-specific errors which can be compensated for through calibration. This section describes how this intrinsic sensor calibration is carried out in this project. The choice of calibration methods described below is based on the idea that the sensor-specific error compensation should be carried out on the sensor itself, and that the calibration parameters should be stored on the sensor. Furthermore, these parameters should be variables which can be changed, so that the sensors could have the same firmware. Lastly, the parameters should persist through firmware upgrades.

The calibration software, which resides mainly on the IMUs, is heavily based on the Attitude and Heading Reference System (AHRS) firmware in [3], which is developed for the same type of IMU. The calibration procedures are therefore also based on the procedures of [4, Sensor calibration].

5.1 Motivation

The gyroscope measures angular velocity. If the gyroscope measurements are to be used for orientation tracking, these measurements must be integrated. Since the gyroscope measurement errors are on the angular velocity, an integration of gyroscope measurements would also mean an integration of the measurement errors. If the errors have a bias, this bias would mean an undesirable drift in orientation. This outlines the primary reason to perform the compensation on the gyroscope data.

The accelerometer measures linear acceleration, including the gravitational acceleration. One of the purposes of accelerometers in orientation tracking is to give a gravity reference vector. This means that when the accelerometer is not moving, the accelerometer measurement should be equal to the gravitational acceleration vector relative to the sensor. Errors on the measurement would mean errors on the gravitational acceleration vector, which reduces the usefulness of the sensor in the sense of orientation tracking. This is one reason to compensate for errors, especially offsets, on the accelerometer data.

The magnetometer measures the strength and direction of the magnetic field. An important property of the magnetometer in orientation tracking is its ability to measure earth's magnetic field as a reference to the orientation on the vertical axis in world coordinates. However, as described below, magnetic field disturbances can offset the magnetometer data and change the scaling of the axes. This leads to errors on the measured magnetic field, which may then yield a wrong orientation reference. This motivates magnetometer calibration.

5.2 Sensor models

This section describes the sensor models used in this project.

Note that although several sensors may be modeled as temperature dependent, only the model

of the gyroscope, which has an internal thermometer, includes temperature. The magnetometer has been shown to also be temperature dependent, but the complexity of the temperature dependent error compensation for the magnetometer, as it is described in [16], was judged too great considering that the thermometer is located inside the gyroscope sensor, and not the magnetometer: The sensors would have to have very similar thermal characteristics, and this compensation would be a sort of sensor fusion.

5.2.1 Gyroscope

The gyroscope sensor model used in this project is relatively simple. As the sensors are very acceleration tolerant [8], linear acceleration effects are assumed negligible and are not included in the model. The measured angular velocities $\vec{\omega}_{meas}$ are modeled as in equation 5.1.

$$\vec{\omega}_{meas} = \begin{bmatrix} \omega_{x,meas} \\ \omega_{y,meas} \\ \omega_{z,meas} \end{bmatrix} = \mathbf{S}_{gyro} (\vec{\omega}_{sensor} + \vec{\omega}_{bias}(T)) + \vec{noise} \quad (5.1)$$

Here, \mathbf{S}_{gyro} is a 3-by-3 matrix with scaling factors as diagonal elements and misalignment errors included in the nondiagonal elements. $\vec{\omega}_{sensor}$ is the true angular velocity of the sensor, $\vec{\omega}_{bias}(T)$ is a vector of gyroscope biases (also known as Zero-Rate Output), which depend on temperature T , and \vec{noise} is sensor noise. To reduce the complexity of the calibration routine, it is assumed that all axes of the gyroscope have the same gain as per the datasheet [8] and that there are no misalignment errors. This is a simplification on \mathbf{S}_{gyro} , as shown in equation 5.2, where S_{gyro} is a scalar gain applied to all axes.

$$\mathbf{S}_{gyro} = S_{gyro} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

The cross-axis sensitivity listed in the datasheet is 2 %, which fits well with the assumption. However, one way to improve the gyroscope model could be to determine the scale factors experimentally, as they have a tolerance of ± 6 % at room temperature. Furthermore, the sensor scale factors vary typically ± 10 % with temperature. To determine the scale factors and their thermal characteristics experimentally, it is necessary to have an angular velocity reference (as an example, the sensor could be mounted on a rotating DC motor with rotary encoders) as well as a means to control the temperature. This kind of experiment has not been carried out in this project due to time constraints and otherwise good tracking results.

The gyroscope bias model is shown in equations 5.3 and 5.4, where T_0 is the working temperature at which the bias is $\vec{\omega}_{bias,T_0}$. The temperature dependencies are modeled as linear and are included in $\frac{d\vec{\omega}_{bias}}{dT}$.

$$\vec{\omega}_{bias}(T) = \vec{\omega}_{bias,T_0} + \frac{d\vec{\omega}_{bias}(T)}{dT} \Delta T \quad (5.3)$$

$$\Delta T = T - T_0 \quad (5.4)$$

The error-compensated gyroscope measurements $\vec{\omega}_{comp}(T)$, which are noise and true measurements, can be computed by equation 5.5.

$$\vec{\omega}_{comp}(T) = \frac{1}{S_{gyro}} (\vec{\omega}_{meas} - \vec{\omega}_{bias}(T)) \quad (5.5)$$

5.2.2 Accelerometer

The accelerometer sensor model used in this project is also simple. The measured accelerations \vec{a}_{meas} are modeled as in equation 5.6.

$$\vec{a}_{meas} = \begin{bmatrix} a_{x,meas} \\ a_{y,meas} \\ a_{z,meas} \end{bmatrix} = \mathbf{S}_{acc} (\vec{a}_{sensor} + \vec{a}_{offset}) + \vec{noise} \quad (5.6)$$

Here, \mathbf{S}_{acc} is a 3-by-3 matrix analogous to \mathbf{S}_{gyro} , of scaling factors and misalignment terms. \vec{a}_{sensor} is the true linear acceleration of the sensor and \vec{a}_{offset} is the offset on each axis. As in the gyroscope model, here it is assumed that misalignment errors are negligible. However, the scale factors of \mathbf{S}_{acc} are relatively easy to measure, and are included in the model. Thus, \mathbf{S}_{acc} is given by equation 5.7.

$$\mathbf{S}_{acc} = \begin{bmatrix} S_{x,acc} & 0 & 0 \\ 0 & S_{y,acc} & 0 \\ 0 & 0 & S_{z,acc} \end{bmatrix} \quad (5.7)$$

The error-compensated accelerometer measurements \vec{a}_{comp} can be computed by equation 5.8.

$$\vec{a}_{comp} = \begin{bmatrix} \frac{1}{S_{x,acc}} & 0 & 0 \\ 0 & \frac{1}{S_{y,acc}} & 0 \\ 0 & 0 & \frac{1}{S_{z,acc}} \end{bmatrix} (\vec{a}_{meas} - \vec{a}_{offset}) \quad (5.8)$$

5.2.3 Magnetometer

The magnetometer sensor model is similar to the accelerometer model in that it includes bias and scaling factors. However, magnetic field distortions are included in the model, as the body tracking will likely take place in environments with magnetic field distortions. As explained in [16], the magnetic measurement distortions fall in two categories: hard and soft iron. Hard iron distortions are caused by external magnetic fields, and soft iron distortions are caused by ferromagnetic materials. A hard iron distortion places a bias on the magnetic field, while a soft iron distortion scales the magnetic field in the sensor axes. Thus, the magnetic field distortions can be directly included in a model similar to that of the accelerometer model, as they simply add additional offset (bias) and scale factors. The magnetometer measurements \vec{H}_{meas} are modeled as in equation 5.9.

$$\vec{H}_{meas} = \begin{bmatrix} H_{x,meas} \\ H_{y,meas} \\ H_{z,meas} \end{bmatrix} = \mathbf{S}_{mag} (\vec{H}_{sensor} + \vec{H}_{offset}) + \vec{noise} \quad (5.9)$$

It is chosen to include all scale factors and offsets, including the distortions, in the calibration parameters. This reduces the number of parameters and simplifies the compensation to equation 5.10.

$$\vec{H}_{comp} = \mathbf{S}_{mag}^* (\vec{H}_{meas} - \vec{H}_{offset}^*) \quad (5.10)$$

\mathbf{S}_{mag}^* and \vec{H}_{offset}^* are 12 sensor- and environment-specific calibration parameters which include hard and soft iron distortions. Note that they are not equal to \mathbf{S}_{mag} and \vec{H}_{offset} .

5.3 On-board compensation

As stated above, the calibration software is based on the firmware in [3]. This firmware is written in Arduino and compensates for errors using the models described above, with the exception that gyroscope bias is modeled constant. All calibration values are hardcoded into the firmware.

Several changes have been made to the Arduino code, to include gyroscope bias temperature compensation and allow parameter changes without modifying the firmware. Through a serial interface, the user can select which compensations to apply. This removes the need for sensor calibration prior to data collection, and allows for use of other error compensation techniques than the ones described above. As an example, one might replace the simple temperature compensation with a third-order polynomial temperature compensation.

The modified firmware utilizes the EEPROM memory, which can be written to and read from while the sensor firmware is running. An arduino library named Board_storage has been developed. It implements the functionality required to store and read calibration parameters as 32-bit floating point numbers to and from EEPROM. This library also allows the storage of a board ID, a two-byte IMU identifier. This facilitates unique identification of each sensor in a collection of sensors (eg. 15 IMUs embedded in a full body suit).

5.4 Sensor calibration methods

5.4.1 Accelerometer

The accelerometer is calibrated using the serial interface implemented by the IMU firmware. The procedure listed below is carried out for each of the three axes:

1. Align sensor so that, on the current axis, the maximal measurement is recorded. This means placing the sensor on one of its sides.
2. Hold sensor completely still so that it only measures gravitational acceleration.
3. Record the axis maximum measurement.
4. Flip the sensor to the other side and repeat steps 1 through 3 to record the axis minimum measurement.

This simple procedure yields minimum and maximum acceleration values, \vec{a}_{min} and \vec{a}_{max} , measured on each axis, in sensor units, for a gravitational acceleration reference. From these values, the accelerometer calibration parameters are found using equations 5.11 and 5.12.

$$\vec{a}_{offset} = \frac{1}{2} (\vec{a}_{min} + \vec{a}_{max}) \quad (5.11)$$

$$\begin{bmatrix} S_{x,acc} \\ S_{y,acc} \\ S_{z,acc} \end{bmatrix} = \frac{1}{g_{ref}} (\vec{a}_{max} - \vec{a}_{offset}) \quad (5.12)$$

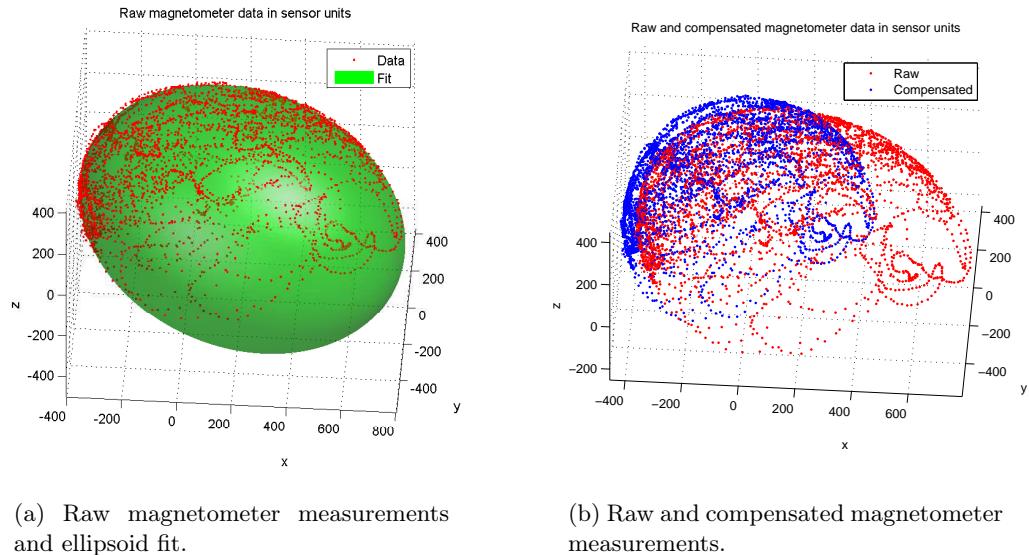
g_{ref} is 1 g (the magnitude of the gravitational acceleration) measured in sensor units. According to the datasheet [1], this value is 256.0.

5.4.2 Magnetometer

The magnetometer is calibrated using a Processing sketch file based on the one in [3]. The Processing sketch connects to the IMU through the serial interface and starts logging raw sensor data. The magnetometer data is visualized in a threedimensional coordinate system coinciding with the magnetometer axes. The IMU is placed in many different orientations so that an ellipsoid forms on the screen.

The magnetometer data is fit to an ellipsoid using an adaption of [11]. This algorithm returns the three center coordinates and radii of the ellipsoid fit, along with its three eigenvectors (directions of the ellipsoid radii). The center coordinates are equal to \tilde{H}_{offset}^* . The eigenvectors give a transformation from magnetometer coordinate system axes to ellipsoid axes. Their inverse gives the inverse transformation. In the Processing sketch algorithm of [3], these properties are used to generate a compensation matrix equal to \mathbf{S}_{mag}^* , from the ellipsoid radii and eigenvectors.

See figure 5.1 for an example of the results of a magnetometer calibration. This calibration was carried out on one of the IMUs ('kj') used in this project, and figure 5.1b clearly shows the impact of magnetometer error compensation on the collected data.



(a) Raw magnetometer measurements and ellipsoid fit.

(b) Raw and compensated magnetometer measurements.

Figure 5.1: Magnetometer calibration example. Data are in sensor units and measure magnetic field strength in each of the magnetometer axes. The data was generated by sampling the magnetometer in different orientations while the IMU was attached to the upper arm of a human in an indoor environment.

Only minor modifications to the Processing sketch have been made in this project, and are primarily adaptions to the modified sensor firmware. One of the changes include the recording of temperature (using the gyroscope thermometer) during magnetometer calibration, so that average calibration temperature is saved alongside magnetometer calibration parameters. Theoretically, the thermal characteristics of the magnetometer calibration parameters could be found by calibrating the magnetometer at different temperatures using the modified Processing sketch. At the very least, one could switch between a small number of different calibration parameter sets, based on the thermal environment (eg. is the sensor mounted directly on skin or on the outside of a helmet).

To get the best measurements without temperature compensation, it is recommended to calibrate the magnetometer directly before use, after letting it reach a constant temperature. Alternatively,

one could measure raw magnetometer data and ensure that the magnetometer is rotated enough during the tracking measurements for an offline ellipsoid fit.

5.4.3 Gyroscope

The purpose of the gyroscope calibration method is to find parameters $\vec{\omega}_{bias,T_0}$ and $\frac{d\vec{\omega}_{bias}}{dT}$. The calibration procedure is very simple:

1. Record raw gyroscope and thermometer data of the static sensor over varying temperatures (within the -40 °C to 105 °C maximum rating, [8]).
2. Select a working temperature T_0 .
3. For each gyroscope axis $i = x,y,z$, fit a first order polynomial (using ordinary least squares) to the gyroscope measurements vs. temperature to get $\frac{d\omega_{i,bias}}{dT}$, and evaluate the fit at T_0 to get $\omega_{i,bias,T_0}$.

If it is chosen not to use temperature compensation, the above procedure simplifies to averaging the collected sensor data on each axis, for a constant temperature T_0 , to get $\vec{\omega}_{bias,T_0}$.

Figure 5.2 shows the results of two gyroscope calibration experiments, one for each IMU. It is clearly seen from the figure, that there is a significant gyroscope bias of up to 3 °/s, and that it varies with temperature. The figure also shows that the bias is different for all axes and IMUs tested. Based on the figure, the assumption of linear temperature dependency seems to be justified for all but the x-axis on the 'bi' IMU, which curves slightly. This curve is assumed negligible for the purposes of this project. However, the curve indicates that a higher-order polynomial model might be a better approximation for general gyroscope bias temperature dependency (the high linearity of the two biases measured here might be a coincidence). Note that the measurements are discretized and that this causes noticeable jumps on the vertical axis.

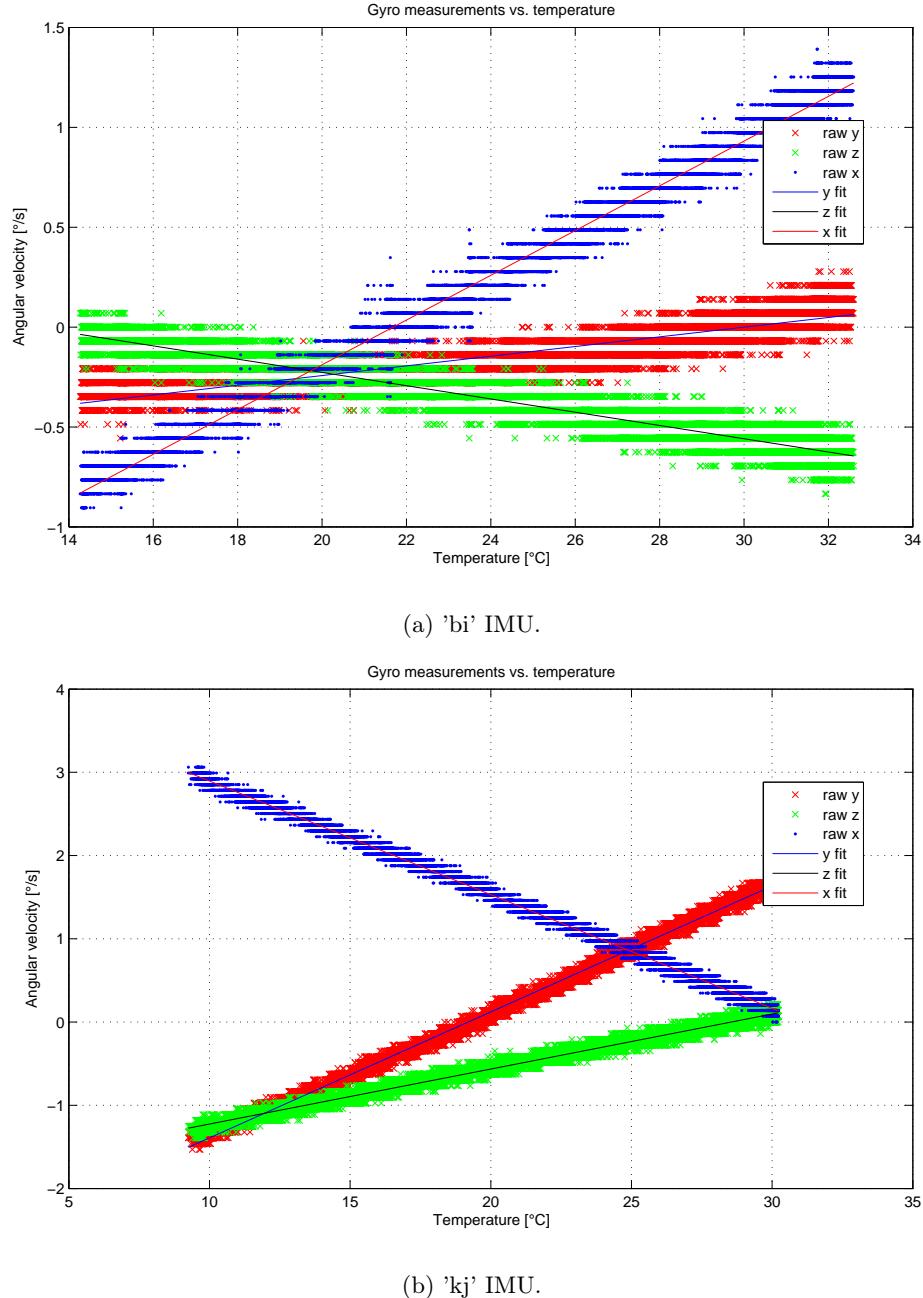


Figure 5.2: Gyro measurements vs. temperature. The IMUs were held still throughout the experiments, so the measurements shown are the Zero-Rate Outputs of the IMUs. The noticeable vertical "jumps" are due to the measurement discretization.

6 Orientation tracking through sensor fusion

The tracking of human body parts using IMUs can be accomplished in many different ways. It has been shown in [5] that very accurate shoulder, elbow and radioulnar joint angle tracking in a setup similar to the one in this project is possible. In [5], the RMS angle error was less than 8° for shoulder and elbow angles with respect to an optical reference system. In [5], the Unscented Kalman Filter (UKF) was implemented for nonlinear state estimation of the joint angles of an arm using data collected from two IMUs at 128 Hz. The state space model proposed in [5] is generic, meaning it can be applied to any joint angle, and orientation tracking is possible if the forward kinematics are known. In [23] the realtime implementation and testing results of a Kalman filter for human body motion are presented, and it is shown that MARG sensors are able to provide satisfactory human body tracking. However, as shown in [2], variations in the direction of the magnetic field in indoor environments with electrical heaters, metal furniture etc. can lead to yaw angle (azimuth) estimation errors of up to 16°. There, it is also shown that they might be avoided if a separation of approximately two feet between sensors and magnetic interferences is maintained. This could complicate user setup in indoor environments, and even more so when tracking orientation of body parts with ferromagnetic prostheses (such as in prosthetic amputee rehabilitation applications).

The approach chosen here is simple: Tracking the orientation of each body part using only data from the IMU attached to it, using an algorithm that is simpler and more time-efficient than a Kalman filter. This approach is not optimal, and is expected to perform with a small loss of accuracy compared to a complete Kalman filter approach.

This section describes tracking of the IMU's orientation in the world frame, relating the sensor frame C_S to C_W and vice versa.

The IMUs are strapped on tight with velcro, the rough side towards the skin, so that the sensor does not change orientation with respect to the anatomy frame.

6.1 Sensor fusion

Traditionally, orientation tracking has been performed through integration of the angular velocities measured by gyroscopes. It has been shown in [6] that the problems of orientation tracking using only gyroscopes has the problems of sensor white noise, drift and calibration errors looking like temporary bias errors while turning. Accelerometers can be used to compensate for this, as they measure the gravitational acceleration. This measurement gives a drift-free estimate of the direction of the vertical axis. Thus, it can be used as a reference orientation in the two horizontal axes.

IMUs with gyroscopes and accelerometers can be used to calculate drift-free orientation in the two horizontal axes (known as tilt, or roll and pitch), assuming that the accelerometer measures gravity only. This still leaves a drifting orientation around the vertical axis, however. This is where magnetometers provide a reference orientation, as the measured magnetic field has a horizontal component (unless the sensor is placed on a magnetic pole). However, the magnetometer provides a magnetic field reference vector in the sensor frame, and the direction of these magnetic field lines vary with geographic location. Thus, the measured vector needs to be projected onto the horizontal plane before it is a horizontal reference vector. This projection in turn needs another reference in Earth's frame, such as the tilt reference obtained from the accelerometer measuring the direction of gravity. Thus, the gravity vector from the accelerometer can be used to compensate for tilt on the magnetometer measurements, so that the magnetometer

measurements provide a horizontal reference vector independent of geographical location.

6.1.1 Madgwick's filter

Succesful implementation results of orientation tracking filters (two version of Madgwick's filter) with and without magnetometers are presented in [12]. These filters are based on a Gradient Descent method for orientation estimation. The source code for these filters is publicly available from [21]. These filters compensate for magnetic distortions by normalizing the measured magnetic field vector to have components only in the X_W and Z_W directions, meaning that errors in inclination are removed and only the yaw angle can be distorted by magnetometer measurements, [12].

The algorithm in [12] calculates orientation by numerically integrating the estimated rate of change of the orientation. This rate of change is calculated as the gyroscope measurements, minus an error term estimated from the accelerometer and magnetometer measurements.

Because these filters demonstrate the advantages of sensor fusion in simple, separate steps, and because the filters are freely available and have proven to perform well, [12], the filters available from [21] are used for orientation tracking in this project. Finally, one advantage of using these filters is that implementations are provided for several platforms, including a Matlab script and an optimized C-version for realtime applications. For details on the algorithm, see [12] and the internal report provided at [21].

6.1.2 Sensor-specific adjustment

Madgwick's algorithm requires one adjustable parameter, the scalar β , which should represent the magnitude of the gyroscope measurement error, $\tilde{\omega}_{max}$, expressed as a quaternion derivative. Thus, for an error-compensated set of measurements, β should be chosen as a small value. The guideline given in [12] for choosing β is shown in equation 6.1.

$$\beta = \sqrt{\frac{3}{4}}\tilde{\omega}_{max} \quad (6.1)$$

If the gyroscope measurement error is chosen as $0.5 \text{ } ^\circ/\text{s}$ using the measurements shown in figure 5.2 as a guide, the resulting β is $7.56 \cdot 10^{-3}$.

6.2 IMU Firmware

The firmware installed on the IMUs is, as stated in section 5, based on the free AHRS firmware of [3], which runs at 50 Hz. The main changes made during this project are the removal of the orientation filter of [3], the extension of on-board compensation capabilities as described in section 5.3, the change of sampling frequency from 50 Hz to 100 Hz, the addition of an immediate force-measurement command, and the option to include Madgwick's filter during compilation.

As Madgwick's filter is provided in several formats, one being in Matlab, it was chosen to run the orientation tracking filter offline. As stated above, the filter, in both its versions, is included in the firmware. However, it was found that the running time of the filter which includes magnetometer readings exceeds 10 ms and is approximately 15 ms, meaning that only the filter not using magnetometer readings can be used in the current version of the firmware if a constant sampling rate of 10 ms is needed.

6.3 Data collection

Error-compensated sensor measurements are read through from the IMUs in realtime using the serial UART interface and FT232RL FTDI USB to TTL Serial Adapter modules available, at the time of writing, at US \$2.73 [15]. These modules also act as power sources providing 3.3 V for the IMUs. This data collection method runs well at a baud rate of 1 Mbps.

Offline data collection using SparkFun's OpenLog, available from [13], has also been tested at 1Mbps and has been found to collect data with no noticeable differences from the USB method above, except for the lack of data cables and realtime capabilities. This method requires a battery power source, however.

Data is transmitted in a binary format as 32-bit floating point numbers, but can be configured to be transmitted in text format with two digits after the decimal points. For the PC-side binary data collection, several Python scripts have been developed in this project, and are available at [17]. The script used for the dual-IMU data collection of this project enables continuous output from both sensors. Some users may not want this behaviour, because the tolerances on the IMU microprocessors mean that a different number of measurements are made on different sensors in the same time span. Therefore, a synchronised version has also been developed, utilising the added force-measurement command implemented in the firmware. This version requires the data to be collected in realtime, and relies on the PC to keep up with a 10 ms deadline. This can be realised by lowering the USB buffer input size and USB driver delays and by increasing the Python script's process priority.

6.4 Offline data processing in Matlab

All data processing of this project is done in Matlab, excluding sensor error compensation, which takes place on the IMUs, and the Python scripts used for data collection. The Matlab scripts developed in this project make use of Madgwick's filter and the quaternion library provided at [21].

The first step of the data processing is to import the data from text files generated by the data collection scripts and to create time vectors of equal timespan but of different length, so that the difference in number of measurements due to IMU clock tolerances are accounted for. It is assumed that the slowest sensor samples at a rate of 100 Hz.

At this point, an optional step is to recompensate the magnetometer measurements according to the method described in section 5.4.2. This method is based on an ellipsoid fit, however, and should therefore only be applied if the data represent many different orientations, meaning that the IMU was sufficiently excited and rotated about all axes.

The second step of the data processing is to convert the sensor measurements to orientations, using Madgwick's filter and the two different sampling rates obtained from the time vectors. This is either done using the magnetometer version of the filter, or the non-magnetometer version.

If the non-magnetometer version of Madgwick's filter is used, the sensors do not necessarily have the same reference yaw-angle. Therefore, an offset in the yaw-angle is subtracted from all sensor measurements. For each sensor, this offset is determined as the mean of the estimated yaw-angle during a time span where the arm is fully extended. Note that this offset is arbitrarily chosen to align the sensors for the fully extended arm posture, even though they may not be exactly aligned about the Z_W axis.

6.5 Results

It has been found that noise is present in the estimated orientations when choosing β according to the guideline presented in section 6.1.2. It has been found, through trial and error, that values between 0.01 and 0.2 yield better results with less noise. Through manual tuning, the lowest β values resulting in smooth estimates for the 'bi' and 'kj' IMUs were found as $\beta_{bi} = 0.05$ and $\beta_{kj} = 0.1$ respectively. According to equation 6.1, these values should correspond to maximum gyroscope measurement errors of $3.3^\circ/s$ and $6.6^\circ/s$. These were used in Madgwick's algorithm to produce orientation estimates.

Error-compensated sensor measurements from after data collection and unit conversion are shown in figure 6.1. Note that the first 60 s of measurements are not shown, as the arm was held in a constant position during this time.

It is observed from figure 6.1 that the accelerations of the forearm are larger in magnitude, almost 2 g, than those of the upper arm, and that the compensated gyroscope measurements appear to have virtually no drift.

The corresponding orientations obtained using Madgwick's filter without magnetometer measurements are shown in figure 6.2 in relation to the fixed frame. Note that an offset has been subtracted from each set of orientations, calculated as the mean of the estimated orientations from 55 s to 60 s. Figure 6.3 shows the orientations obtained with magnetometer measurements.

The following is observed from figures 6.2 and 6.3:

- The tracked orientations are approximately the same, except for a constant difference in yaw.
- The non-magnetometer filter stabilises within 10 s, while the magnetometer version takes up to nearly 60 s to stabilise.
- The non-magnetometer filter appears to have a slight yaw drift in the first 70 s, whereas magnetometer version does not.
- Quick jumps of 90 ° and 180 ° magnitude appear around 100 s.

Scatter plots for visualisation of the similarities and differences between the two versions of the filters are shown in figure 6.4. These are scatter plots of the upper arm angles (shown vs. time in figures 6.2 and 6.3). The yaw-angle, which drifts in the non-magnetometer version, is displayed in the largest graph.

It is observed from figure 6.4 that the two filters have excellent agreement on the sensor orientation in roll and pitch angles, not taking into account the offsets introduced by the difference in yaw reference angle. In the yaw-angle, the filters are also very similar, but a time-varying drift of one sensor with respect to the other is observed (bottom left part of figure 6.4). This drift appears in equal magnitudes in both directions, and the difference between the measured angles, after subtracting the yaw-angle reference offset, does not exceed 4.6 °.

6.6 Discussion

One possible reason for the noisy orientation estimates could be that more noise is present on the gyroscope measurements when they are worn by a person. The noise observed might actually be due to the "shaking" effect of wearing the IMUs. This could explain the need for larger values of β to filter out this noise, and the reason why β_{bi} of the 'bi' IMU worn on the forearm is

larger than β_{kj} of the 'kj' IMU worn on the upper arm. In this project, the presence of this type of noise in the orientation estimates is regarded as unwanted. However, one could include this noisy movement in the orientation estimates by simply following the guideline for choosing β , presented in section 6.1.2.

The tracked orientations for the magnetometer and non-magnetometer versions of Madgwick's filter are approximately the same. As the magnetometer measurements are error compensated, and as the filters are similar (except for the inclusion of magnetometer measurements in one filter), this is expected.

The slight drift in yaw in the non-magnetometer version as opposed to no drift in the magnetometer version indicates that the gyroscope error compensation is imperfect, and that some bias exists on the measurements. However, the observed drift is small (below 4.6 °for the upper arm), as observed in figures 6.1 and 6.4, and appears to change direction during movement, so it is not expected to be a major source of errors in human body motion tracking.

The quick jumps of 90 °and 180 °magnitude appearing around 100 s are in the roll and yaw angles, and do not represent large changes in sensor position or orientation, even though at first glance, one might assume they were indicator of filter stability problems. This is due to the nature of a three-angle representation, where the -180 °yaw change is countered somewhat by the 180 °change in roll angle.

Based on the results presented above, both filters are judged capable of performing IMU orientation tracking with the IMUs used in this project.

6.7 Summary

Realtime data collection using USB-to-TTL adapters and USB cables has been performed with a baud rate of 1 Mbps. Offline data collection using the same baud rate is also possible using SparkFun OpenLog sensors and a battery as power source.

Using sensor error compensation and Madgwick's filter, it is possible to estimate the orientation of the IMUs used in this project in the fixed frame. The estimates are calculated in Matlab scripts developed for the purposes of this project, and are similar whether including magnetometer measurements in the estimation or not. The only noticeable difference is a small time-varying drift about the yaw angle. This is expected, as the filters are similar, and as the magnetometer is what provides a fixed frame yaw angle reference.

Both filters are judged appropriate for the orientation tracking needs of this project.

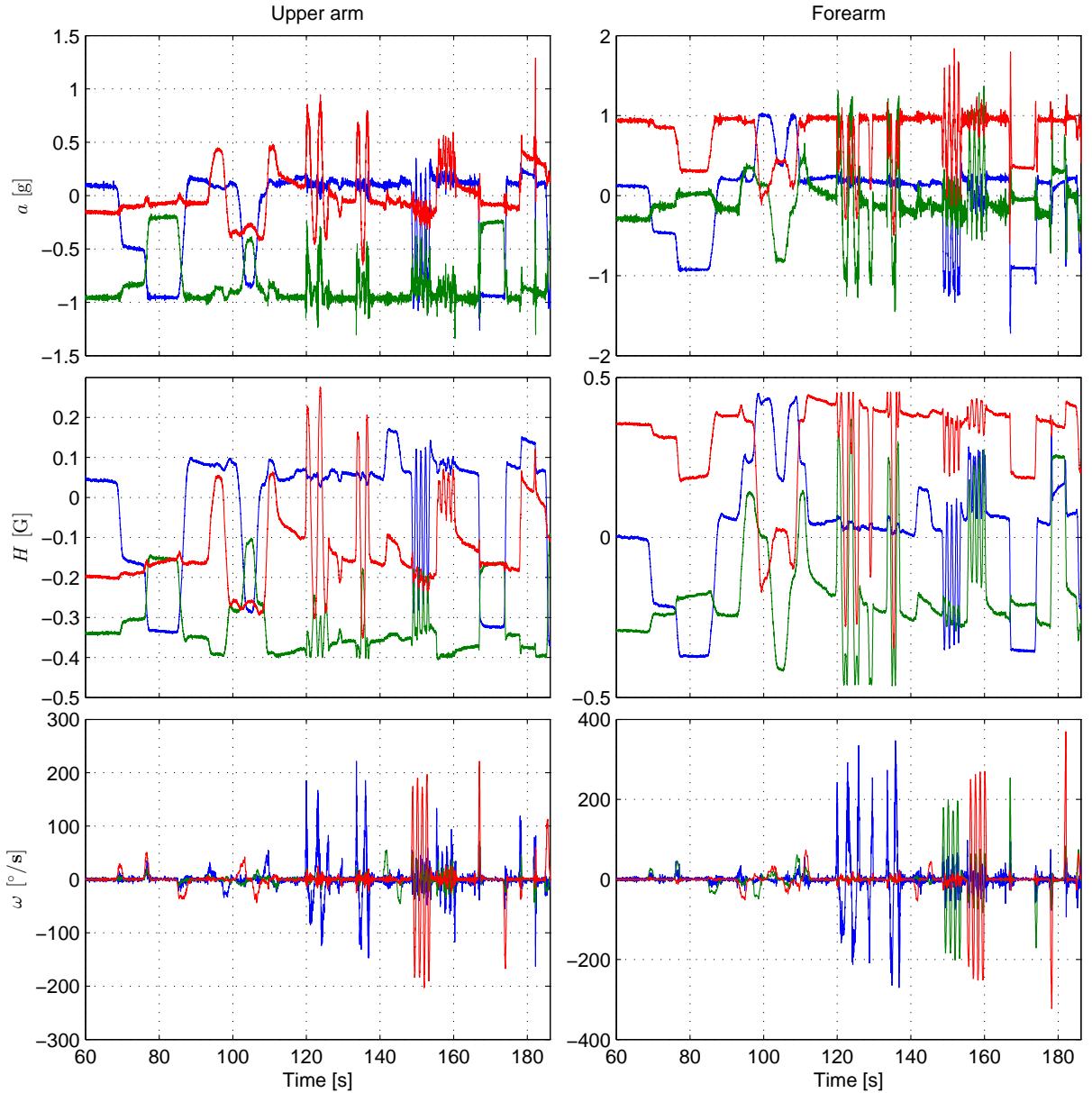


Figure 6.1: Sensor measurements in C_S after unit conversions. Linear acceleration a (top), magnetic field strength H (middle) and angular velocity ω (bottom). The graphs on the left show the measurements of the 'kj' IMU attached to the upper arm, and the measurements on the right show the measurements of the 'bi' IMU attached to the forearm. The first 60 s are not shown because the arm was held still during this time.

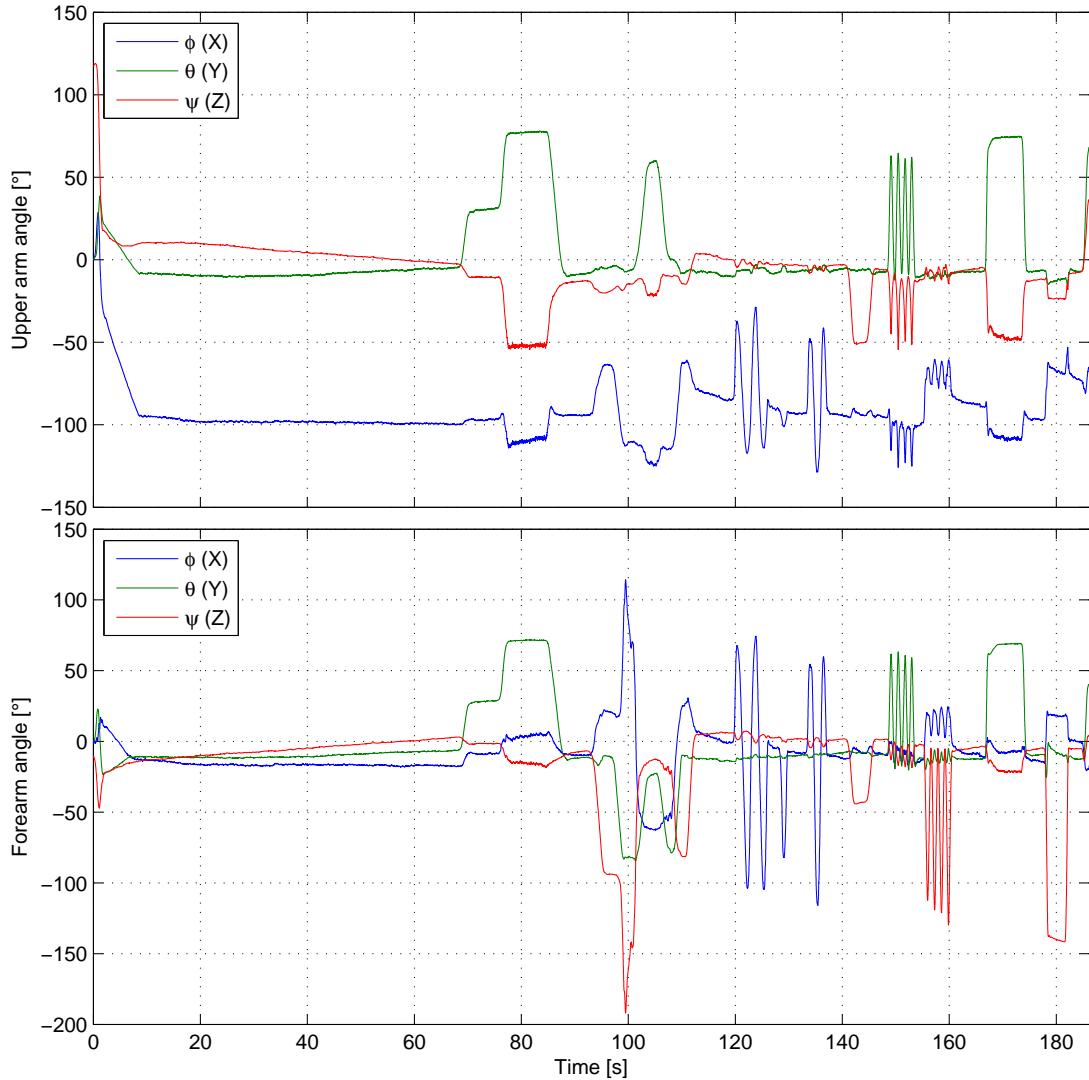


Figure 6.2: Sensor orientation, no magnetometer. Expressed as an intrinsic z-y'-x" rotation in Tait-Bryan angles. ϕ is the rotation about the X_W -axis (roll), θ is the rotation about the Y_W -axis (pitch), and ψ is the rotation about the Z_W -axis (yaw). The sensor orientation is obtained using Madgwick's filter without magnetometer measurements.

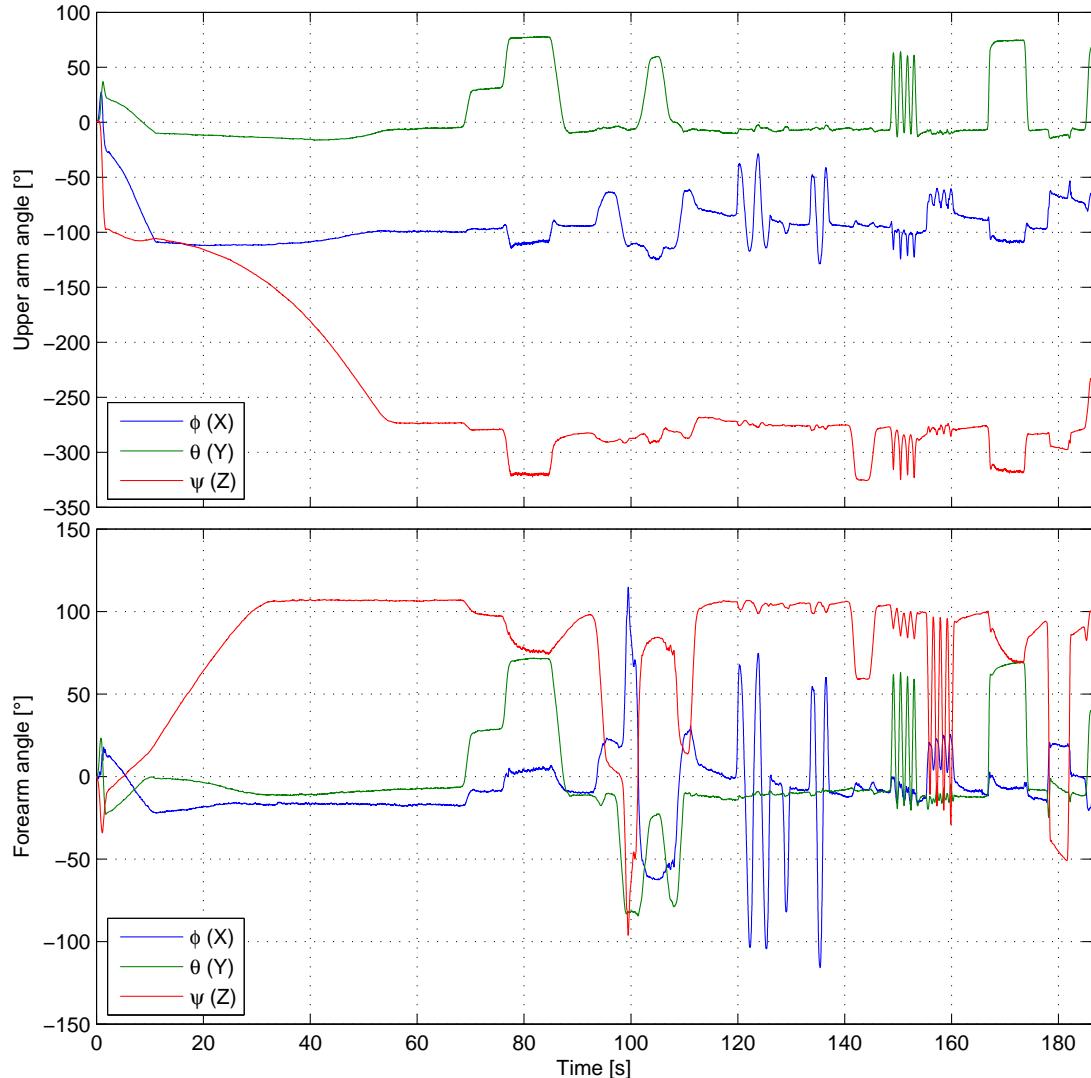


Figure 6.3: Sensor orientation, with magnetometer. Expressed as an intrinsic z-y'-x" rotation in Tait-Bryan angles. ϕ is the rotation about the X_W -axis (roll), θ is the rotation about the Y_W -axis (pitch), and ψ is the rotation about the Z_W -axis (yaw). The sensor orientation is obtained using Madgwick's filter and magnetometer measurements.

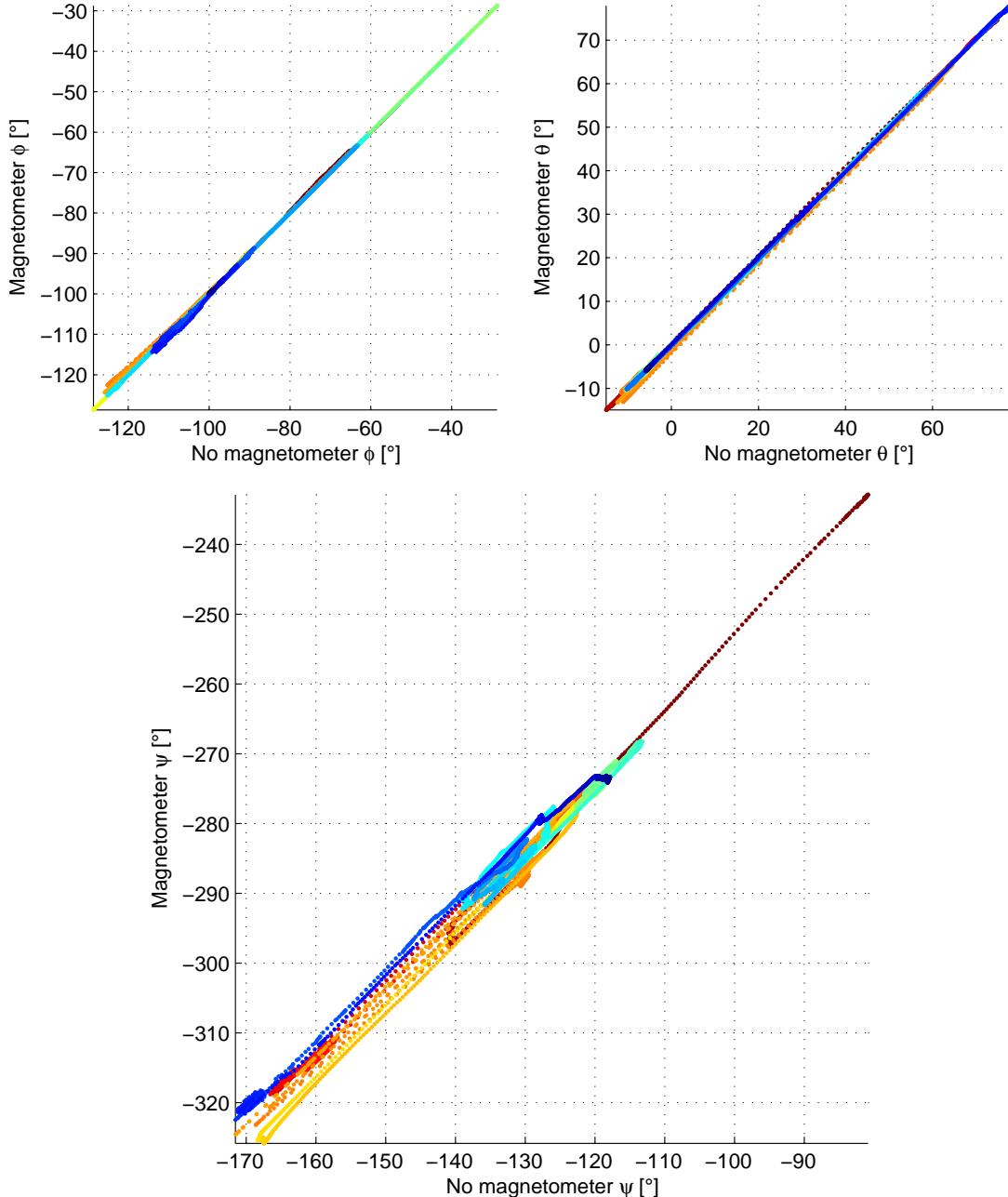


Figure 6.4: Scatter plots of upper arm sensor orientation. The vertical axes are the orientations obtained using Madgwick's filter with magnetometer measurements, and the horizontal axes are the orientations of the non-magnetometer version of Madgwick's filter. ϕ is the rotation about the X_W -axis (roll), θ is the rotation about the Y_W -axis (pitch), and ψ is the rotation about the Z_W -axis (yaw). The colors of the markers indicate the time, meaning that similar colors are measurements from time points close to each other. Similarly, large contrasts between marker colors mean that the measurements correspond to times far apart. The differences between the two filters are most clearly seen in the yaw-angle, hence the enlarged bottom graph.

7 Orientation calibration

This section presents the sensor-to-anatomy orientation calibration carried out in this project. In this paper, what is meant by sensor-to-anatomy orientation calibration is a transformation of the estimated IMU orientations to estimated human body part orientations. The IMU orientations describe rotations in the fixed frame to describe the fixed frame relative to the sensor frame. The purpose of the sensor-to-anatomy orientation calibration is to describe the fixed frame relative to the anatomy frame. If the IMUs are aligned perfectly with the anatomy axes, this calibration does nothing. This is not the case however, and different approaches to sensor-to-anatomy orientation calibration exist.

Three methods of sensor-to-anatomy orientation calibration of the forearm are compared in [9]. The first method is based on a single accelerometer measurement made when the arm is held still with the palm facing down, and a single gyroscope measurement during rotation of the forearm. The second is similar to the first method and is based on averaged gyroscope measurements instead of a single measurement. The third method is more complex, and shown in [9] to be optimal in a statistical sense, as it is based on principal component analysis (PCA). The repeatability of each method was also tested in [9]. The PCA-based method showed reasonable repeatability with an angular difference mean of 1.3917° and a standard deviation of 0.7160° between 30 trials total. The simpler method using averaged measurements, which was shown to be second best, had an angular difference mean below of 1.5436° and a standard deviation of 0.6865° .

The PCA-based method of [9] as well as the simpler averaged measurements method were both implemented, and are part of the Matlab scripts developed for this project. They are equally simple to implement in Matlab, as it has built-in PCA functions. Note that in [9], only calibration of the forearm orientation is carried out. In this project, the exact same calibration procedures are applied to both arm segments, as the anatomy coordinate systems of each segment are defined in the same way, see 4.3, and almost coincide when the arm is fully extended, see figure 4.1.

The sections below describe both calibration methods used in this project. The second method of [9], based on averaged gyroscope measurements, is abbreviated the gravity-based method here.

As shown in figure 4.1, the Y and X axes of the sensor frame and the anatomy frame are switched due to the Z-axes pointing in opposite directions. This is taken into account in the calibration methods, when finding the transformation from one frame to another.

7.1 Gravity-based method

The accelerometers measure the gravitational acceleration when there is no movement. Following the definition of the anatomy coordinate system axes, the acceleration vector, when expressed in the anatomy frame, should only have a Z-component when the palm is facing down. This definition is used in the gravity-based method, and the offset from the vector with a Z-component of 1 is found from a single accelerometer measurement when the palm is facing down.

To find the Y-axis offset, the arm is rotated about the anatomy Y-axes while kept horizontal. If the IMU was perfectly aligned with the anatomy frame, the gyroscope should measure angular velocities only about the Y-axis. The offset is found as the difference between the averaged normalized angular velocities and the vector with a Y-component of 1.

The X-axis offset is found as the cross-product of the Y- and Z-axis offsets.

As a final step, the Z-axis offset is then adjusted slightly by being recalculated as the cross-product of the X- and Y-axis offsets. This is done to ensure an orthonormal basis of the anatomy coordinate system.

Thus, bearing in mind the switch between X- and Y-axes, the rotation from sensor frame to anatomy frame using the gravity method, $\mathbf{R}_{SA,grav}$, is given by equation 7.1, where \vec{X}_{off} and \vec{Y}_{off} are the offset column vectors.

$$\mathbf{R}_{SA,grav} = \begin{bmatrix} \vec{Y}_{off} & \vec{X}_{off} & \vec{Y}_{off} \times \vec{X}_{off} \end{bmatrix} \quad (7.1)$$

7.2 PCA-based method

The PCA-based method does not need accelerometer measurements for the calculation of an orthonormal basis of the anatomy coordinate system.

As in the gravity-based method, the Y-axis offset calculation requires a rotation about the anatomy Y-axis only. The Z-axis offset calculation requires a rotation about the anatomy X-axis only.

The calibration movements required for this method are thus defined as follows: Rotate the arm about the anatomy Y-axes. Then, rotate the arm about the anatomy Z-axes.

The principal component analysis finds the coordinate transformation rotation matrix $\mathbf{R}_{SA,pca}$ which has the largest covariance when projected to a new coordinate system [9]. It does this using all the gyroscope measurements recorded during the one-axis movements described above. The X- and Y-axes of these measurements are switched before being input to the PCA, in order to conform with the definition visualised in figure 4.1.

7.3 Data used for orientation calibration

A subset of the measurements used in section 6 was used for the orientation calibration. Most of the estimated IMU orientations for this subset are shown in figure 7.1. The only measurement of the subset not shown is the one used for finding the Z-axis offset using accelerometer measurements when the palm is facing down. This measurement was picked at time 65 s.

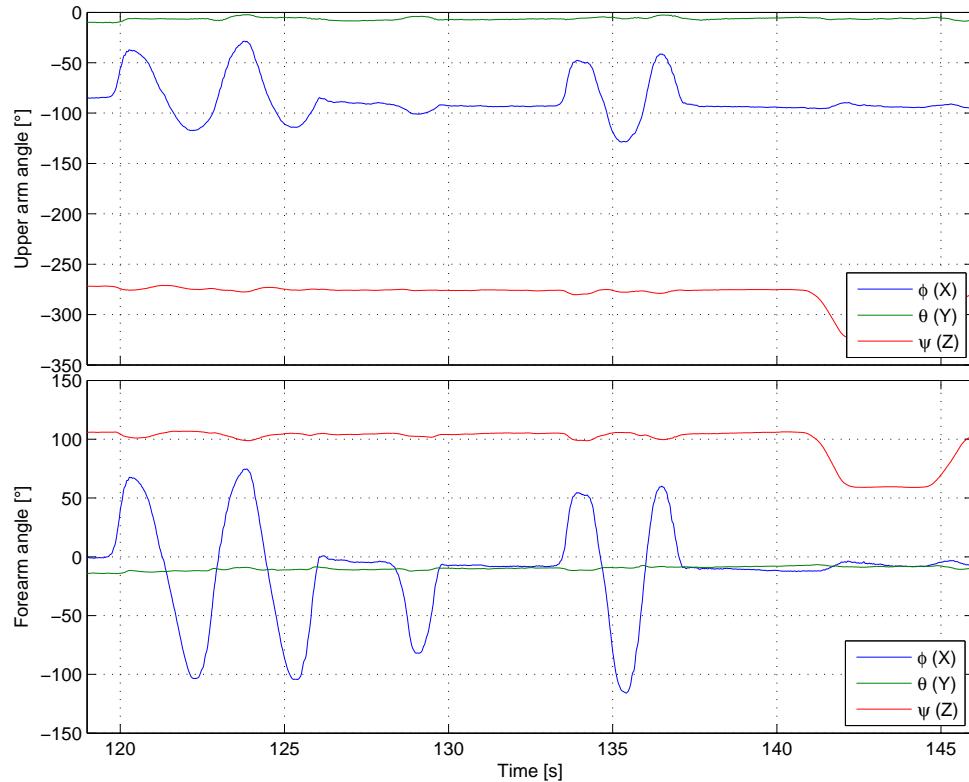


Figure 7.1: Sensor orientation, subset used for orientation calibration, with magnetometer. Expressed as an intrinsic z-y'-x'' rotation in Tait-Bryan angles. ϕ is the rotation about the X_W -axis (roll), θ is the rotation about the Y_W -axis (pitch), and ψ is the rotation about the Z_W -axis (yaw). The sensor orientation is obtained using Madgwick's filter and magnetometer measurements.

7.4 Results

Comparison between no orientation calibration, the gravity-based calibration and the PCA-based calibration is done for each arm segment in the following way:

1. Create a gyroscope measurement N-by-3 matrix Ω_a by switching X and Y axes of the original gyroscope measurements.
2. Rotate the rows of Ω_a by $R_{SA,grav}$ and store the result in N-by-3 matrix Ω_g .
3. Rotate the rows of Ω_a by $R_{SA,pca}$ and store the result in N-by-3 matrix Ω_p .
4. Create covariance matrices C_a , C_g and C_p from Ω_a , Ω_g and Ω_p respectively.
5. Compare nondiagonal elements of the three covariance matrices. Small values indicate a good anatomy frame basis.
6. Compare animations of the entire set of estimated orientations applied to fixed-frame vectors, with no calibration applied, with gravity-based calibration applied and with PCA-based calibration applied to a video recording of the movements.

The reason that small nondiagonal elements indicate a good anatomy frame basis, is that the recorded movements happen in only one anatomy axis at a time. The covariance matrices, subscripted with the letters 'f' for forearm and 'u' for upper arm, are shown in equations 7.2, 7.3, 7.4, 7.5, 7.6 and 7.7. The elements are in $(^{\circ}/s)^2$ and have been rounded to nearest integral values for readability.

$$C_{a,u} = \begin{bmatrix} 16 & -44 & -8 \\ -44 & 7912 & -94 \\ -8 & -94 & 135 \end{bmatrix} \quad (7.2)$$

$$C_{g,u} = \begin{bmatrix} 32 & -372 & -17 \\ -372 & 7881 & 334 \\ -17 & 334 & 149 \end{bmatrix} \quad (7.3)$$

$$C_{p,u} = \begin{bmatrix} 16 & 47 & -12 \\ 47 & 7855 & -667 \\ -12 & -667 & 192 \end{bmatrix} \quad (7.4)$$

$$C_{a,f} = \begin{bmatrix} 140 & 100 & 3 \\ 100 & 1944 & -61 \\ 3 & -61 & 21 \end{bmatrix} \quad (7.5)$$

$$C_{g,f} = \begin{bmatrix} 135 & -7 & -5 \\ -7 & 1858 & 414 \\ -5 & 414 & 112 \end{bmatrix} \quad (7.6)$$

$$C_{p,f} = \begin{bmatrix} 135 & -6 & -8 \\ -6 & 656 & 909 \\ -8 & 909 & 1314 \end{bmatrix} \quad (7.7)$$

It is observed that the nondiagonal elements of the uncalibrated measurements are, in fact, the smallest. The comparison animations and video recording are not shown here in any way. However, with respect to visualisation of movement, all three sets of orientations performed well. One set produced smaller movements during the rotation about the anatomy Y-axis: The PCA-calibrated set.

7.5 Discussion

The covariance results indicate that the sensor-to-anatomy orientation calibration methods applied are worse alternatives to simply not performing calibration. However, when visualised as animations, the best method seems to be the PCA-calibrated set, as it produces smaller movements during rotations about anatomical axes.

This indicates that the proposed method of comparing covariances of rotated is flawed, and that the PCA-based orientation calibration method is best for visualisation purposes.

7.6 Summary

A PCA-based method, [9], is found to produce the best anatomy orientation estimates when visualised in animations. A proposed method of comparing orientation calibration methods using covariances of rotated gyroscope measurements failed to produce neither satisfactory nor meaningful results.

8 Kinematic model calibration through position estimation

A kinematic model includes parameters which describe the length of the links. For visualisation of the anatomical segment orientation, which is an important part of human body motion tracking, the length of the links must be known. For calculations of motion dynamics, the positions of the IMUs on the anatomical segments must be known. In this project, the length of the upper arm length can be approximated using data from the IMUs, because the elbow joint is located between the IMUs. This length is used for 3D visualisation of the arm movement. The forearm length can not be determined unless the forearm IMU is guaranteed to be worn on the wrist. This guarantee is not assumed. However, the distance to the forearm IMU can be determined. Both link lengths can be approximated using other methods, such as measuring with a ruler. Considering that this needs to be done once per wearer, it is considered a minor complication.

An attempt has been made on estimating the upper arm length by estimating the position of both IMUs during circular movement. The idea is that the upper arm length can be determined from the radii of the circular movements which are estimated using linear acceleration data from the IMUs. This attempt is what this section presents.

When double integrating linear acceleration, a , linear position, s , is obtained. The problem with doing this naïvely is that measurement errors get double integrated too, causing a position drift which grows quadratically [6]. A solution proposed by Sebastian Madgwick in [22] is to high pass filter the velocity estimates and the position estimates when the motion is oscillatory.

Thus, in theory it should be possible to determine the length of the upper arm using recorded accelerations during oscillatory movements positioned on a circle.

These types of movements were performed, and are part of the recordings used in section 6.

8.1 IMU position estimation

The method for position estimation during oscillatory movement, as proposed in [22], begins with tilt-compensating the accelerometer measurements. This means the accelerometer measurements are rotated using the filtered orientation estimates. This is done to make the accelerometer vectors uniform, so that during all periods of no movement, a 1 g component is measured in the Z-axis.

The second step is to subtract a 1 g component, corresponding to the gravitational acceleration, from the accelerometer measurements so that only the linear accelerations in the fixed frame are left.

These linear accelerations are integrated numerically, to yield linear velocities. The linear velocities will drift because of the accelerometer measurement errors, and this drift is removed through high-pass filtering using a 1st order Butterworth filter.

The filtered velocities are integrated numerically to yield linear position, which will drift because of errors in the estimated velocity. This drift is removed by high-pass filtering the position using a 1st order Butterworth filter.

The cutoff frequencies for each IMU should be based on the sampling rate of the IMU and the bandwidth of the recorded motion. To remove as much low-frequency drift as possible, the cutoff frequencies are chosen as high as possible without removing too much information from the signals.

8.2 Upper arm length estimation

The first calibration motion sequence is a flapping of the arm. The flapping should be within the accelerometer sensitivity range, and as close to it as possible. The flapping should span a circle in a world plane in which the arm lies, and be repeated several times.

The second calibration motion sequence consists of fast, repeated flexions/extensions of the elbow joint while keeping the elbow still.

The position estimates of subintervals of the first motion sequence are used to create circle fits in 3D, and the mean of the radii of these circles should indicate the distance from the shoulder joint to the forearm IMU and the upper arm IMU. Only the distance to the forearm IMU is necessary to determine the upper arm length.

The position estimate of the forearm IMU during the elbow flexion/extension is fit to circles in a similar manner, and the mean of the radii of the circles should indicate the distance from the elbow joint to the forearm IMU.

The estimated distance from the elbow joint to the forearm IMU is subtracted from the estimated distance from the shoulder joint to the forearm IMU, giving an approximation of the upper arm length.

8.2.1 Results

The normalised velocity high-pass filter cutoff frequencies were found to produce accurate results when equal to 0.24 rad/s and 0.07 rad/s for the upper arm and forearm IMUs respectively. Likewise, the normalised position high-pass filter cutoff frequencies found to produce accurate results are 0.22 rad/s and 0.06 rad/s for the upper arm and forearm IMUs respectively. These values required a high amount of manual tuning dependant on the period of the flapping motion recorded.

The linear velocity estimates are shown in figure 8.1, while the high-pass filtered velocity estimates are shown in figure 8.2.

The linear position estimates are shown in figure 8.3, while the high-pass filtered position estimates are shown in figure 8.4.

An example of one subinterval used for a circular fit is shown in figure 8.5.

It is seen from the figures that the high-pass filtering improves the estimates by removing low-frequency drift and that the position estimates seem to lie on circular paths.

The length of the upper arm is estimated from this data to 33.80 cm. The length measured using a ruler is approximately 31 cm, which means the difference is 2.8 cm.

9 Discussion

After fine-tuning the high-pass filter cutoff frequencies manually according to the period of the flapping motion, the estimated upper arm length is 33.80 cm using the position estimation method for oscillatory motion. This value differs by 2.8 cm from a reference measurement made using a ruler, meaning a 9 % deviation. Note that it might very well be the case that the 33.80

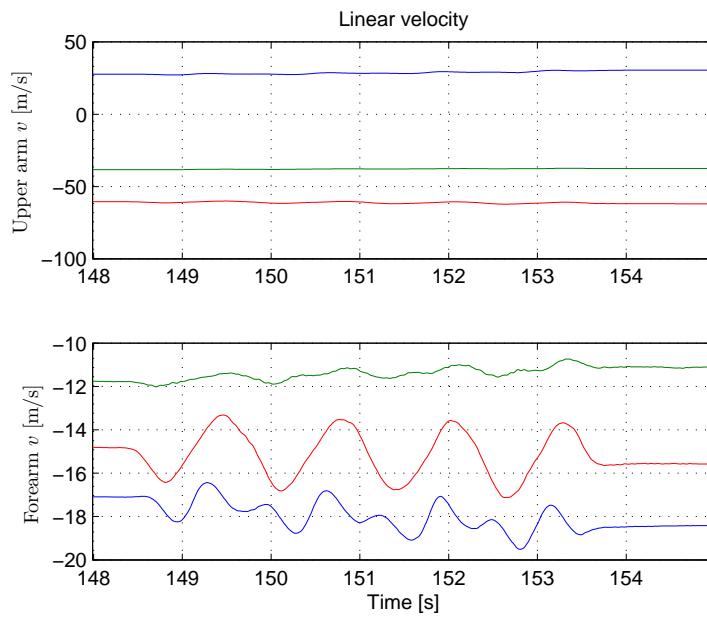


Figure 8.1: Linear velocity, obtained by integrating linear acceleration estimates. X, Y and Z axis measurements are shown in blue, green and red respectively.

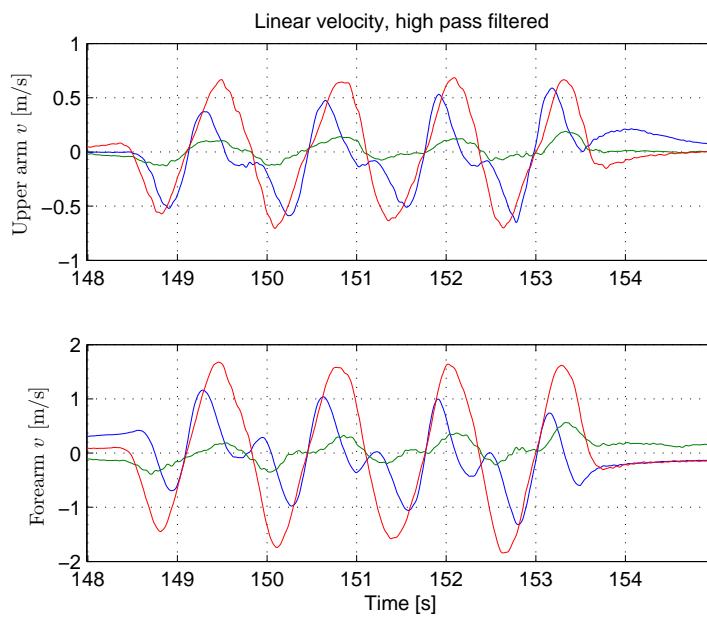


Figure 8.2: Linear velocity, high-pass filtered. X, Y and Z axis measurements are shown in blue, green and red respectively.

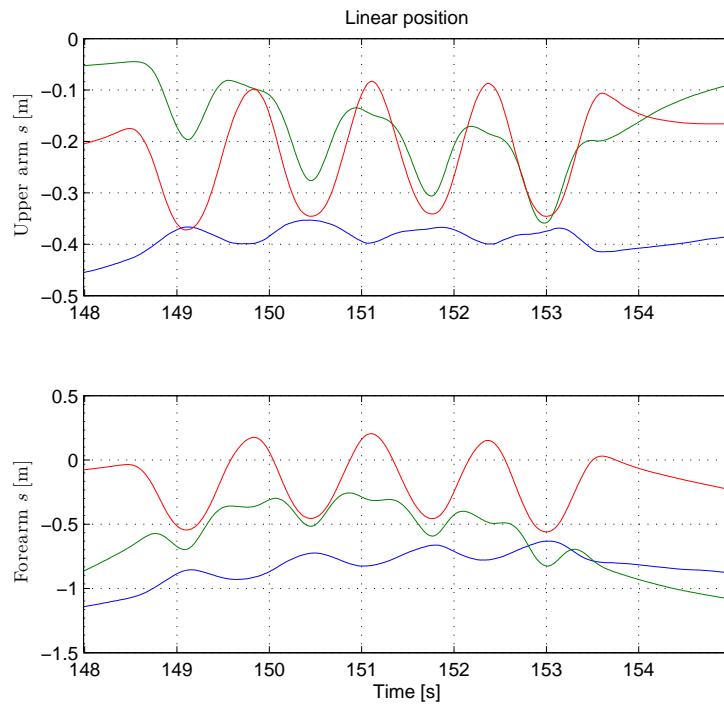


Figure 8.3: Linear position, obtained by integrating high-pass filtered velocity estimates. X, Y and Z axis measurements are shown in blue, green and red respectively.

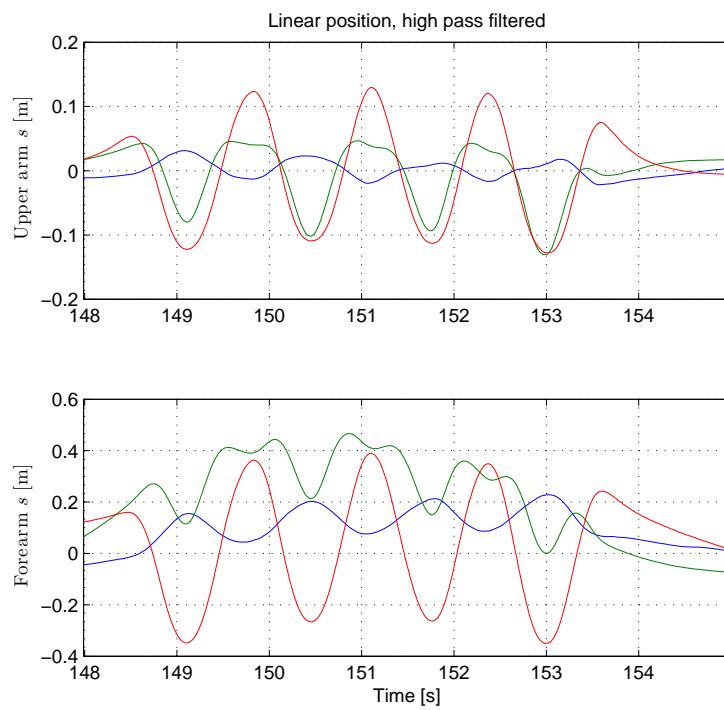


Figure 8.4: Linear position, high-pass filtered. X, Y and Z axis measurements are shown in blue, green and red respectively.

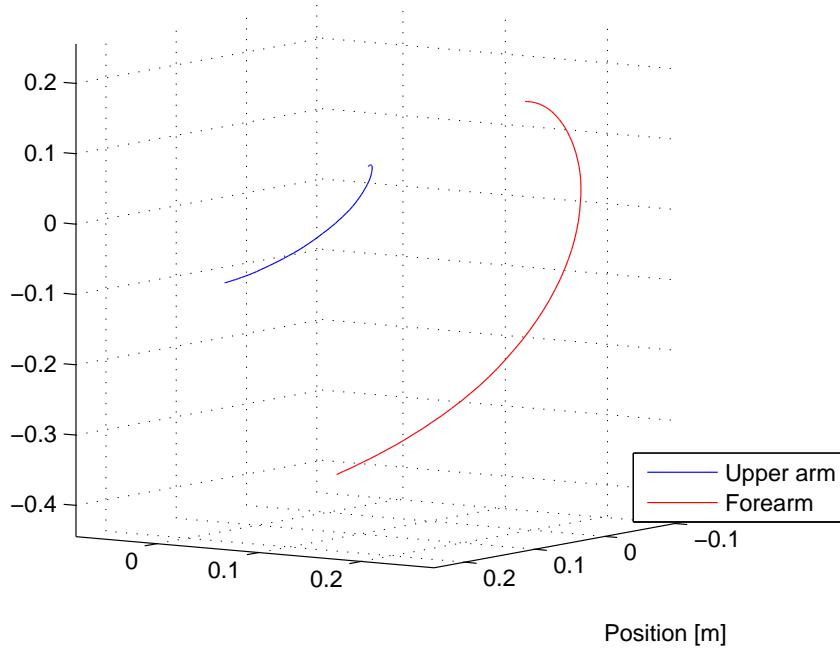


Figure 8.5: Position estimates used for circular fit. All axes are in meters.

cm estimate is closer to the "true" link length estimate, and that the results of this section do not indicate a perfect reference upper arm length with respect to the kinematic model. This is considered satisfactory for some applications, but the amount of required manual tuning to the individual trials renders the method's repeatability very low, as it is subjective to human error.

10 Summary

Position estimation of oscillatory motion has been implemented, but the current implementation requires a considerable amount of manual adjustment. This means the method is not very repeatable, and a method based on measurements of the anatomy using rulers or similar is probably more reliable. However, when fine tuned, the results of this method correspond to those found using ruler measurements, with a deviation of 9 %.

11 Human body tracking results

This section presents qualitative results of the human body tracking, after performing the calibration described in sections 5, 7 and 8.

The orientation is tracked using Madgwick's filter using magnetometer measurements.

An animation showing the tracking of the upper arm and forearm anatomy for 186 seconds is available at [17], along with a corresponding video recording of the tracked movements.

Figure 11.1a shows one example of the tracked anatomy with the corresponding video recording frame in figure 11.1b. In this example, the arm is held at an angle of approximately 45 °.

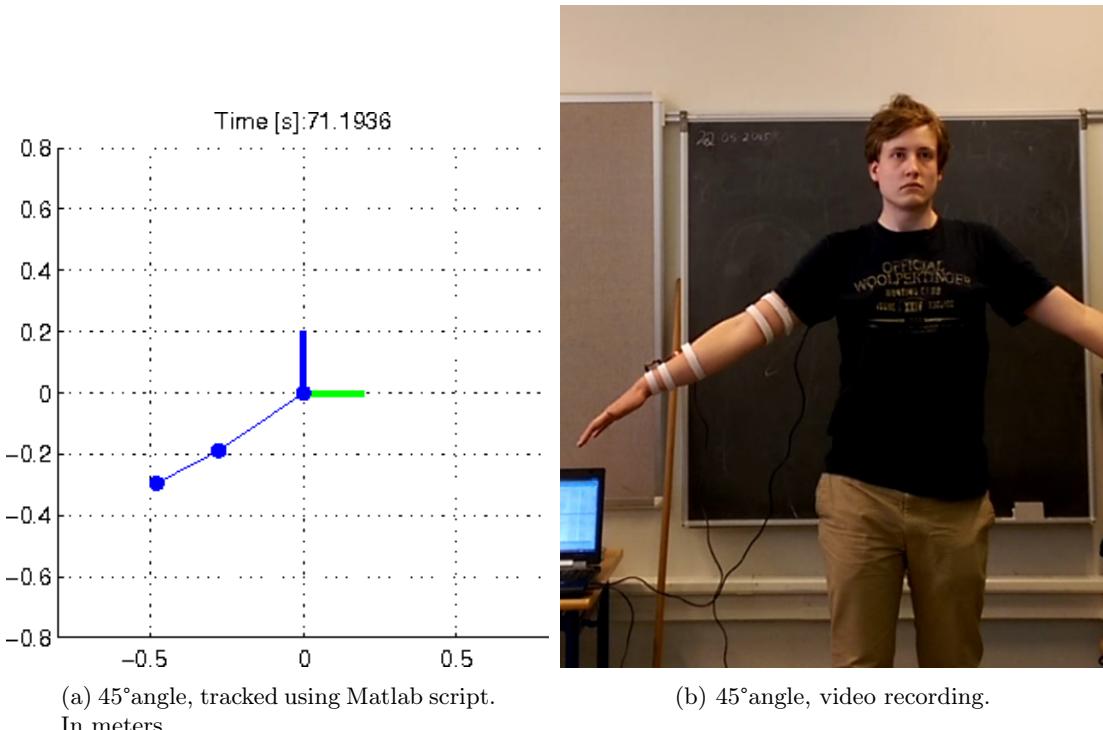


Figure 11.1: 45°angle example. The animation of the tracked anatomical segments (left) and the corresponding video frame (right).

Figure 11.2a shows another example of the tracked anatomy, with the corresponding video frame in figure 11.2b. Here, the arm is held in a downwards orientation.

It is seen from both examples that the anatomy orientation tracking error is small, and that the tracking provides satisfactory results for visualisation.

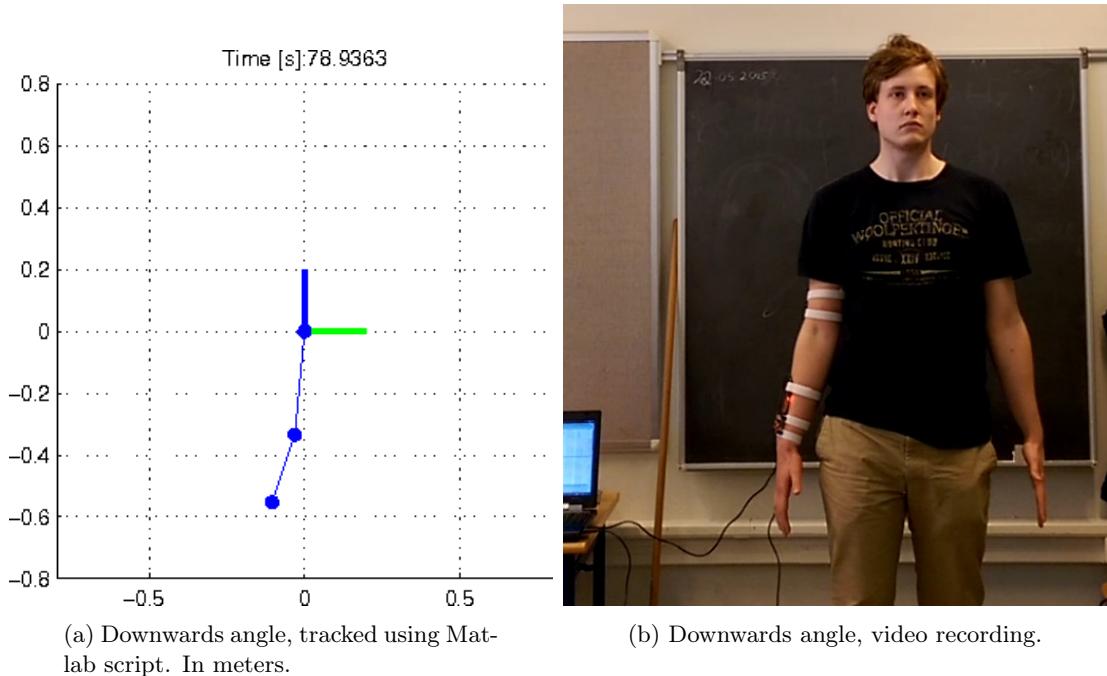


Figure 11.2: Downwards angle example. The animation of the tracked anatomical segments (left) and the corresponding video frame (right).

12 Discussion

The intrinsic sensor measurement error compensation technique presented in section 5 has been shown to reduce drift in gyroscope measurements to negligible values. This makes the magnetometer measurements unnecessary for orientation tracking, as shown in 6, where Madgwick’s filter is implemented in a non-magnetometer version as well as a version using magnetometer measurements. For these reasons, the orientation tracking can be considered robust.

The methods for sensor calibration presented in section 5 allow the sensor’s accelerometer and gyroscope to be fully error-compensated after one calibration. The magnetometer must be calibrated with respect to the environment in which it is placed. If the magnetometer has recorded sufficient measurements along all its axes, this calibration can be performed offline, thus removing the need for any user involvement with the sensor calibration.

Madgwick’s filter has been shown to produce accurate estimates of IMU orientation with and without magnetometer measurements. The estimates are very similar, and this is attributed partially to the intrinsic magnetometer measurement error compensation carried out on the IMUs. The magnetometer version of Madgwick’s filter takes as long as 60 s to stabilize, while the magnetometer free version stabilizes within 10 s. For applications where the small gyroscope drift in the non-magnetometer versions is negligible, this allows for stable and accurate orientation estimates after 10 s.

The user needs to perform certain calibration procedures listed in section 7 in order to allow for sensor-to-anatomy orientation calibration. The sensor-to-anatomy orientation calibration method which showed the best results in terms of 3D visualisation in this project, is based on PCA and has been proved repeatable and statistically optimal in [9]. This method requires only very simple calibration movements: Rotation about only the anatomy Y-axis, and rotation about only the anatomy Z-axis. This is considered to be a very small user requirement, as the arm does not need to be aligned to the fixed frame in a specific way.

A calibration of a kinematic model has been performed: The upper arm length is determined through double integration and high-pass filtering of linear acceleration during oscillatory motion. This method requires the following calibration movements: A fast flapping motion of the entire arm, and a fast elbow flexion/extension motion. These are considered to be moderate user requirements, as not all people are able to move their arm fast. The method has low repeatability and requires substantial amounts of manual adjustments to the high-pass filters used.

The tracking of human arm orientation using the above listed calibration techniques has been tested qualitatively and found to be satisfactory for the purposes of this project which include 3D visualisation.

The IMU firmware is written in Arduino and Processing, is customisable, based on already existing firmware, and is freely available and extensible. The IMU firmware is supported by Python scripts developed in this project. These Python scripts enable realtime asynchronous and synchronous data collection, as well as one-click calibration of the IMUs. The calibration parameters are stored on the IMUs to reduce the amount of user setup, and to allow firmware upgrades which do not reset the calibration parameters. Therefore, the IMU firmware is considered highly extensible.

Matlab scripts have been developed for processing of IMU data. The scripts take as input the error-compensated measurements of the IMUs and produce visualisations of the calibration steps taken along the way, and of the tracked human arm orientations. These scripts are commented, making them easy to read. They are provided freely online, and are extensible by others.

13 Conclusion

Reliable and robust human upper arm and forearm tracking has been implemented at a sampling rate of 100 Hz, using two Sparkfun 9 Degrees of Freedom Razor IMUs [14], equipped with accelerometers, gyroscopes and magnetometers. Satisfactory results were achieved using all three sensors, as well as using only accelerometers and gyroscopes. The data collection is carried out in real-time while the orientation estimation is carried out offline.

An intrinsic sensor measurement error compensation technique has been implemented, and shown to reduce drift in gyroscope measurements to negligible values. This makes the magnetometer measurements unnecessary for orientation tracking. The corresponding implemented methods for sensor calibration allow the sensor's accelerometer and gyroscope to be fully error-compensated after one calibration. The magnetometer must be calibrated with respect to the environment in which it is placed. If the magnetometer has recorded sufficient measurements along all its axes, this calibration can be performed offline, removing the need for any user involvement with the sensor calibration.

Madgwick's filter has been shown to produce accurate estimates of IMU orientation with and without magnetometer measurements. These estimates are used in a statistically optimal and repeatable sensor-to-anatomy orientation calibration method, which aligns the sensor frame with the anatomy frame, to the extent required by this project.

It has been shown that the upper arm length can be determined through double integration and high-pass filtering of linear acceleration measured by an IMU on the forearm during oscillatory motion. This method proved to be too complicated to be considered repeatable.

The IMU firmware implemented in this project is written in Arduino and Processing, is customisable, freely available online and highly extensible. The IMU firmware is supported by Python scripts developed in this project, which enable realtime asynchronous and synchronous data collection, as well as one-click calibration of the IMUs.

Extensible, easy to read and understandable Matlab scripts have been developed and are available online.

Future work within the sensor compensation includes extending the sensor model of the gyroscope to include axis-specific scale factor errors and extending the magnetometer model to include temperature dependencies on all 12 calibration parameters.

Future work should also focus on implementing the orientation estimation with magnetometers and sensor-to-anatomy calibration in realtime, so that the IMUs can be configured to output anatomy orientation directly.

Further testing of the anatomy orientation tracking is necessary to quantify the errors of each step of the calibration performed in this project.

14 References

- [1] Analog Devices. Adxl345 datasheet, Revision 0, 2009. URL <http://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf>.
- [2] Bachman, E. R. An investigation of the effects of magnetic variations on inertial/magnetic orientation sensors. In *Robotics and Automation, 2004. ICRA 2004. Proceedings of the 2004 IEEE International Conference on*, pages 1115–1122 Vol. 2, April 2004.
- [3] Bartz, P. AHRS Firmware for the SparkFun 9DOF Razor IMU and SparkFun 9DOF Sensor Stick, 2013. URL <https://github.com/ptrbrtz/razor-9dof-ahrs>.
- [4] Bartz, P. Tutorial: Building an AHRS using the SparkFun "9DOF Razor IMU" or "9DOF Sensor Stick", 2014. URL <https://github.com/ptrbrtz/razor-9dof-ahrs/wiki/Tutorial>.
- [5] El-Gohary, M. and McNames, J. Shoulder and Elbow Joint Angle Tracking With Inertial Sensors. *Biomedical Engineering, IEEE Transactions on*, 59(9):2635–2641, 2012.
- [6] Foxlin, E. et al. Motion tracking requirements and technologies. *Handbook of virtual environment technology*, pages 163–210, 2002.
- [7] Honeywell. 3-Axis Digital Compass IC, HMC5883L datasheet, Revision d, 2011. URL <http://cdn.sparkfun.com/datasheets/Sensors/Magneto/HMC5883L-FDS.pdf>.
- [8] InvenSense. ITG-3200 Product Specification Revision 1.4, 2010. URL <https://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf>.
- [9] Kim, J.-N., Ryu, M.-H., Choi, H.-R., and Yang, Y.-S. Anatomy Calibration of Inertial Measurement Unit Using a Principle Component Analysis. *International Journal of Bio-Science and Bio-Technology*, 5(6):181–190, 2013.
- [10] Paradiso, J. A. *A Wearable, Wireless Sensor System for Sports Medicine*. PhD thesis, Massachusetts Institute of Technology, 2008.
- [11] Petrov, Y. Ellipsoid fit, 2009. URL <http://www.mathworks.com/matlabcentral/fileexchange/24693-ellipsoid-fit>.
- [12] Sebastian O.H. Madgwick, R. V., Andrew H.L. Harrison. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *Rehabilitation Robotics, 2011. ICORR 2011. IEEE International Conference on*, Zurich, Switzerland, July 2011.
- [13] SparkFun Electronics. SparkFun OpenLog, . URL <https://www.sparkfun.com/products/9530>.
- [14] SparkFun Electronics. 9 Degrees of Freedom - Razor IMU, SEN-10736, . URL <https://www.sparkfun.com/products/10736>.
- [15] tinxi clothes. FT232RL FTDI 3.3V 5.5V USB to TTL Serials Adapter Module Mini Port for Arduino, 2015. URL <http://www.ebay.com/itm/FT232RL-FTDI-3-3V-5-5V-USB-to-TTL-Serials-Adapter-Module-Mini-Port-for-Arduino-/301606607491/>. [Online; accessed 30-May-2015].
- [16] VectorNav. Magnetometer. URL <http://www.vectornav.com/support/library/magnetometer>.
- [17] Westermann, M. Source code and additional files for this project, 2015. URL <https://github.com/madsherlock/Low-cost-wearable-body-tracking/>.

- [18] Wikipedia. Aircraft principal axes — Wikipedia, The Free Encyclopedia, 2015. URL http://en.wikipedia.org/w/index.php?title=Aircraft_principal_axes&oldid=662916600. [Online; accessed 27-May-2015].
- [19] Wikipedia. Gimbal lock — wikipedia, the free encyclopedia, 2015. URL http://en.wikipedia.org/w/index.php?title=Gimbal_lock&oldid=652054431. [Online; accessed 27-May-2015].
- [20] Wikipedia. Quaternion — wikipedia, the free encyclopedia, 2015. URL <http://en.wikipedia.org/w/index.php?title=Quaternion&oldid=662856353>. [Online; accessed 27-May-2015].
- [21] x-io Technologies. Open source IMU and AHRS algorithms, July 2010. URL <http://www.x-io.co.uk/open-source-imu-and-ahrs-algorithms/>. [Online; accessed 30-May-2015].
- [22] x-io Technologies. Oscillatory Motion Tracking With x-IMU, November 2013. URL <http://www.x-io.co.uk/oscillatory-motion-tracking-with-x-imu/>. [Online; accessed 30-May-2015].
- [23] Yun, X., Aparicio, C., Bachmann, E. R., and McGhee, R. B. Implementation and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, Barcelona, Spain, April 2005.
- [24] Zhou, H. and Hu, H. Human motion tracking for rehabilitation—A survey. *Biomedical Signal Processing and Control*, 3:1–18, January 2008.