A rotation experiment on the Digital Motion Processor of the Midge

Bent Engbers*
Supervisors: Stephanie Tan, Hayley Hung†
EEMCS, Delft University of Technology, The Netherlands

January 23, 2022

Abstract

This thesis has researched how the Midge compares to a modern mobile phone regarding the accuracy and reliability of the rotation vector from the DMP in both devices. The rotation of the main axis of the Midge accurately matches that of the modern mobile phone, which means that the accuracy and reliability of the rotation vector from the DMP of both devices are mostly similar. The rotation of the secondary axes of the Midges do not exactly match that of the mobile phone. Additionally, the rotation of the secondary axes differs for each Midge. These differences might suggest that the DMP of the Midges are less accurate at detecting small changes. Further research is required to draw a definite conclusion.

*B.J.Engbers@student.tudelft.nl
†{S.Tan-1, H.Hung}@tudelft.nl

Contents

List of Figures

List of Tables

1 Introduction

Smart wearables are increasingly becoming an integral part of our society today. Smart wearables can be defined as compact computers that the human body can wear. Furthermore, these wearable computers consist of sensors that can determine the orientation of itself and of its wearer [1]. As part of the innovation and development of smart wearables, their sensors are continuously becoming smaller and smarter. This enables the development of new applications, such as post-operative activity in healthcare [2], and fall detection in elderly care [3]. To add, the rise of the Internet of Things has made the data of these sensors available anywhere and anytime [4], [5]. Hence, smart wearables are a helpful tool for research of human behaviour. For example, in the field of healthcare, patient behaviour could be monitored by measuring motion across three dimensions (i.e., the movement in all directions) [6]. This motion is measured by sensors called the Internal Measurement Unit (IMU) and the Digital Motion Processor (DMP). Wearable sensors can, therefore, also help in the COVID-19 pandemic, by aiding in remote patient monitoring and tracking human activity[7].

Observing human behaviour in a natural social context — meaning an environment in which the studied person does not feel observed — is more difficult than doing so in a controlled environment. However, the conclusions drawn on behaviour in a natural social context are often more accurate than when drawn in a controlled environment. The reason for this is that human behaviour changes when feeling observed [8].

Mobile devices with sensors in a natural social context could take observatory difficulties away from researchers. Consequently, the quality of the study of human behaviour could improve. An example of such a mobile device with sensors is the Midge, developed by the Socially Perceptive Computing Lab at the Delft University of Technology for researching human behaviour.[1] This device was developed to register the movements of the wearer to allow the researchers to draw conclusions on the behaviour of this person based on these movements.

Currently, modern electronic devices that need to register movement like the Midge often contain an Internal Measurement Unit (IMU). An IMU is a sensor that can measure acceleration, rotation and magnetic fields. With the data derived from the IMU, the Digital Motion Processor (DMP) can, in turn, calculate the rotation vector. The rotation vector - also referred to as orientation - can be calculated through multiple algorithms, such as the Extended Kalman Filter, the Multiplicative Extended Kalman Filter and the Complementary Filter [9].

This thesis aims to systematically characterise the accuracy and reliability of the rotation vector from the DMP in the Midges and will analyse the following research question:

How does the Midge compare to a modern mobile phone regarding the accuracy and reliability of the rotation vector from the DMP in both devices?

The hypothesis to this research question is that the rotation vector of the DMP in the Midge will be as accurate and reliable as that of a mobile phone. To research this hypothesis, I will first discuss some background information concerning the interpretation of DMP data. Section 3 will then explain the methodology, including the identification of necessary tools to measure, gauge and capture this vector from the DMP. Section 3 will also elaborate on the complications with regard to the data gathering of the Midge, which was necessary for

[1] Midge repository: `https://github.com/TUDelft-SPC-Lab/midge-code`.

the execution of the experiment. This will follow in the experimental setup and an analysis of the results of the experiments, after which the recommendation for future work and the conclusions will be discussed in Section 6. Finally, the ethical aspects of this research project will be discussed in Section 7.

2 Background

Rotations can be represented as Quaternions and Euler angles, both of which have certain advantages and disadvantages. It is important to understand the differences between these concepts, as both are needed to interpret the data of the DMP.

2.1 Euler angles

Rotations are commonly represented as Euler angles. The 3 components of Euler angles correspond to three dimensions. This makes them easier to visualise than Quaternions, as the latter describes a 3D rotation in 4 dimensions.

The Yaw angle corresponds to the z-axis, the Pitch angle corresponds to the y-axis and the Roll rotation corresponds with the x-axis.

These angles correspond to ϕ, θ and ψ respectively as shown in Table 1 and Figure 1.

Name	Greek letter	Symbol	axis of rotation
Yaw	Phi	ϕ	z
Pitch	Theta	θ	y
Roll	Psi	ψ	x

Table 1: The relation between the name, symbol and axis of rotation

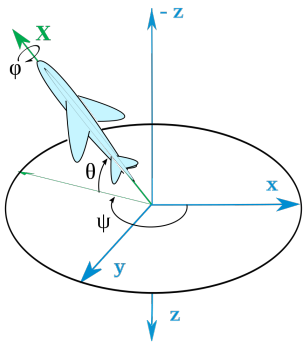


Figure 1: The effect of Euler angles (θ, ψ and ϕ) on orientation [10].

These angles are rotated after one another. In the example of Figure 1, a possible order of rotation is $z \rightarrow y \rightarrow x$. This means that after the first rotation around the z-axis, the orientation of the y and x-axis changes. After the second rotation around the y-axis, the orientation of the last axis x, is changed again. Figure 2 shows how these rotations influence the other axes.

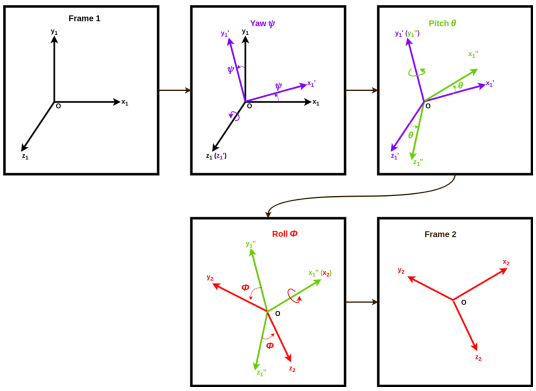


Figure 2: The effects of a sequence of rotations ($z \rightarrow y \rightarrow x$) on the other axes [11].

2.2 Gimbal lock

For most of the rotations, the order of the rotations does not influence the outcome. In other words, a rotation using the order $z \rightarrow y \rightarrow x$ usually produces the same rotation as $y \rightarrow x \rightarrow z$ when given the same values for x, y and z.

If, however, the rotation is a 90° rotation on the y-axis, as shown in Figure 3, a rotation for the order $z \rightarrow y \rightarrow x$ will not produce the same outcome as another order of axes. The x and z-axis line up in such a way that not all directions can be reached by rotating the x and z-axis, losing a degree of freedom. This phenomenon is called a gimbal lock. All orders of axes have their own gimbal lock. Which particular rotation is needed for a given order is determined by the axis in the middle of the order . This means that for the order $y \rightarrow x \rightarrow z$ a 90° rotation of the x-axis is required, for the order $z \rightarrow y \rightarrow x$ a 90° rotation of the y-axis and for the order $x \rightarrow z \rightarrow y$ a 90° rotation of the z-axis. Figure 4 shows the all the corresponding gimbal locks for their corresponding order.

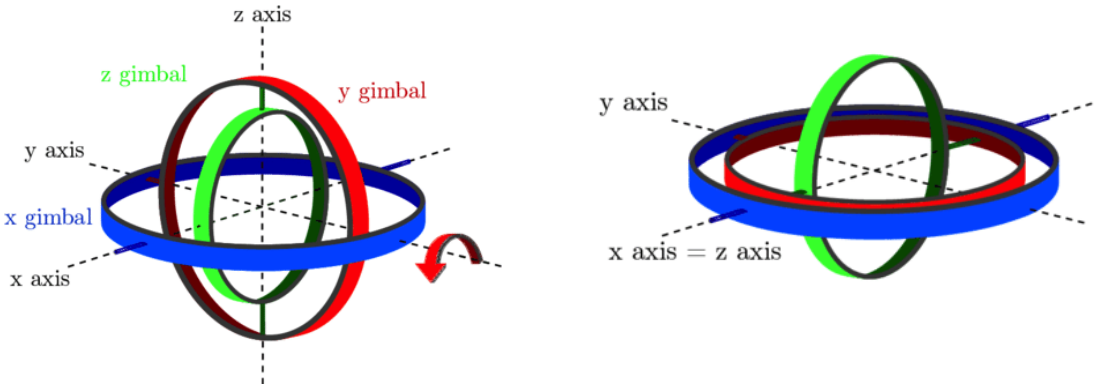
2.3 Quaternions

A solution to the gimbal lock is the use of quaternions. This form of rotation provides a way to describe a three-dimensional rotation using four variables.

2.3.1 Definition

Quaternions are defined by 4 (w, x, y, z) real numbers, and 3 basic quaternions [14].

$$w + x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k} \tag{1}$$



(a) The initial position of all gimbals.

(b) Position after a rotation of the 90° rotation of the y-axis, causing the x and z-axis line up, losing a degree of freedom.

Figure 3: A 90° rotation of the y-axis causing gimbal lock [12].

This can be devided into a vector part $x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$ and a scalar part w. Where for the basic quaternions $\boldsymbol{i}$,$\boldsymbol{j}$ and $\boldsymbol{k}$ the following holds [14]:

$$\boldsymbol{i}^2 = -1 \tag{2}$$
$$\boldsymbol{j}^2 = -1 \tag{3}$$
$$\boldsymbol{k}^2 = -1 \tag{4}$$
$$\boldsymbol{ijk} = -1 \tag{5}$$
$$\boldsymbol{k} = \boldsymbol{ij} = -\boldsymbol{ji} \tag{6}$$
$$\boldsymbol{i} = \boldsymbol{jk} = -\boldsymbol{kj} \tag{7}$$
$$\boldsymbol{j} = \boldsymbol{ki} = -\boldsymbol{ik} \tag{8}$$
$$\tag{9}$$

Quaternions solve the gimbal lock problem, by adding an extra axis (w) to a three-dimensional rotation. If a gimbal lock occurs in the vector part, there is always a fourth axis that is not in gimbal lock.

2.3.2 Conversion to Euler angles

The conversion from Quaternions to Euler angles is according to the following formulas [15].

$$\phi = \arctan\left(\frac{2(wx + yz)}{1 - 2(x^2 + y^2)}\right) \tag{10}$$
$$\theta = \arcsin\left(2(wy - zx)\right) \tag{11}$$
$$\psi = \arctan\left(\frac{2(wz + xy)}{1 - 2(y^2 + z^2)}\right) \tag{12}$$

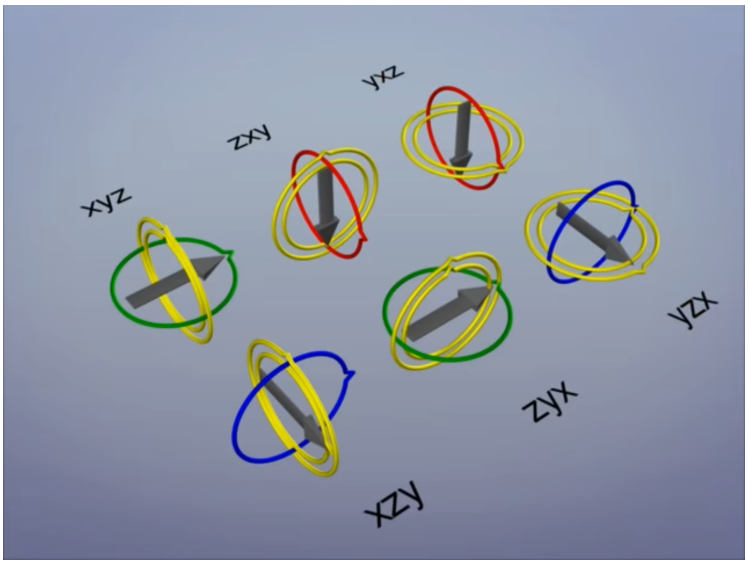


Figure 4: All different sequences of rotation, and their corresponding gimbal locks [13].

In the python library used for this conversion, $\arctan\left(\frac{a}{b}\right)$ is replaced by $\text{atan2}\,(a, b)$ and $\arcsin\left(\frac{a}{b}\right)$ is replaced by $\text{asin}\,(a, b)$[16].[2]

$$\phi = \text{atan2}\left(2(wx + yz), 1 - 2(x^2 + y^2)\right) \tag{13}$$

$$\theta = \text{asin}\,(2(wy - zx)) \tag{14}$$

$$\psi = \text{atan2}\left(2(wz + xy), 1 - 2(y^2 + z^2)\right) \tag{15}$$

3 Methodology

This thesis aims to test the rotation vector of the DMP, by comparing the signals of the rotation vector from the DMP in the Midge to signals of that from a modern mobile phone[3]. Using a smartphone as comparison material enables me to identify possible flaws in the study methodology. The results of these experiments depend on how precise and repeatable the sensor can rotate. A programmable rotating platform can repeat the same rotation multiple times. The degree and the speed in which the platform rotates need to be controlled and carefully monitored.

This section will first identify the tools necessary to do the experiment, after which some complications concerning data gathering will be discussed.

[2]The exact implementation in the squaternion library can be found here: github.com/MomsFriendlyRobotCompany/squaternion.

[3]Modern mobile phones also contain a DMP. For this experiment an iPhone 12 was used.

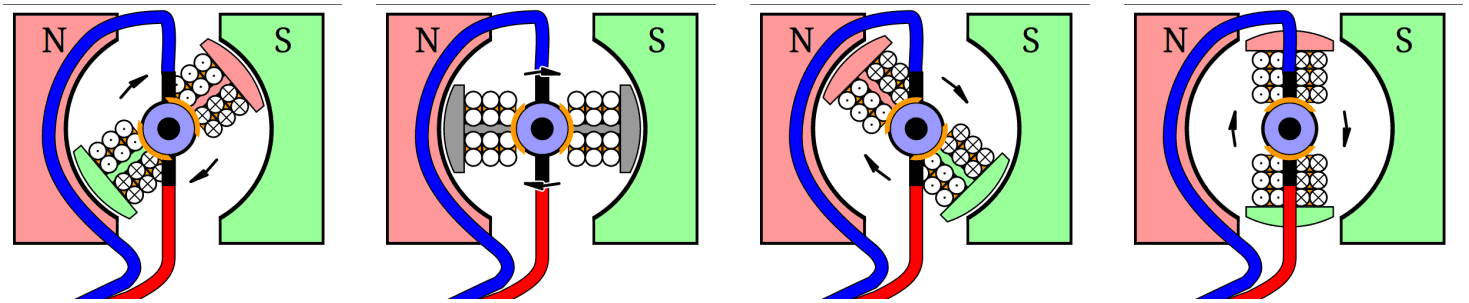
3.1 Tools

3.1.1 Motor

The preciseness of the rotation of a programmable rotation platform depends on the motor and driver. A distinction can be made between two sorts of electrical motors: direct current (DC) motors and stepper motors. DC motors are efficient and better for high speed rotation, but they lack the precise positional control [17].

Figure 5 shows the different steps in the rotation of a DC motor. Stepper motors, on the other hand, rotate an exact number of steps. Combined with the correct driver, these motors can even rotate in smaller substeps. Figure 6 shows the different steps in the rotation of a stepper motor. The speed of the rotation is dependent on the electrical power and the load on the motor. In order to produce a stable rotation, the use of a DC motor would require feedback of the accurate rotation measurement and a controller that can quickly respond to this feedback.

As a consequence, the complexity of the setup increases and the experiment will be more error-prone. A stepper motor is more suitable for this project because of its preciseness.



(a) Electricity flows through the armature.

(b) The commutators do not touch the wires. The armature is off.

(c) Electricity flows through the armature in the oposite direction of a.

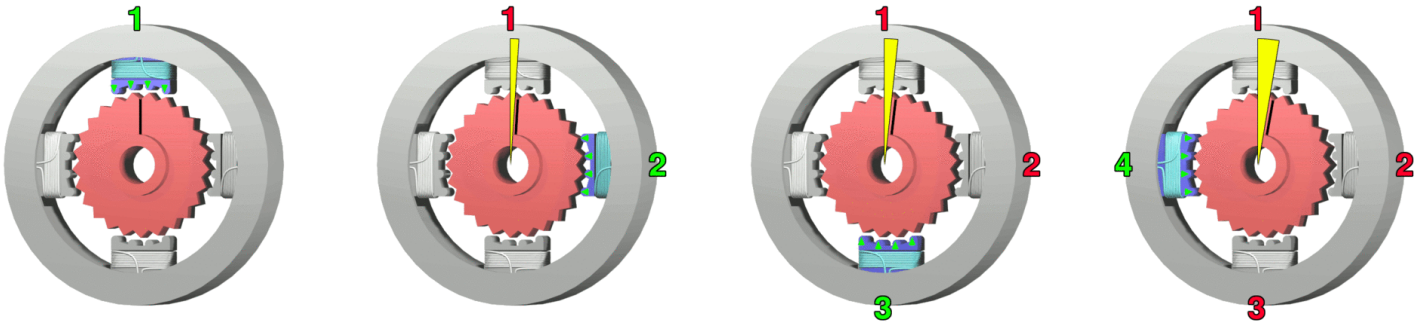
(d) The armature continues its rotation.

Figure 5: The rotation of a direct current (DC) motor [18].
The orange parts are the commutators. The rotating center is called an armature. When the commutators make contact with the wires, electricity starts flowing through coils. This causes a magnetic field that changes with part of the turn.

The rotation of the motor will only be accurate when the physical resistance on the motor is substantially smaller than its torque. The motor will skip steps if the physical resistance is too large. Therefore, this experiment needs a stepper motor which can produce a large amount of torque. The particular stepper motor used for this setup is a Wantai 42BYGHW811 NEMA 17. This motor is popular in 3D printers and CNC machines, which require a large degree of precision. This motor has a step size of 1.8 degrees, and a step accuracy of +/- 5% [19].

3.1.2 Driver

The driver of this stepper motor was an important factor for controlling the precision, because some drivers are capable of half stepping and/or micro stepping. The smaller the step of the driver, the more accurate the stepper motor and the more steps per rotation.



(a) Coil 1 is active. **(b) Coil 2 is active.** **(c) Coil 3 is active.** **(d) Coil 4 is active.**

Figure 6: The rotation of a stepper motor.
The red gear shaped part is called the rotor. The yellow triangle shows the angle rotated by the rotor.

Table 2 shows this relation between the step size, degrees per step and the amount of steps per rotation. Smaller steps also reduce the amount of torque the stepper motor can create and the speed of the rotation of the stepping motor. When steps are too big they may cause non-linearity in the rotation, because the stepwise electrical power input in the motor coils from the controller. Ultimately, this may cause periodical acceleration and deceleration in a twitching motion.

Step size	Degrees per step	Steps per rotation
1	1.8	200
$\frac{1}{2}$	0.9	400
$\frac{1}{4}$	0.45	800
$\frac{1}{8}$	0.225	1600
$\frac{1}{16}$	0.1125	3200
$\frac{1}{32}$	0.05625	6400

Table 2: The relation between the different modes of stepping and the rotation speed.

3.1.3 Wiring

The DRV8825 driver is controlled by five pins as shown in Figure 7. Three pins are used to control the different stepping modes ($\frac{1}{2}$-stepping, $\frac{1}{4}$-stepping, $\frac{1}{8}$-stepping, $\frac{1}{16}$-stepping and $\frac{1}{32}$-stepping) [21]. One pin is used to control the direction of rotation.

The last pin controls the amount of steps that are taken in which time frame. This last pin uses a protocol called Pulse Width Modulation (PWM): the amount of pulses determines the amount of steps the stepper motor will take. In turn, the frequency of pulses determines the speed of the steps the stepper motor will take. The parameters used for the experiment can be found in Section 4.1.3.

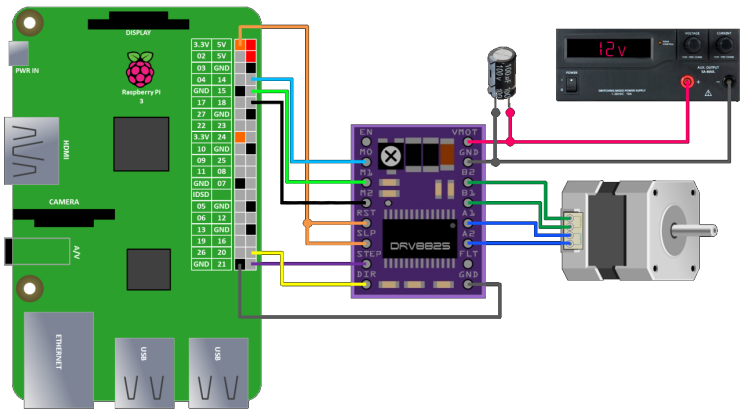


Figure 7: The wiring between the Raspberry Pi, the driver and the stepper motor [20].

3.1.4 Pigpio

As the timing of these pulses needs to match the timing of the direction pin, hardware timers are preferred over software timers. A library which enables such timings is Pigpio, which can create waveforms. Consequently, waveforms use Direct Memory Access (DMA) to create pulses, which are accurate to the microsecond.

The section about the experimental setup will describe the way the tools for the experiment are used. However, before being able to start with the setup of the experiment, there were some implementation complications that had to be solved.

3.2 Data gathering complications

There have been several complications in this research project, which caused an initial failure of the experiment. Gathering the data from the Midges is necessary to answer the research question. In order to manage the Midges for execution of the experiment, it was necessary that the hub was well-functioning and usable. The hub, a program written in python, manages the Midges as follows. The hub is responsible for sending Bluetooth Low Energy commands with the initial timestamp to the Midges. In this way the hub synchronises the timestamps of the Midges. This section discusses these complications and how we solved them.[4]

3.2.1 Connection between the hub and the Midges

Firstly, there were multiple complications concerning the hub and the Midges. In previous experiments the code was executed on a so-called *dedicated laptop* [22]. This is a laptop on which the hub and all its dependencies were installed. The dedicated laptop was not solely available for the experiment of this research project.

Therefore, we installed the hub software on a Raspberry Pi. However, the hub code was still running on python 2.7, which had reached its end of life on 1 January 2020 [23]. To be able to use the hub code, It had to be migrated to python 3.9.

[4]Ties Schwertasek from the research group needed a similar experimental setup.

In addition, there were issues with the dependencies of the hub. The hub initially used a package, dependency and environment manager called Conda. We were provided with an environment file, that describes the environment for the hub consisting of 24 dependencies. This file was put into Conda[24]. Unfortunately, all the dependencies of this environment file were outdated; some were so outdated that they could not be downloaded anymore. The supervisor clarified that there were only three dependencies necessary for the hub to work, All of which where still available.

3.2.2 Automatic stop

At first, when managing the Midges with the hub, the recording of the Midges could only be stopped manually. To resolve this, we migrated a newer version of the hub, which came from a second private repository, from python 2.7 to 3.9. From there on, the hub was able to stop the Midges automatically.

3.2.3 Parser

The Midges save their sensor data in binary format on a micro-SD card, with a separate file for each sensor, containing the measurements. Each of these files consists of binary chunks that start with a Unix timestamp and ends with the measurements of the different sensors. We were provided with a parser, to convert the binary format to a more easily readable file format

After parsing the data with this parser, the following problems emerged:

- The parser saved the parsed data and the plots on the micro-SD cards instead of on the computer. This caused a substantially slower workflow, as we had to copy the parsed data from the micro-SD cards to our computers and erase this data from the micro-SD cards manually and for each experiment separately. This is a problem, because of the substantial amount of different files that are saved on the micro-SD cards. It is not feasible to manually save each parsed file individually in the allocated time for the execution of this experiment. Additionally, as the parser did not erase the parsing data of the previous attempts, the data of different attempts easily got mixed up, making it unclear which files of parsed data were from a previous attempt and which of the current one.

- The parser did not backup the raw data (in binary format) on the computer. Again, this is a problem, because of the substantial amount of different files that are saved on the micro-SD cards. If, for example, the parser contained an error and this error is amended, having a backup of the raw data is essential to be able to go through the parsing process again. As the parser did not automatically backup the raw data, this was not possible.

- The name of the different folders with the files of both the raw and the parsed data of each experiment only consisted of a Unix timestamp. Therefore, it was very difficult to make a distinction between all the different experiments.

- It was unclear how the command-line interface flags, which were incorporated in the parser, should be used.

- The parser did not plot the data from the DMP on a graph. Therefore, it was impossible to interpret the data and, ultimately, to draw any conclusions from the experiments.

All of these issues were resolved by rewriting a substantial part of the parser.

3.2.4 Binary chunks

It is crucial for the experiment that the size of the binary chunks is correct, which means that the chunk size in the parser is equal to the chunk size that comes out of the Midges. If the size of the binary chunks is incorrect, this will result in incorrect parsed data and, ultimately, in unusable results of the experiments.

We parsed the timestamps to check whether the size of the binary chunks was correct. After parsing the timestamps should have a reasonable value. A value of a timestamp is reasonable when it is parsable and when its date is congruent with the date of the experiment. If most of the timestamps do not have a reasonable value, it is very likely that the size of the binary chunks is incorrect, hence the data are corrupted. Some Midges solely produced unreasonable timestamps, while others only produced a few. Hence, the Midges that solely produced unreasonable timestamps, also produced binary chunks that had an incorrect size and, therefore, resulted in incorrect data.

This problem occurred because the Midges that solely produced unreasonable timestamps used a newer version of the firmware than the other Midges. Midges that used the older version of the firmware produced binary chunks of 32 bytes, while the Midges that used the newer version produced binary chunks of 24 bytes. As the parser was set on a chunk size of 32 bytes, the newer version of the firmware was incompatible with the parser. The difference in firmware versions of the Midges was only discoverable after parsing the raw data of the different Midges.

Another cause for the binary chunk size problem was that the Midges were programmed for older and slower micro-SD cards. However, these micro-SD cards are not available anymore. Therefore, we could not completely solve this problem. We worked around this problem by keeping the experiments short.

4 Rotation experiment: setup

This section will describe the setup of the experiments and analyse its results. Unfortunately a lot of time was lost to solve the failed data transmission and handling from the midget Therefore, there was not enough time left to fully complete the experiments.

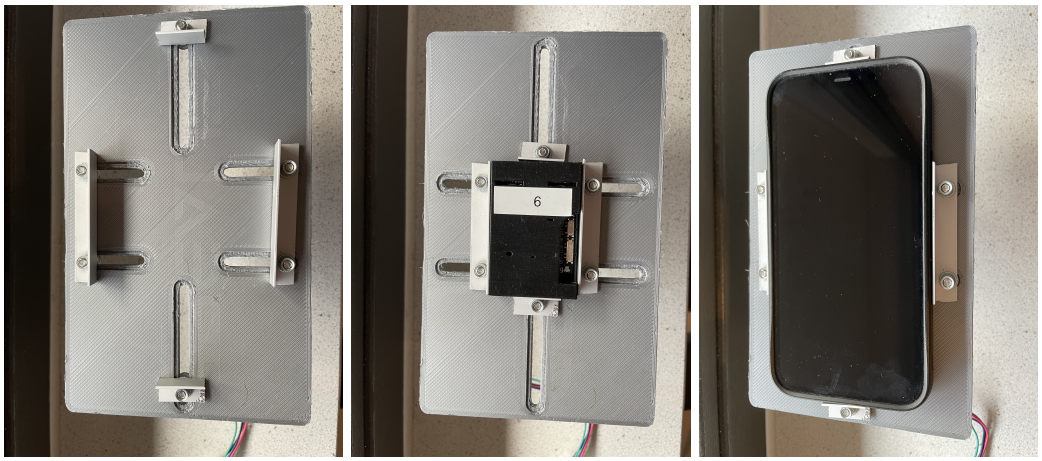
4.1 Setup

The setup of the experiment consists of the following three steps:

1. setting up the wiring as described under Section 3.1.3 in figure 7;
2. connecting the stepper motor to both a Midge and the mobile phone; and
3. writing a program that could create waveforms using the Pigpio daemon.

4.1.1 Platform

The connection between the stepper motor to the Midge and the mobile phone will be realised by using a platform with horizontal clamps and a shaft collar. For this purpose I designed a 3D printable platform, as shown in Figure 8. The platform can clamp both



(a) Empty platform. **(b) Platform with Midge.** **(c) Platform with mobile phone.**

Figure 8: Different configurations of the platform.

the Midge and the mobile phone to the platform using horizontal clamps. These clamps ensure a similar direction of rotation for both devices. Additionally, the devices can easily be removed using the clamps. This is useful for extracting the data of the devices after the execution of the experiment. In turn, the shaft collar connects the platform and the stepper motor (see figure 9).

For this step it is important that both devices are tightly secured to the stepper motor; there must be no slack between the devices and the stepper motor. This could influence the accuracy of the results of the experiment.

The Midges and the mobile phone are a completely different size. Therefore, to prevent slack, the clamps need to be perfectly adjustable to both sizes.

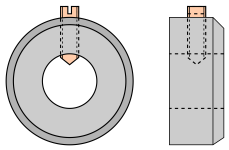


Figure 9: A shaft collar [25].

4.1.2 Rotation Program

The third step consists of writing a program that could create waveforms using the Pigpio daemon. The rotation program that sends commands to Pigpio, needs to have the following properties. The program needs to be able to run multiple separate experiments interchangeably. This is important, because it allows for easy iteration of the experiment, without modifying the original experiment.

The rotation program also needed a safety stop, that would the rotation of the motor.

This is important, because the stepper motor, the driver and its surroundings could be damaged. This required a large change in the program flow. The rotation script would finish before the motor started rotating, because it only constructs the commands for the Pigpio deamon, and it does not wait for the execution of all commands. I resolved this by calculating how long all commands would take, and keeping the script active during this period. If 'Ctrl-C' is pressed during the execution of the program, a command to the Pigpio deamon is sent, clearing all waveforms, thus stopping the motor.

The rotation problem can run experiments interchangeably, by reading a JSON file that contains the instructions for the experiment. The contents of this JSON file are first validated using a predefined schema. The schema consists of a list of instructions, with each instruction consisting of three fields:

- *steps*: The amount of steps to take.
- *freq*: The frequency (in Hertz) of the steps.
- *dir*: The direction of the steps (*right* for clockwise, *left* for counter clockwise).

Several different sorts of experiments can be done in this manner.

4.1.3 Experiment Parameters

For this experiment, I chose a stepsize of $\frac{1}{2}$ with a constant step frequency of 75 Hz turning clockwise. The stepsize was decreased from full, to provide a smoother motion and reducing the risk of non-linearity.

To make it easier to identify a correct result, I set the amount of steps to 1200 for three full rotations (see Table 1). This makes it easier to check if the corresponding graph is correct, because all Euler angles should start and stop at the same value. These parameters resulted in a 1080° rotation in 16 seconds.

4.1.4 Mobile phone recording

The Phyphox application was used to record data from the mobile phone[26]. To record the DMP values from the mobile phone an experiment needed to be loaded from the wiki of Phyphox which can be found here [27].

5 Rotation experiment: results

The results of the experiment are set out in the following figures and tables in the Appendix:

- Figure 10 refers to the raw quaternion components from the three Midges before the correction in the parser and the mobile phone.[5]
- Figure 11 shows the raw quaternion components for the three Midges and the mobile phone after the correction in the parser.
- Figure 12 shows the computed Euler angles using the formulas from Section 2.3.2 (the axes are called X2, Y2, Z2 to differentiate between the X-, Y- and Z-grap of Figure 10).
- Figure 13 shows the corresponding Euler angles, computed with the corrected parser.

[5]Midge 6a and Midge 6b refer to two separate executions of the experiment using Midge 6.

- Figure 14 illustrates the raw quaternion components from the three Midges only.
- Table 3 and 4 contain the minimum and maximum values of each of the components respectively.

The data of the devices in Figures 10and 11 are manually synchronised using the value of the W-graph. As Euler angles are computed from the quaternion values, Figure 12 and 13 are also automatically synchronised.

5.1 Visual analysis of the mobile phone

As follows from the data in Figure 12, the different angles of the rotation of the mobile phone are perfectly in sync. Between the plots of the X- and Y-graph on the one hand and, respectively, of the X2- and Y2-graph on the other, Figures 10 and 2 relatively small differences are visible. Presumably, this is because the platform is not completely level, which causes a slight tilt on the x- and y-axis in every rotation.

The plot of the Z2-graph shows a rotation of constant speed, as the degrees as function of the time increases linearly. The maximum degrees of the Z2-graph is 179.39°, and its minimum value is −179.97°. The difference between the minimum and maximum value is 359.36°, which is almost exactly a full rotation. The pattern repeats itself three times, which matches the three full rotations. The value in degrees of the X, X2, Y, Y2 are also in sync with the three rotations. The value in degrees of the X2-, Y2- and Z2-graphs at the starting point and at the end point of the graph are almost exactly the same, as can be expected with the same start- and end-position.

5.2 Visual analysis of the Midges

The data of the Midges have been parsed twice because of a bug in the parser. For the sake of being complete, both analyses have been added to this thesis. In Table 3 and 4, y and z of all the Midges initially seemed to produce the same values. Additionally, the plot in Figure 14 shows that the graphs of Y and Z of each Midge are identical. The similarity between these values was caused by a bug in the parser through which the data of Y overrode the data of Z.

5.2.1 First analysis

The data from the Midges does not rhyme with the data from the mobile phone. For each Midge the rotations of X, Y and Z of Figure 10 are almost perfectly in sync (except for Midge 3 at the 18 seconds mark). However, when the different Midges are compared to each other, they produce different values of raw data at similar stages of the rotations. Even when the experiment is executed with the same Midge multiple times (Midge 6, a and b), it results in different values of raw data as function of the time.

The different values of raw data between the Midges at the same stages of rotation is also apparent in Figure 12. The value in degrees at a certain time in the graphs of X2, Y2 and Z2 of Figure 2 are not the same for each Midge. From the data of Figure 2, it follows that the values of X2 and Y2 drift out of phase with each other during the experiment.

The start and end time of the movement of the Midges and the mobile phone is the same. The interval between the start and the stop time (approximately 16 seconds) corresponds with the total time of the experiment. However, the values at which each Midge starts and

stops is not the same. At the end time in the graphs X2 and Y2 of Figure 2, the Midges seem to correct back to zero degrees.

5.2.2 Second analysis

The Z and Z2 graphs of Figure 11 and 13 show data from the Midges after correcting the parser, which is more similar to the data of the mobile phone than it was before correcting the parser. The Z-axis is the main axis of rotation and, therefore, the most important element for answering the research question.

The values in raw data and degrees respectively of X, Y, X2 and Y2 still seem to be different for each Midge. The platform has been levelled using a spirit level before the experiment, but small variations in the x- and y-axis are to be expected. The time and magnitude of these variations should occur at constant intervals during the experiment, because the platform stays exactly the same in each setup. The Midges do not show this constant interval, because the position of these variations is different for each Midge. Furthermore, the Midges still show a slight drift during the experiment on the X, Y, X2 and Y2 axis as described in the first analyses. This seems to suggest that the Midges are less accurate in measuring small rotations ($< 10°$). However, as these variations are difficult to control, further research is needed to be certain about this.

6 Conclusions and Future Work

This thesis has researched the following question:

How does the Midge compare to a modern mobile phone regarding the accuracy and reliability of the rotation vector from the DMP of both devices?

The short answer to this question is that the rotation of the main axis of the Midge accurately matches that of the modern mobile phone, which means that the accuracy and reliability of the rotation vector from the DMP of both devices are similar. The vertical values on the X- and Y-graph (the secondary axes) of the Midges do not match those of the mobile phone. Additionally, the secondary axes differ for each Midge. These differences might suggest that the DMP of the Midges are less accurate at detecting small changes ($<10°$). However, further research on this concept is necessary to draw a definite conclusion. Further research is also necessary to conclusively assess the accuracy and reliability of the rotation vector from the DMP in the Midge. For a conclusive research, the experiment has to be repeated while changing the following variables:

- the direction of the rotation;
- the speed of the rotation;
- the orientation of the Midge on the platform;
- the duration of the experiment; and
- a combination of these variables.

I have contributed to future research regarding the above-mentioned elements, as I have debugged the hub and the parser. I have also added some features to the parser and improved the quaternion and Euler plotter, which contributes to the efficiency of future experiments with this setup.

Furthermore, to assess whether the Midge can actually be used to research human behaviour, the speed and interval of the rotation of the platform need to be compared to those of humans.

7 Responsible Research

Responsible research has become an increasingly important aspect of modern research [28]. This section analyses the ethical aspects of this research project and discusses the reproducibility of the methods used. Reproducibility of research is important, because it enables scientific work to be verified. Important factors that assess the responsibility of research are integrity, quality and social impact [29]. When applying these factors to the underlying research project, the following topics should be ethically assessed:

- the safety of the experiment;
- the reproducibility of the rotation experiment; and
- the fundamental implications, such as privacy.

Firstly, the safety of the experiment improved by adding a manual stop option to the rotation program. Although the speed of the rotation in this experiment was quite low, the stepper motor used in this research project is capable of high speeds. A misconfiguration of the speed can cause higher speeds than expected which could cause safety risks for the executer of and the materials for the experiment. The manual stop option enables an immediate stop of the stepper motor in case of an emergency.

Secondly, as this thesis reports the way I resolved all the problems that arose during the implementation of the experiment and describes how the experiment needs to be executed, this experiment can easily be reproduced. However, when reproducing this research, one does need to take into account that the versions of the hub, the operating system and the parser are up-to-date. The programs used for this research project are all openly available[6].

A final important aspect for responsible research are the fundamental implications of the research project. This is especially important in this research project, as these sensors will ultimately analyse human behaviour, and thus will process personal data. When using sensors that monitor human behaviour, it is important that the researcher only uses the collected data for its research purpose and ensures that the data is securely stored. According to the General Data Protection Regulation, which applies to the processing of data in the European Union, sufficient safeguards need to be put into place to guarantee a safe level of protection of the collected personal data [30].

Appendix A Results

[6] https://github.com/TUDelft-SPC-Lab/midge-code

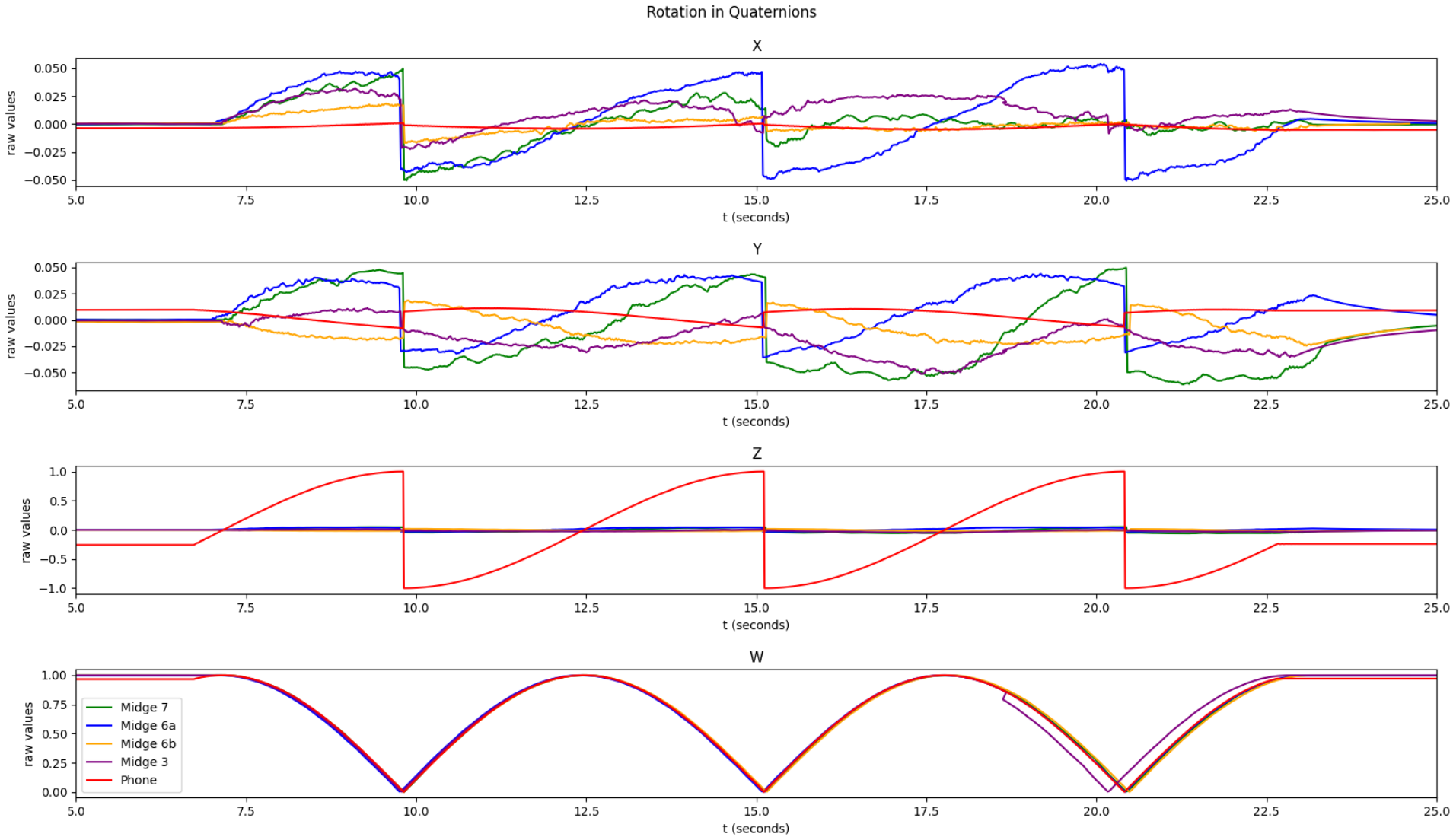


Figure 10: Raw quaternion output of a constant clockwise rotation
1200 steps at 0.9° per step (3 full rotations)
Step speed 75 Hz (16 seconds for 3 full rotations)

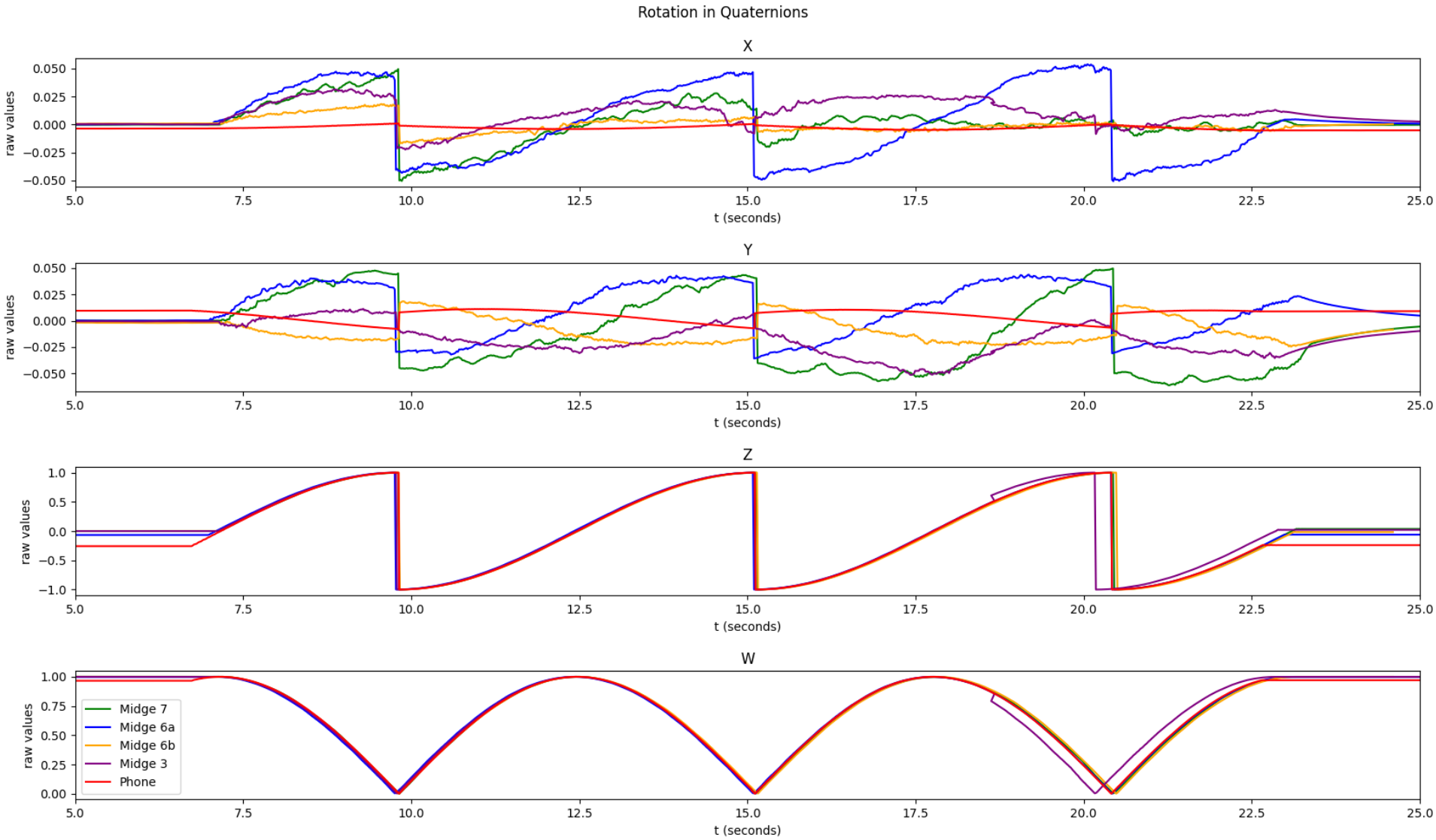


Figure 11: Raw quaternion output of a constant clockwise rotation after parser correction
1200 steps at 0.9° per step (3 full rotations)
Step speed 75 Hz (16 seconds for 3 full rotations)

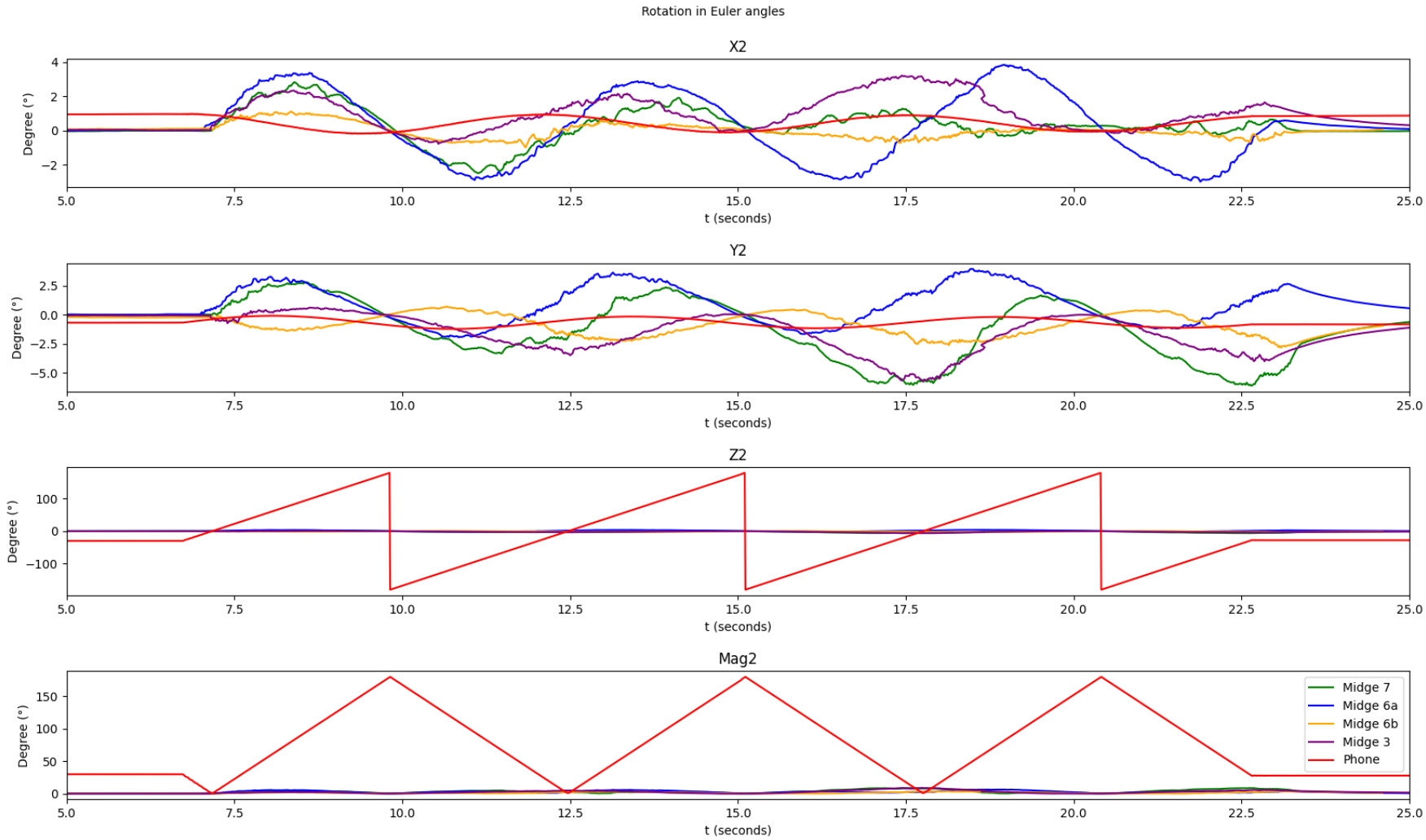


Figure 12: Parsed Euler angles a constant clockwise rotation
1200 steps at 0.9° per step (3 full rotations)
Step speed 75 Hz (16 seconds for 3 full rotations)
The X,Y and Z axis are named X2,Y2,Z2 to differentiate between the X,Y,Z values of Figure 10

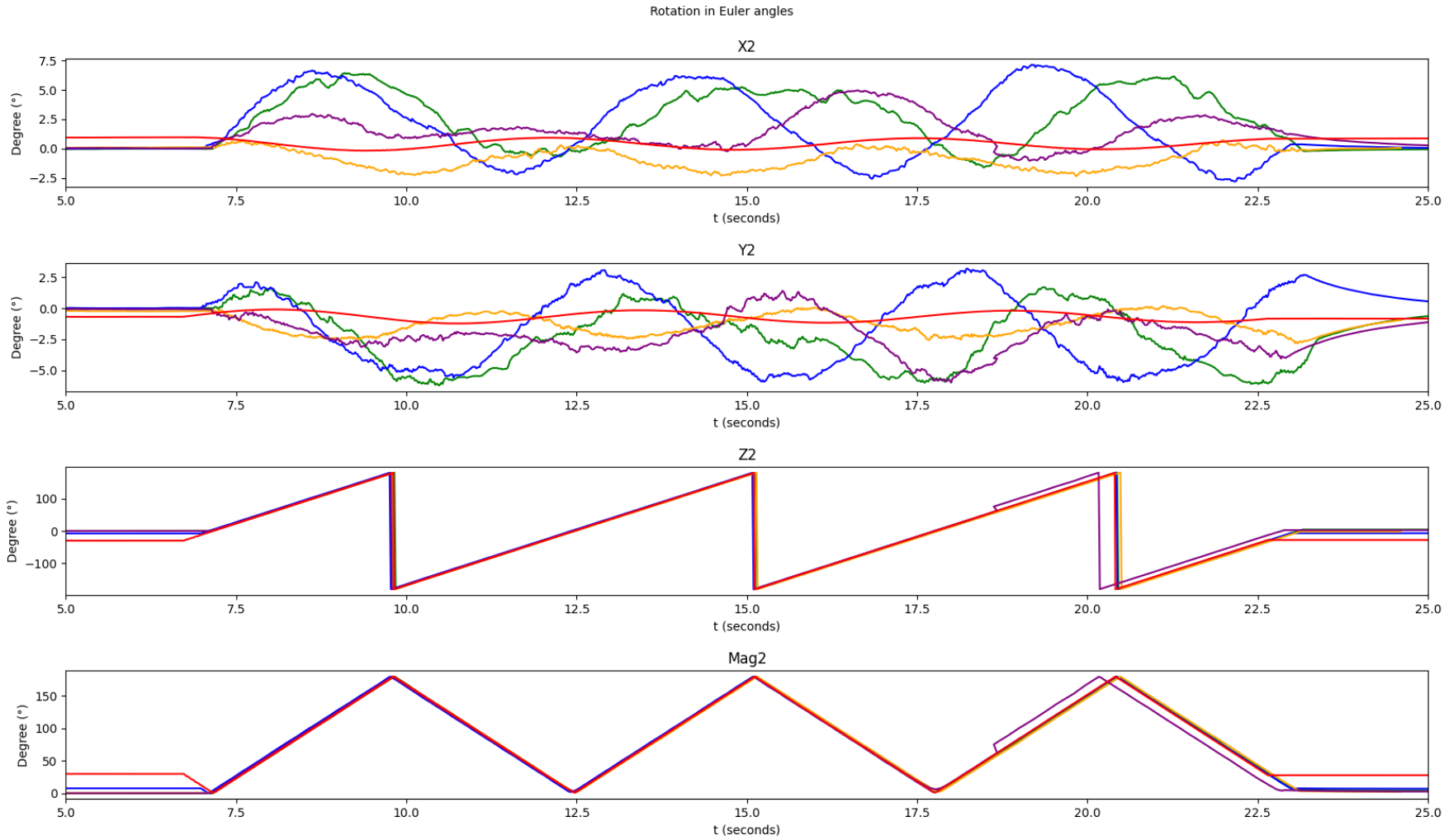


Figure 13: Parsed Euler angles a constant clockwise rotation after parser correction
1200 steps at 0.9° per step (3 full rotations)
Step speed 75 Hz (16 seconds for 3 full rotations)
The X,Y and Z axis are named X2,Y2,Z2 to differentiate between the X,Y,Z values of Figure 10

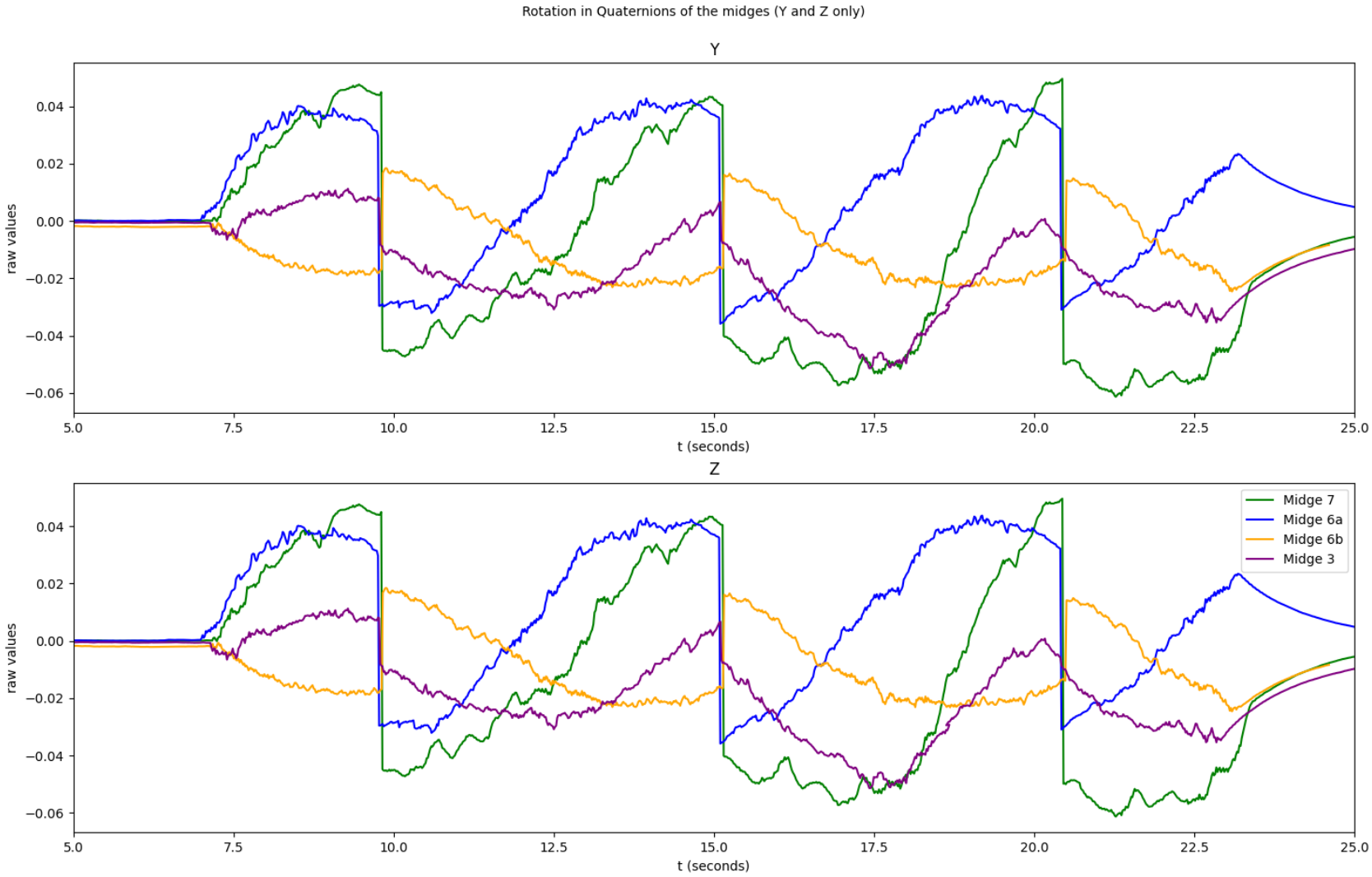


Figure 14: Raw quaternion output of a clockwise rotation, Y and Z of the Midges only
1200 steps at 0.9° per step (3 full rotations)
Step speed 75 Hz (16 seconds for 3 full rotations)

name	unit	Midge 7	Midge 6a	Midge 6b	Midge 3	Mobile phone
w	raw	0.000725	0.001363	0.000133	0.001161	0.000258
x	raw	-0.050413	-0.050629	-0.016993	-0.022413	-0.016515
y	raw	-0.061332	-0.035832	-0.024607	-0.051396	-0.007953
z	raw	-0.061332	-0.035832	-0.024607	-0.051396	-0.999978
x2	Degree °	-2.480995	-2.972083	-0.981015	-0.769497	-0.181795
y2	Degree °	-6.104008	-1.942513	-2.826422	-5.714405	-1.230364
z2	Degree °	-6.164570	-1.711691	-2.812751	-6.027807	-179.970284
mag2	Degree °	0.001803	0.003425	0.022389	0.003418	1.009897

Table 3: The minimum values of each of components of Figure 10 and 12

name	unit	Midge 7	Midge 6a	Midge 6b	Midge 3	Mobile phone
w	raw	1.000000	0.999986	0.999999	1.000000	0.999961
x	raw	0.049573	0.053823	0.018329	0.031620	0.006884
y	raw	0.049644	0.043725	0.018495	0.011268	0.019083
z	raw	0.049644	0.043725	0.018495	0.011268	0.999964
x2	Degree °	2.819912	3.827211	1.113479	3.205538	2.430617
y2	Degree °	2.800991	3.949630	0.710058	0.608040	0.199119
z2	Degree °	3.071085	4.151165	0.675548	0.669809	179.391111
mag2	Degree °	8.691455	6.458397	3.997164	8.865180	179.970476

Table 4: The maximum values of each of components of Figure 10 and 12

References

[1] N. Niknejad, W. B. Ismail, A. Mardani, H. Liao, and I. Ghani, "A comprehensive overview of smart wearables: The state of the art literature, recent advances, and future challenges," *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103 529, 2020, ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2020.103529. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197620300348.

[2] D. Olguin Olguin, P. A. Gloor, and A. Pentland, "Wearable sensors for pervasive healthcare management," in *2009 3rd International Conference on Pervasive Computing Technologies for Healthcare*, 2009, pp. 1–4. DOI: 10.4108/ICST.PERVASIVEHEALTH2009.6033. [Online]. Available: https://dspace.mit.edu/bitstream/handle/1721.1/60066/Olguin-2009-Wearable%20sensors%20for%20pervasive%20healthcare%20management.pdf?sequence=1&isAllowed=y.

[3] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable sensors for reliable fall detection," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005, pp. 3551–3554. DOI: 10.1109/IEMBS.2005.1617246.

[4] K. J. Kim and D.-H. Shin, "An acceptance model for smart watches: Implications for the adoption of future wearable technology," *Internet Research*, 2015.

[5] C. Perera and A. V. Vasilakos, "A knowledge-based resource discovery for internet of things," *Knowledge-Based Systems*, vol. 109, pp. 122–136, 2016.

[6] A. Jalal, M. A. K. Quaid, and A. S. Hasan, "Wearable sensor-based human behavior understanding and recognition in daily life for smart environments," in *2018 International Conference on Frontiers of Information Technology (FIT)*, 2018, pp. 105–110. DOI: 10.1109/FIT.2018.00026.

[7] D. R. Seshadri, E. V. Davies, E. R. Harlow, *et al.*, "Wearable sensors for covid-19: A call to action to harness our digital infrastructure for remote patient monitoring and virtual assessments," *Frontiers in Digital Health*, vol. 2, p. 8, 2020. [Online]. Available: https://doi.org/10.3389/fdgth.2020.00008.

[8] J.-M. Hoc, "Towards ecological validity of research in cognitive ergonomics," *Theoretical Issues in Ergonomics Science*, vol. 2, no. 3, pp. 278–288, 2001. DOI: 10.1080/14639220110104970. eprint: https://doi.org/10.1080/14639220110104970. [Online]. Available: https://doi.org/10.1080/14639220110104970.

[9] R. Bruens, M. Liefaard, T. Huisman, H. Kroon, and K. van Heel, "A literature study on imu orientation estimation using sensor fusion," sensor fusion.

[10] Juansempere. "Yaw, pitch and roll." (2009), [Online]. Available: https://commons.wikimedia.org/wiki/File:Plane.svg.

[11] A. Deshpande. "Rotation using euler angles." (2020), [Online]. Available: https://adipandas.github.io/posts/2020/02/euler-rotation/.

[12] J. Zeitlhöfler, "Nominal and observation-based attitude realization for precise orbit determination of the jason satellites," Ph.D. dissertation, Jun. 2019, p. 16. [Online]. Available: https://mediatum.ub.tum.de/doc/1535899/file.pdf.

[13] GuerrillaCG. "Euler (gimbal lock) explained," YouTube. (Jan. 14, 2009), [Online]. Available: https://www.youtube.com/watch?v=zc8b2Jo7mno.

[14] R. Mukundan, "Quaternions : From classical mechanics to computer graphics, and beyond," 2002. [Online]. Available: https://atcm.mathandtech.org/EP/2002/ATCMA107/fullpaper.pdf.

[15] J.-L. Blanco, "A tutorial on SE(3) transformation parameterizations and on-manifold optimization," University of Malaga, Tech. Rep. 012010, 2010, repository: https://github.com/jlblancoc/tutorial-se3-manifold. [Online]. Available: http://ingmec.ual.es/~jlblanco/papers/jlblanco2010geometry3D_techrep.pdf.

[16] K. Walchko, *Squaternion*, version 0.3.3, repository: https://github.com/MomsFriendlyRobotCompany/squaternion, Nov. 21, 2021. [Online]. Available: https://pypi.org/project/squaternion/.

[17] M. Grant, "Quick start for beginners to drive a stepper motor," *Freescale Semiconductors*, vol. 6, 2005. [Online]. Available: https://www.nxp.com/docs/en/application-note/AN2974.pdf.

[18] M. Frey. "Simple, two-pole, brushed, dc motor." (2015), [Online]. Available: https://commons.wikimedia.org/wiki/File:Animation_einer_Gleichstrommaschine_(Variante-Langsam).gif.

[19] *Wantai mini stepper product specifications 42byghw811*, AG26.8, Wantai. [Online]. Available: https://www.phidgets.com/productfiles/3312/3312_0/Documentation/3312_0_Datasheet.pdf.

[20] Rototron, *Raspberry pi stepper motor tutorial*, Jun. 2017. [Online]. Available: https://www.rototron.info/raspberry-pi-stepper-motor-tutorial/.

[21] *Drv8880 2-a stepper motor driver with smart tune*, SLVSD18C, Revision C, Texas Instruments, Aug. 2017. [Online]. Available: https://www.ti.com/lit/ds/symlink/drv8880.pdf.

[22] L. Cabrera-Quiros, E. Gedik, and H. Hung, "No-audio multimodal speech detection in crowded social settings task at mediaeval 2018," in *Mediaeval 2018 Workshop*, 2018. [Online]. Available: http://pure.tudelft.nl/ws/portalfiles/portal/50324359/47773175_MediaEval_18_paper_3.pdf.

[23] Python Software Foundation. "Sunsetting python 2." (), [Online]. Available: https://www.python.org/doc/sunset-python-2/.

[24] Anaconda, Inc. "Conda." (), [Online]. Available: https://docs.conda.io.

[25] Biezl. "A one piece shaft collar, secured with a set screw." (Sep. 2008), [Online]. Available: https://commons.wikimedia.org/wiki/File:Stellring.svg.

[26] RWTH Aachen University. "Phyphox, your smartphone is a mobile lab." (), [Online]. Available: https://phyphox.org/ (visited on 01/22/2022).

[27] "Attitude sensor - phyphox." (Apr. 27, 2021), [Online]. Available: https://phyphox.org/wiki/index.php/Attitude_sensor (visited on 01/22/2022).

[28] European Commission. "Responsible research & innovation." (Dec. 7, 2020), [Online]. Available: https://ec.europa.eu/programmes/horizon2020/en/h2020-section/responsible-research-innovation (visited on 01/23/2022).

[29] zonmw. "Fostering responsible research practices." (), [Online]. Available: https://www.zonmw.nl/en/research-and-results/fundamental-research/programmas/programme-detail/fostering-responsible-research-practices/ (visited on 01/23/2022).

[30] G. . Chassang, "The impact of the eu general data protection regulation on scientific research," *ecancermedicalscience*, vol. 11, 2017. DOI: 10.3332/ecancer.2017.709.