The impact of a personal monitoring device with smart-cueing on progressive postural variation, sedentary time and associated health outcomes

Subtopic:
Hardware and Communication

Final Report

Author:
Alexander Tonino
Bachelor's program Medical Device Technology
at the University of Applied Science Upper Austria,
School of Applied Health and Social Sciences
Garnisonstraße 21, 4020 Linz, Austria

Date of Submission:
Berkeley, on the 31st of January 2018

Academic Advisors:
FH-Prof. DI Dr. Robert Merwa
University of Applied Science Upper Austria,
Department of Medical Engineering

Carisa Harris Adamson, PhD, CPE, PT
Assistant Professor, Department of Medicine Director,
UCSF/UCB Ergonomics Research & Graduate Training Program

Table of Contents

List of Figures

List of Tables

List of Codes

List of Equations

1 Initial Situation / Problem

Today most office jobs entail long, uninterrupted periods of work in front of a computer terminal. Numerous studies have shown that increased activity during this work is not only beneficial to the health of the worker, but also increases the quality of the work [1]–[3]. Therefore, posture and activity monitoring are an important part of human health monitoring and a dominant application for body-mounted sensor networks [4]–[6]. There is a trend towards wearable systems to monitor and log everyday activities. However, those networks are mostly heavy, unergonomic and difficult to mount and therefore not usable in an everyday setting [7]. On the other hand, mobile devices such as smartphones and smartwatches do not provide reliable and real-time information about the posture and current activity [8]. To address this problem and increase the activity of the worker, a lightweight tracking device has been developed, which can track the user activity, and transmit that information to a mobile phone app that allows the user access to the results of the measurements. The small off-the-shelf sensor MetaMotionR which features an accelerometer, a pressure sensor and is connectable to a smart phone via Bluetooth® is used [9]. This sensor is mounted in the pants pocket of the user and used to simultaneously log the accelerometer and the pressure sensor data and send it to the phone, where an algorithm performs the classification of the various postures/activities. To enable fast progress of the development and the comparison of the result of the algorithm to a gold standard, it was necessary to develop the algorithm in MATLAB®. Therefore, an approach to automatically distinguish between several postures like “Standing”, “Sitting”, “Perching” and activities as like “Walking”, “Running”, “Biking” based on a single sensor placed in a user’s pocket is proposed.

The content of this thesis covers the development of a device that is capable of detecting the actual posture, as well as diverse everyday activities such as biking or walking. The algorithm used to distinguish the different postures and activities is described and validated in this paper.

In future the described algorithm should be run by the sensor itself. The sensor will stream data to a smart phone where it is summarized and viewable by the user via an app. The app can be set to different target goals parameters and real-time progress to goal can be viewed. The app will include links to a survey tool to collect data on task, discomfort, and perceived barriers to adhering to smart cues. The design and development of the app is not part of this paper.

1.1 Introduction Posture Detection

There are many different approaches to building a posture detection algorithm. Many approaches are implemented via video object detection [10]–[14] or using a Body Area Sensor Network with several sensors placed over the body [5]–[7], [15], [16].

Accelerometers are broadly accepted sensors for wearable devices to quantity and measure physical activity. Those solutions have demonstrated their usefulness and practicability. Available single sensor solutions for posture and activity detection and logging are used for many different studies. In Table 1, commonly used products and solutions for posture and activity detection are shown. Those sensors are used and validated in numerous studies [17]–[23].

Product [Manufacturer] Reference	**Positions on body**	**Provided data – epoch length**
Actigraph Model 7164 [Actigraph LLC Pensacola, FL]	Hip, Ankle, Wrist	Activity counts, Energy expenditure– 5-60 seconds
Actical [Mini-Mitter Sunriver, OR]	Hip, Ankle, Wrist	Activity counts, step counts, Energy expenditure – 15-60 seconds
SenseWear [BodyMedia Inc]	upper arm	Activity counts, Energy expenditure, Metabolic equivalent (MET) sleep duration – n.d.
IDEEA [MiniSun]	Feet, Waist, upper leg	Activity counts, Energy expenditure, gait types, some postures – n.a.
CT1/RT3 [StayHealthy Inc.]	Waist	Activity counts, Energy expenditure, MET – n.a.
GT1M Actigraph [Actigraph LLC]	Waist	Activity counts, step counts, MET, Activity counts
activPAL [PAL Technologies Ltd.]	Upper leg	Sedentary and upright time, steps, stepping time, cadence, sit-to-stand activities, MET, PAL, kCal – ~15 seconds
Mathie et al [24]	Hip	Activities: fall, walking Postures: sit, stand, lying – n.a.

Table 1: Overview of commonly-used accelerometer based activity and posture monitors [24]–[26]

However, those solutions are not for everyday use. On one hand they are not equipped with a wireless transmitting solution, to send data to a phone. On the other hand, they are not easy to activate, and they must be taped on the participant's body. An example is shown in Figure 1.

Figure 1 IDEEA system of MiniSun placed on a proband of the validation study of Maffiuletti et al. [22]

Ideally, this hardware and software design process should lead to an everyday wearable device which is easy to use, easy to handle and easy to connect. Furthermore, it should include a well-tested algorithm for posture detection.

1.2 Posture Detection Objectives / Specification

Due to the targeted small form factor of the device and the advanced claims in usability, it is not preferable to build a sensor network with more than one sensor. Due to this restraint in sensor placement and number of sensors the amount of available information is limited. The goal is to design, implement and test an algorithm that can sense a predetermined number of states with satisfactory detection accuracy and reacts to a change in posture within a second.
The algorithm is built to distinguish the postures and movements shown in Table 2.

Name of activity	**Description**
Sitting	Sitting calm as well as sitting active (kneeling chair or perching chair)
Standing	Standing still as well as standing and moving
Perching	Using a leaning or perching chair
Walking	Walking
Running	Jogging and running
Biking	Sitting and pedalling on a bike
Climbing Stairs	Ascending and descending of stairs

Table 2: Different Activities to be detected with the Algorithm

2 Methods / Evaluation

The development of the algorithm originates in an analysis of motion and orientation changes of the body during changes in posture. The observations and results of this analysis lead to the design of the algorithm. Therefore, the algorithm is based on feature extraction of the provided signals by the used sensor. After development of the algorithm, the accuracy must be evaluated.

For the evaluation of the algorithm there will be a small study with approximately 15-20 representative test subjects. The study is based on a specific procedure of activities and postures each of the participants should perform. These procedures are filmed and logged with sensors at least two different positions at the body (e.g. belt, pants pocket, upper leg). These two datasets will be compared *post hoc* for accuracy. After synchronizing the data and the video, the result of each one of the three main parts and the conclusion of the combined logic is superimposed on to the video, and evaluated in several accuracy aspects.
The result is a table for each stage, providing the accuracy for static activities (without transitions) as well as the accuracy with the transitions (Table 3). An additional table where each misinterpreted Posture/Activity is documented is also generated (Table 4).
In addition to the comparison between the actual posture and the predicted posture, each participant is equipped with three state-of-the-art Smartwatches (Apple iWatch 3, Samsung Gear S3 and Fitbit Ionic). The step-counter of these smartwatches will be compared to the step-counter of the algorithm.

Special Pattern Recognition: Cases	Accuracy (%)	Accuracy (%)
Biking	Min/Max/Mean	Min/Max/Mean
Walking	Min/Max/Mean	Min/Max/Mean
Perching/Leaning Chair	Min/Max/Mean	Min/Max/Mean
…	Min/Max/Mean	Min/Max/Mean

Table 3: Example Information Table for each Stage

Target\Actual	Sit	Stand	Walk
Sit	99%	1%	0%
Stand	1%	98%	1%
Walk	0%	1%	99%

Table 4: Example Confusion Matrix

The following protocol is used for the evaluation of each work stage used for the Posture Detection Algorithm (Figure 3). Each participant performs all of these activities (Table 5).

Nr.:	**Activity**	**Duration (s)**	**Reason**
0	Tip on all Sensors for Sync	5	Calibration
1	Sitting	5	
2	Standing	5	
3	Walking	10-15	Determine Difference of Activity Level
4	Jogging	10-15	
5	Sprinting	5	
6	Jogging	5-10	
7	Walking	5-10	Change Walk/Sit/Walk
8	Sitting	5-10	
9	Walking	5-15	
10	Stairs UP	1-2	
11	Walking	15-25	Simulate Office
12	Sit on Office Chair	20-30	
13	Lean Back in Chair	10-20	
14	Walk to perching Chair	5-10	
15	Sit/Lean on perching Chair (Task on PC)	20-30	
16	Standing Workplace	10-20	
17	Walk to Couch	15-25	
18	Sit on Couch	30-40	
19	Walk	5-10	Simulate Way to Office
20	Stairs down	2	
21	Walk	5-10	
22	Biking seated	40	
23	Biking seated active/passive	40	
24	Walking	20-25	
25	Stairs Up	10	
26	Stand	5-10	
27	Stairs Down	10	
28	Walk	40-60	
29	Standing	5	
30	Tip on all Sensors for Sync	5	

Table 5: Evaluation Protocol

3 Posture Detection Algorithm

The considered algorithm is able to detect several activities (Table 2). To enable it to do that, the whole algorithm is composed of five separate work stages (Figure 3).

- Data Preparation:
  In this stage of the algorithm the data is prepared for the stages after. This contains filtering the raw data, removing offset, merging signals and several other tasks.

- Standing/Sitting
  In this stage the prepared data is used to distinguish if the person is in a sitting or an upright position.

- Special Pattern Recognition
  The Special Pattern Recognition is supposed to detect special states of activity like walking, biking, or a leaning chair (Figure 2).

- Level of Activity
  This part is used to distinguish between low load activities such as walking or standing, and high load activities such as running or sprinting.

- Combine Logic
  In this section of the algorithm, the results of the three main parts (Standing/Sitting, Special Pattern Recognition and Level of Activity) are combined to a single conclusive perception. This is used to improve the accuracy and calculate the final activity. For example, preventing classification of the state as “Walking” while the person is seated.

Figure 2: leaning seat (https://www.focalupright.com)

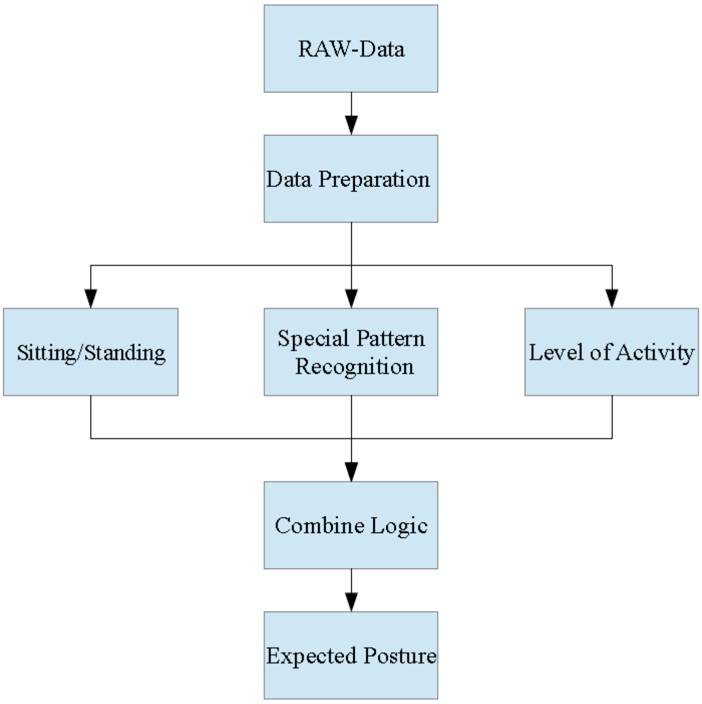

Figure 3: Flowchart Posture Detection

3.1 The Sensor

The sensor used in this study is MetaMotionR (MBIENTLAB INC San Francisco, CA., USA). It features a 3-axis accelerometer, which streams the raw data via Bluetooth LE to a Smartphone or a PC. While it features a small form factor, it also includes a USB rechargeable battery. It is placed on a Clip-On to enable the user to clip it onto the belt or into the pocket.

The sensor features many sensors, such as a humidity sensor, barometric pressure sensor, accelerometer, gyroscope, magnetometer, ambient light sensor and a thermistor. However, it is the accelerometer and the barometric pressure sensor data that is used for the algorithm [9].

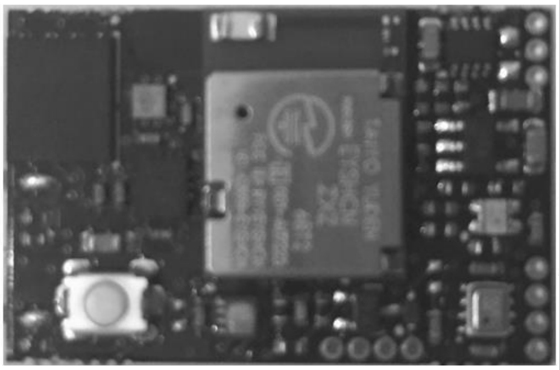

Figure 4: Top view of MetaMotionR without housing [9].

Figure 5: Left, sensor. Right, sensor placed on rectangular Clip-On.

3.2 Data Preparation

The Data Preparation is used to calculate and provide several signals and values out of the input sensor data to following steps of analysis.

3.2.1 Offset Subtraction and Magnitude

Part of the calculated additional information is the magnitude of the derived 3D vector. The derivation is used to remove the signal offset caused by gravity $\vec{g}$, (Equation 1). Due to the further signal processing and pattern recognition it is not necessary to perform the reverse integral on the signal.

$$\frac{d}{dt}(\vec{a}(t) + \vec{g}) = \vec{a}'(t)$$

Equation 1: Remove offset by derivation of the Signal [27], where $\vec{a}$ is the vector output of the accelerometer

The magnitude of a vector is calculated by application of Pythagoras' theorem [27] (Equation 2).

$$\left|\begin{pmatrix} a'_x \\ a'_y \\ a'_z \end{pmatrix}\right| = \sqrt{{a'_x}^2 + {a'_y}^2 + {a'_z}^2}$$

Equation 2: Magnitude of the derived acceleration vector [27]

This provides the information of the acting acceleration while losing the direction information. Because of this, the calculated magnitude is independent of the direction, and can be filtered with two different filters. The first is used for the detection of steps while walking or running, and the second for the detection of the activity level.

3.2.2 Data Filtering

The filter used on the signals is a simple Moving Average Filter. A moving average filter has a smoothing effect, and shows the general trend of a signal depending on the size of the window [28]. With this filter a reduction of motion artefacts and noise is accomplished. However, it also broadens the effects of a high peak on the signal, and the longer the window, the worse the quality of the waveform [29]. Due to the simple implementation and fast performance the simplest moving average filter is used:

$$y[n] = \frac{1}{N}\sum_{i=0}^{N-1} x[n-i]\,, n = N, N+1, \ldots, L$$

Equation 3: Simple Moving Average Filter [28]

Where N is the widow size and L is the length of the data. In a further implementation of the algorithm, on a smart phone or on the sensor itself, it is not necessary to save the whole data. It is sufficient to save only as many data points as the longest window uses. This filter is used on the raw data of the accelerometer data and on the previously calculated data to smooth the signals while different sizes of windows for different tasks and analysis are used.

3.2.3 Signal Synchronization

Due to the filtering through averaging several data points, there is a time delay in the waveform of the signals (Figure 6, Table 6). To ensure the algorithm is consistent, the data must be aligned in time. To do this, the start and end points of the filters must be adjusted, such that they are centered relative to the buffer. This is then the same size as the longest filter (Table 7, Figure 7). This means that for a proper synchronization of the used signals it is necessary to only use only odd window sizes for the filters. Furthermore, it means that there will always be a delay of the detection that is about half the size of the longest filter.

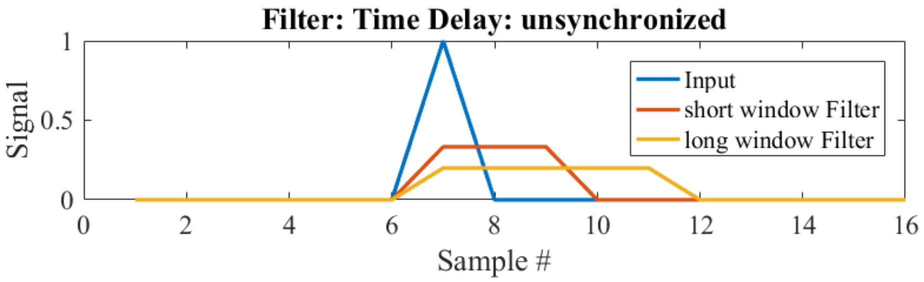

Figure 6: The effect of the size of the simple moving average filter on a signal.

raw Signal	N/A								
short window	S		A		E				
long window	S				A				E

Table 6: Representation of time delay of different sized filters: A is the currently active sample used for posture detection and N is the new sample in the system. S and E are the start and the end of the filter

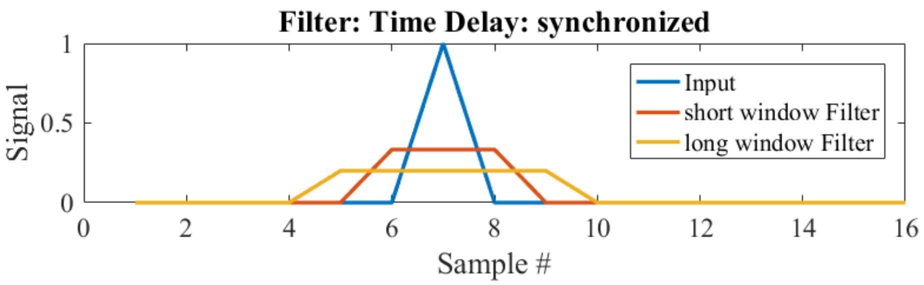

Figure 7:The synchronized, filtered signal

raw Signal	N				A				
short window			S		A		E		
long window	S				A				E

Table 7: Representation fixed time delay, A is the currently active sample used for posture detection and N is the new sample in the system. S and E are the start and the end of the filter

3.2.4 Vector system orientation and angle calculation

For further analysis of the data, it is useful to orientate the coordinate system to a more useful direction. The coordinate system in this case is turned in such a way that the gravity vector during sitting is pointing along the X-axis. Then, it is useful to turn the system such that the change from “Sitting” to “Standing” is visible in the plane of the X- and Z-axes.

To turn the system in described way it is necessary to transform the acceleration vector of the “Sitting” state, which was gathered during the calibration from Cartesian into spherical coordinates (Figure 8, Equation 4). This will provide the information of the azimuthal (θ) and elevation (φ) angles (Figure 8, Equation 4) necessary to orient the system [30].

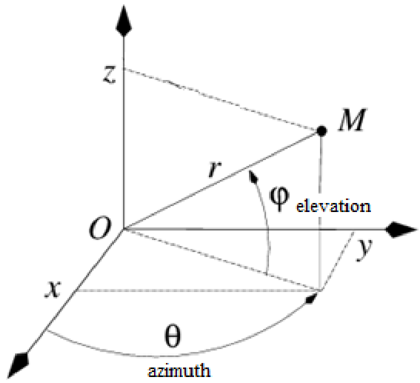

Figure 8: Cartesian coordinates P(x,y,z) and spherical coordinates P(λ, φ, r) (Source: Adapted from http://www.iecl.univ-lorraine.fr)

$$\theta = arctan2(x, y)$$

$$r = \sqrt{x^2 + y^2}$$

$$\varphi = arctan2(r, z)$$

Equation 4: Transformation Cartesian coordinates P(x,y,z) to spherical coordinates P(θ, φ, r) [30] with the function arctan2 [31].

After determination of φ and θ, the incoming vectors of the accelerometer are rotated using the Euler rotation [30]. It is necessary that the first axis of rotation is the Z-axis (Equation 5). By doing this the outcoming vector will be in the XZ-Plane, as explained above. After that the vector is rotated in the Y-axis (Equation 6). This will make the X-axis align with the vector of gravity in the case of “Sitting” (Equation 7).

$$Mz = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Equation 5: Euler rotation matrix for a rotation in Z axis counterclockwise[30]

$$My = \begin{pmatrix} \cos(-\varphi) & 0 & \sin(-\varphi) \\ 0 & 1 & 0 \\ -\sin(-\varphi) & 0 & \cos(-\varphi) \end{pmatrix}$$

Equation 6: Euler rotation matrix for a rotation in Y axis clockwise. [30]

$$\overrightarrow{a_{new}}(t) = (\overrightarrow{a_{raw}}(t) * Mz) * My$$

Equation 7: Turning the incoming acceleration vector into new coordinate system.

After these rotations, the system must be turned a final time to align the XZ-plane with the two states “Sitting” and “Standing”. To do this, it is necessary to calculate the angle of the “Standing” vector in the YZ-plane (Figure 9, Equation 8, Equation 9). After that the Euler rotation matrix for the rotation in X-axis is calculated (Equation 10).

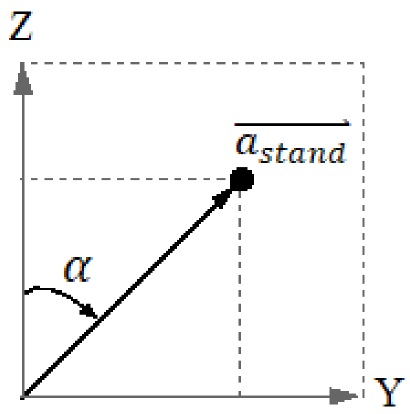

Figure 9: Angle α in the YZ-plane of the $\overrightarrow{a_{stand}}$ vector. In this graphic the vector is already rotated with Mz and My

$$\overrightarrow{a_{stand}} = (\overrightarrow{a_{rawstand}} * Mz) * My$$

Equation 8: Transforming the vector into the coordinate system, where the vector of gravitation is aligned with the X-axis.

$$\alpha = \arctan2(a_{stand\,Z}, a_{stand\,Y})$$

Equation 9: Calculation of the angle α as seen in Figure 9 with the function arctan2 [31]

$$Mx = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Equation 10: Euler rotation matrix for a rotation of α in X-axis counter clockwise. [30]

After combining the three rotation matrices (Equation 11), the acceleration for all data points is oriented toward the X-axis during “Sitting” and is in the XZ-plane during “Standing”. Thus, all the necessary information to process the incoming acceleration data in the same orientation has been collected.

$$M = (Mz * My) * Mx$$

Equation 11: Combination of all the Euler matrices in a single rotation matrix

Once the incoming filtered vector is rotated in this way, the calculation of the angles can be performed. As mentioned, every movement between the states “Standing” and “Sitting” is taking effect in the XZ-plane. Therefore, an angle of the acceleration vector (β) of 0° in this plane will be calculated during “Sitting” ($\overrightarrow{a_{tSit}}$), while another angle bigger than that is calculated during “Standing” (Figure 10). The exact value is calculated during calibration by rotating the vector $\overrightarrow{a_{rawstand}}$ with M to $\overrightarrow{a_{tStand}}$ (Equation 12), and calculating the angle in the XZ-plane with the arctan2 function (Equation 13). This provides β_{stand}, which is the angle of the sensor/leg during “Standing”. β is calculated for each incoming sample, and saved in a buffer as an additional information source. Although this signal is not the actual hip/upper leg angle it can be used to detect “Biking” or “Sitting in a perching chair” (3.5 Special Pattern Recognition). Fixation on the actual upper leg would provide a better estimate of the actual orientation of the femur.

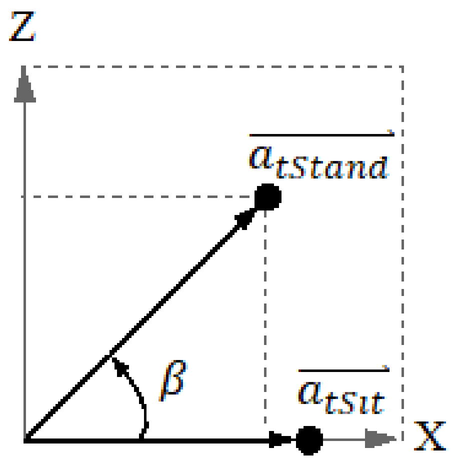

Figure 10: Angle β in the XZ-plane of the $\overrightarrow{a_{tStand}}$ and $\overrightarrow{a_{tSit}}$ vector. In this graphic the vectors are already rotated with M

$$\overrightarrow{a_{tStand}} = \overrightarrow{a_{rawstand}} * M$$

Equation 12: Turing of the vector recorded while standing during calibration

$$\beta = \arctan2(a_{stand\ X}, a_{stand\ Z})$$

Equation 13: Calculation of β with the function arctan2 [31]

3.3 Level of Activity (LOA)

The level of activity is provided as additional information of the current movement. Activities are separated into the different levels: light, moderate and vigorous [32], [33].These will referred to as Level 0, Level 1, and Level 2, respectively. Examples for these activities can be seen in Table 8. While this is inspired by the metabolic equivalent of task (MET) and features the same levels as this type of classification it doesn't measure the MET, due to missing heartrate and oxygen sensors [34]. However, it may be extended to a MET – calculation in future work.

Level 0 **Light**	**Level 1** **Moderate**	**Level 2** **Vigorous**
Playing most instruments	Walking briskly (4 mph)	Jogging at 6 mph
Walking slowly	Heavy cleaning (washing windows, vacuuming, mopping)	Shoveling
Sitting using a computer	Mowing the lawn (power mower)	Soccer game
Standing doing light work (cooking, washing dishes)	Cycling with light effort (10-12 mph)	Basketball game
Fishing while sitting	Tennis doubles	Tennis singles

Table 8: Different activities sorted into their level of activity. [33]

To distinguish between these levels of activity, there were several datasets recorded that included the states "Sitting"," Standing"," Walking"," Running" and "Sprinting". The sensor was placed in the right pocket of the test subject. That states were chosen to represent the different levels of activity. "Sitting" and "Standing" represent Level 0, "Walking" Level 1, and "Running" Level 2. "Sprinting" was used to determine a possible Level 3. The dataset of one test subject is presented in Figure 11.

For these datasets the root mean square (RMS) signal was calculated (3.2.1 Offset Subtraction and Magnitude). Those RMS signals were compared to each other to define thresholds to distinguish the different levels of activity. The results were used to create the following function (Code 1), which can detect different levels of activity for every new sample.

```
function  level = activitylevel(data,cal,ii)

if (data.filt(ii)>1.1)
    level = 3; % Sprint
elseif (data.filt(ii)>0.35)
    level = 2; % Running
elseif (data.filt(ii)>0.025)&&(data.filt(ii)<0.35)
    level = 1; % Active
else
    level = 0; %No Movement
end
end
```

Code 1: Implementation of the activity level distinguishing algorithm in MATLAB. data.filt is the previously filtered RMS signal. This function is called for every new sample in the dataset and gives a prediction how active the person is in this moment.

The result of Code 1 on the recent recorded data (Figure 11) is shown in Figure 12.

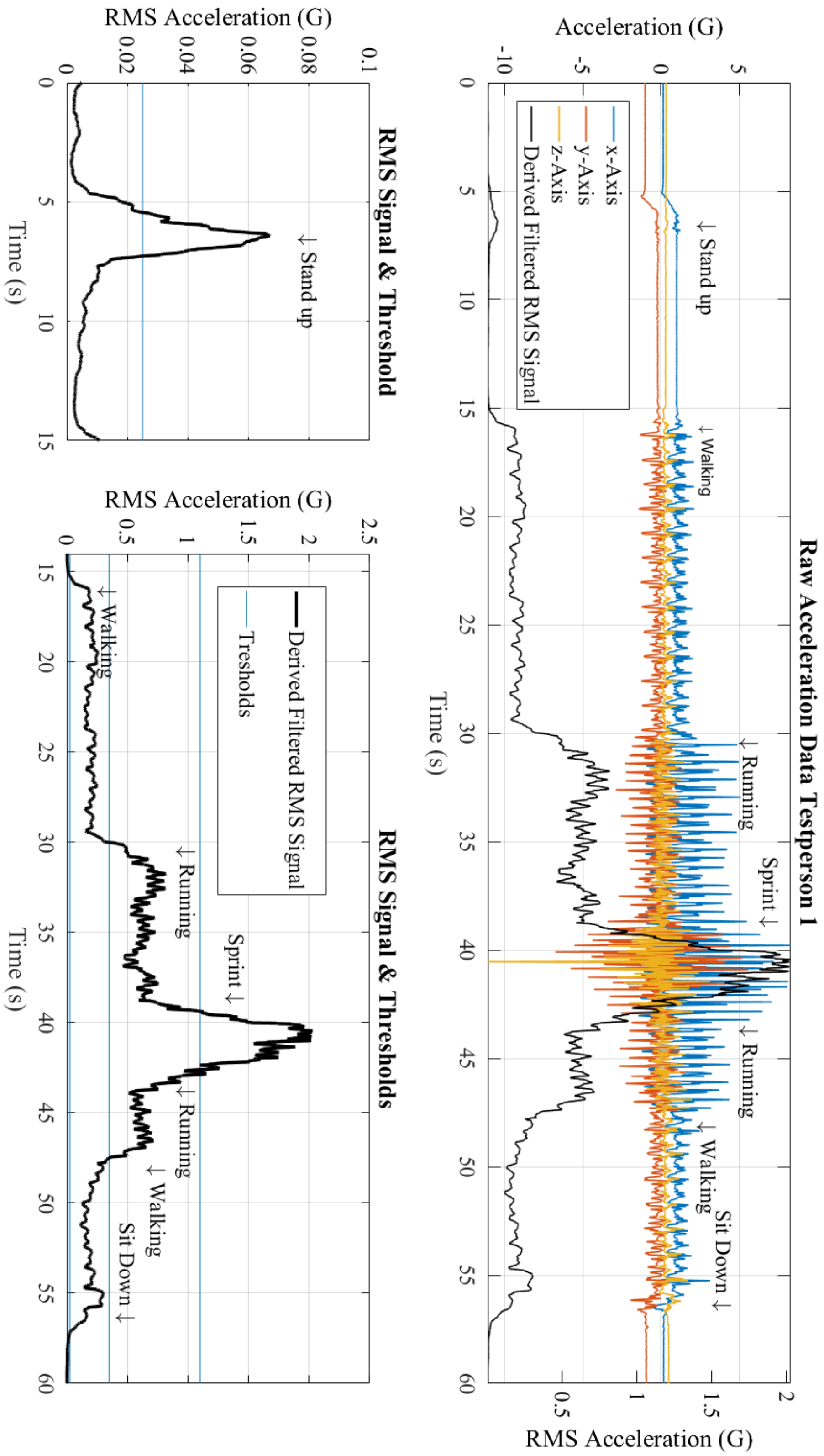

Figure 11: Dataset for distinguishing different Levels of Activity. Plot 1 shows the raw data of the Accelerometer (Sampling Rate 50Hz) and the derived filtered RMS Signal (3.2.1 Offset Subtraction and Magnitude). The second and third Plot shows only the RMS Signal and the thresholds which were arrived at by the comparison of several datasets.

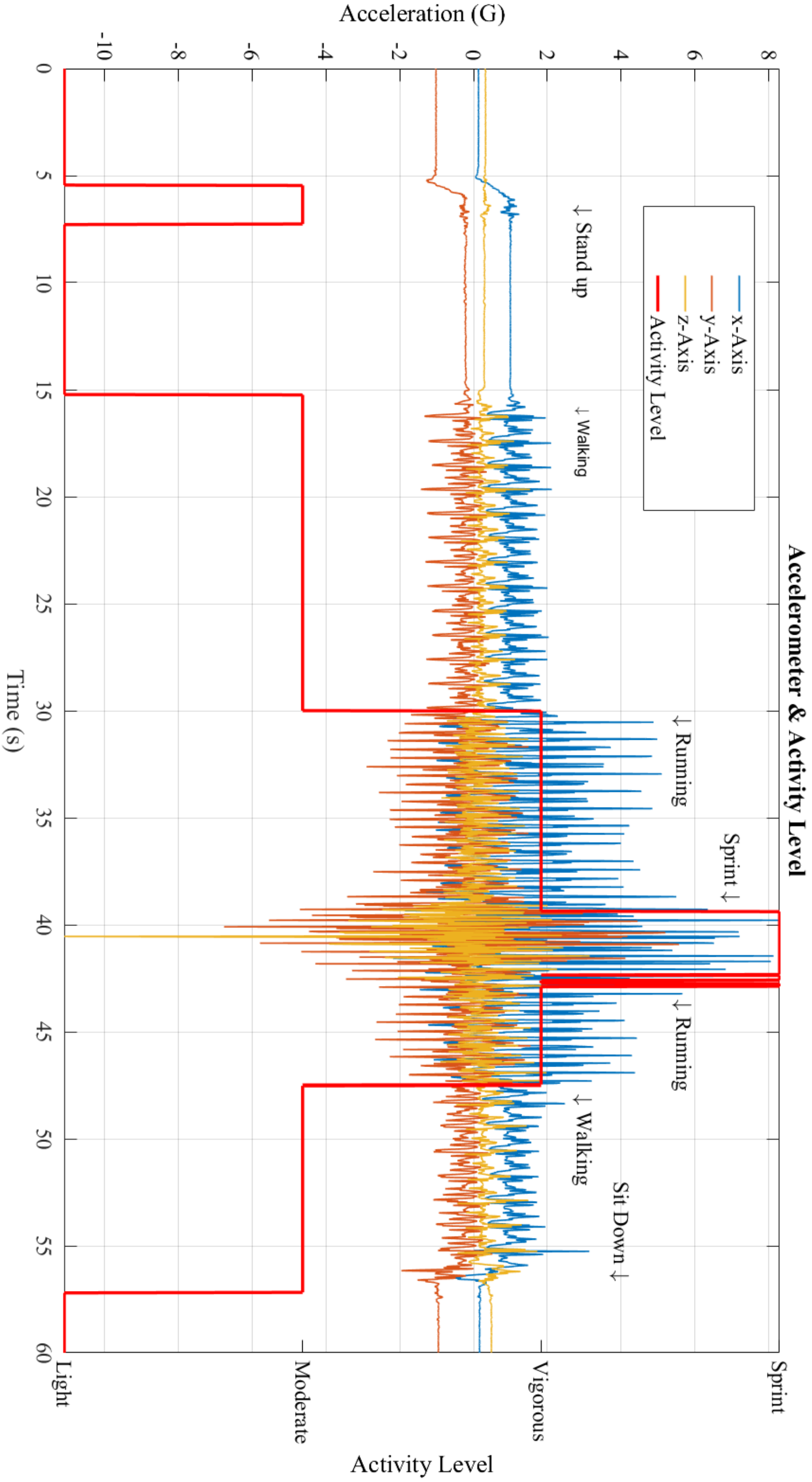

Figure 12: Distinguishing different levels of activity at the dataset of test person 1 using Code 1

3.4 Distinguishing “Standing” vs. “Sitting”

To enable the algorithm to detect the difference between sitting and standing there must be a significant changing of the orientation of the sensor regarding the gravity. Ideally this position is on the upper leg because it is the only part of the body that is changing its relative orientation during the transition between sitting and standing (Figure 13).

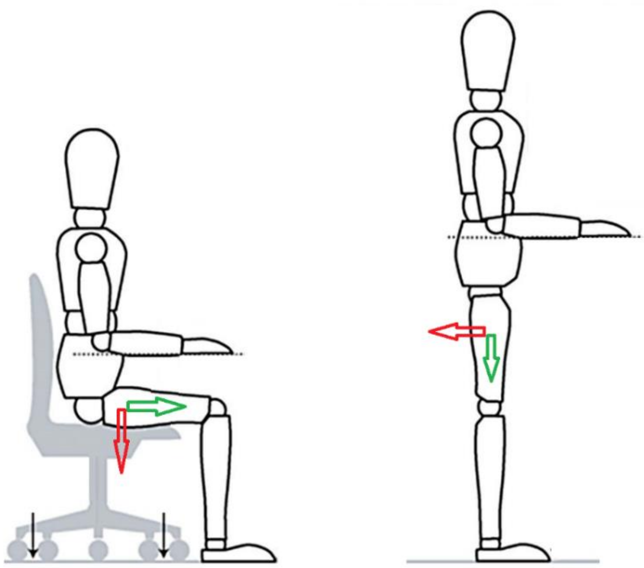

Figure 13: Changing of Orientation of the Upper Leg during a Transition between Sitting and Standing (Source: Adapted from http://www.interiorvues.com)

The accelerometer sensor is capable of detecting the gravity as well. So as long as there is no additional acceleration affecting the sensor, the magnitude of the 3D vector is approximately equal to 9.81m/s² or 1G. Furthermore, this vector is pointing in the direction of the gravitational acceleration (down).

The approach to distinguish sitting from standing is to calibrate the system in these two states of posture. With the information of the direction of the gravitational acceleration it is possible to decide whether the person is sitting or standing.

For this purpose, a dataset was recorded that included the three main states “Sitting”,” Standing”, and” Walking”. The sensor was placed in the right pocket of the test subject. An accelerometer sampling rate of 50Hz was chosen.
These data were imported in MATLAB (plotted in Figure 14), separated into these three states and replotted to show the difference between the states (Figure 15).

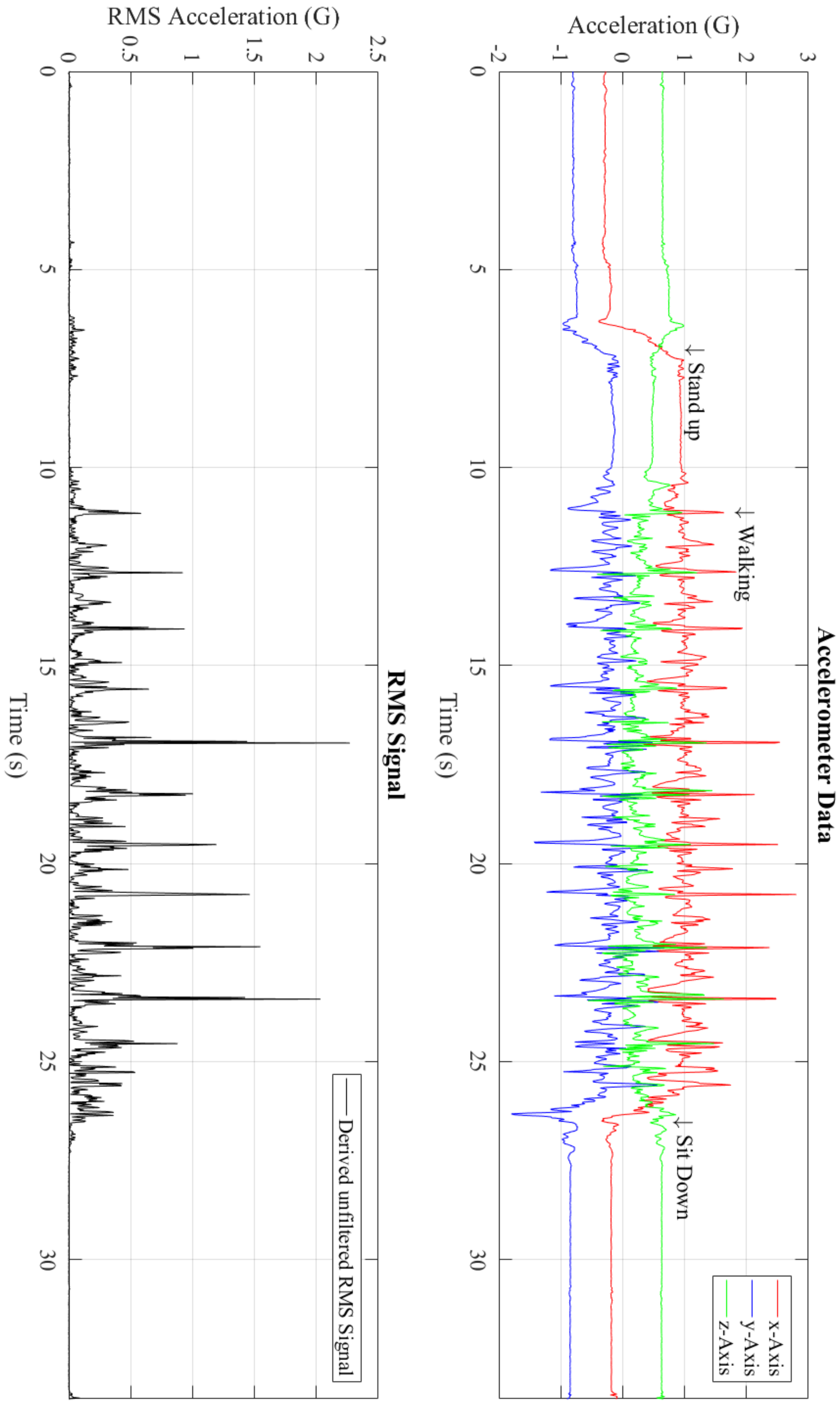

Figure 14: Dataset for Sitting until 6s, standing up, stand until 10s, walking until 26s, and sitting again. (Sampling Rate 50Hz). The first plot presents the raw accelerometer data in X-, Y-, and Z-axes. The second plot represents the absolute RMS acceleration without the offset.

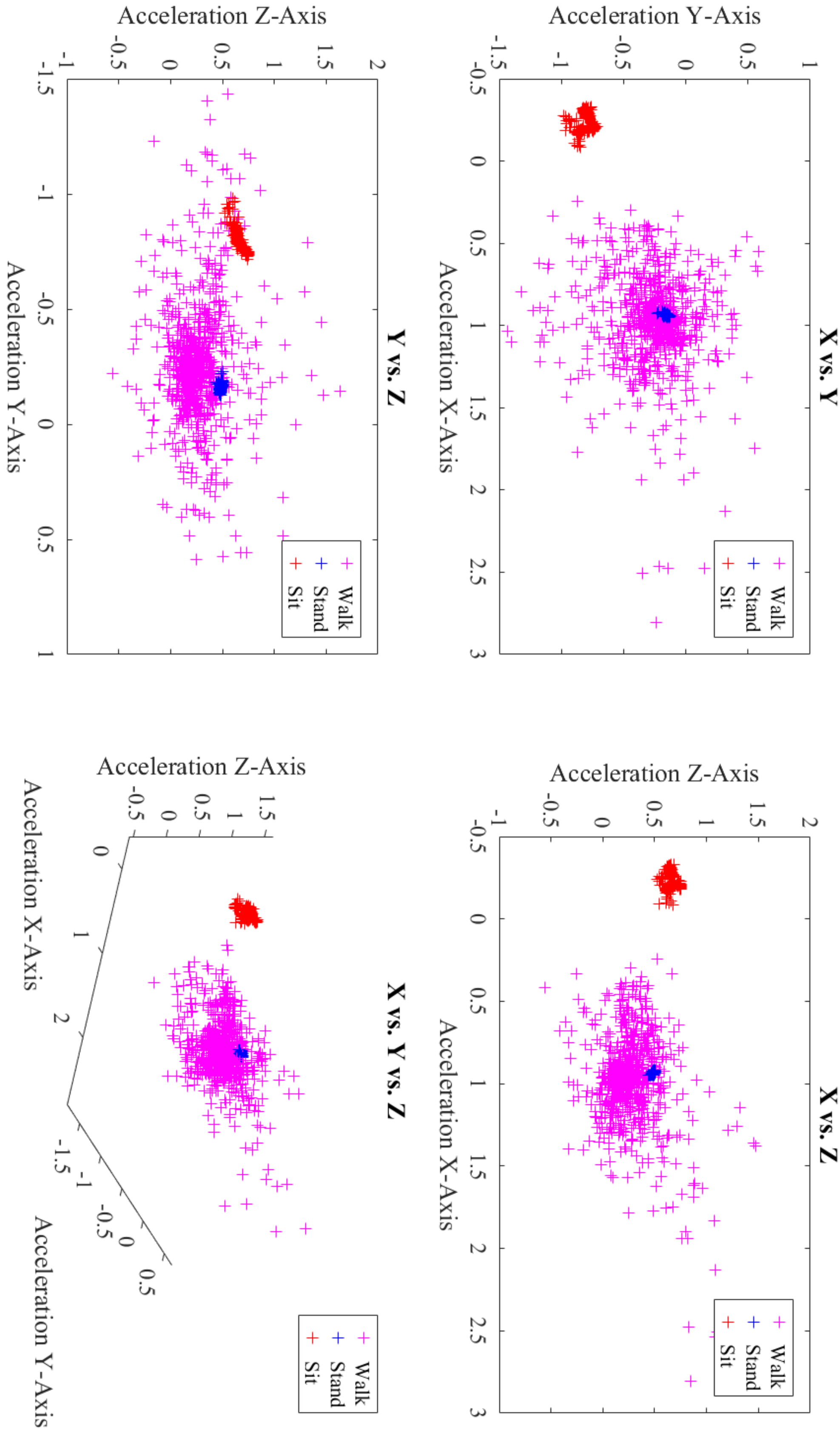

Figure 15: Separation of the data presented in Figure 14 in the states “Sit”, “Stand”, “Walk” (sampling rate 50Hz).

Additional evaluations (Table 9) have shown that there were changes in the orientation. The difference between "Sitting" and "Standing" is especially notable in the X- and Y-axes. The difference in the Z-axis is lower in magnitude because the axis of rotation was mainly the Z-axis. This data is to be interpreted as a single sample and should represent the function and the approach to distinguish these two states.

Acceleration (G)	Acceleration X	Acceleration Y	Acceleration Z	RMS Value
Median SIT	-0.2840	-0.7990	0.6450	1.0654
Mean SIT	-0.2742	-0.7903	0.6612	1.0663
Median STAND	0.9390	-0.1515	0.4800	1.0654
Mean STAND	0.9378	-0.1542	0.4792	1.0644
Median WALK	0.9570	-0.2520	0.2550	1.0219
Mean WALK	0.8923	-0.3150	0.2958	0.9914

Table 9: Median and Mean of the Accelerometer Data: "Sitting" vs. "Standing" vs. "Walking"

During walking or running the accelerometer data is constantly recording the sum of all accelerations, which results in a wide field of data points due to complex accelerations during movements. Through filtering the raw signals with a moving average filter it is possible to shrink this field. By comparison of the results of "Walking" with "Sit" and "Stand", the filtered "Walking" signal could be interpreted as standing in all three directions (Figure 16).

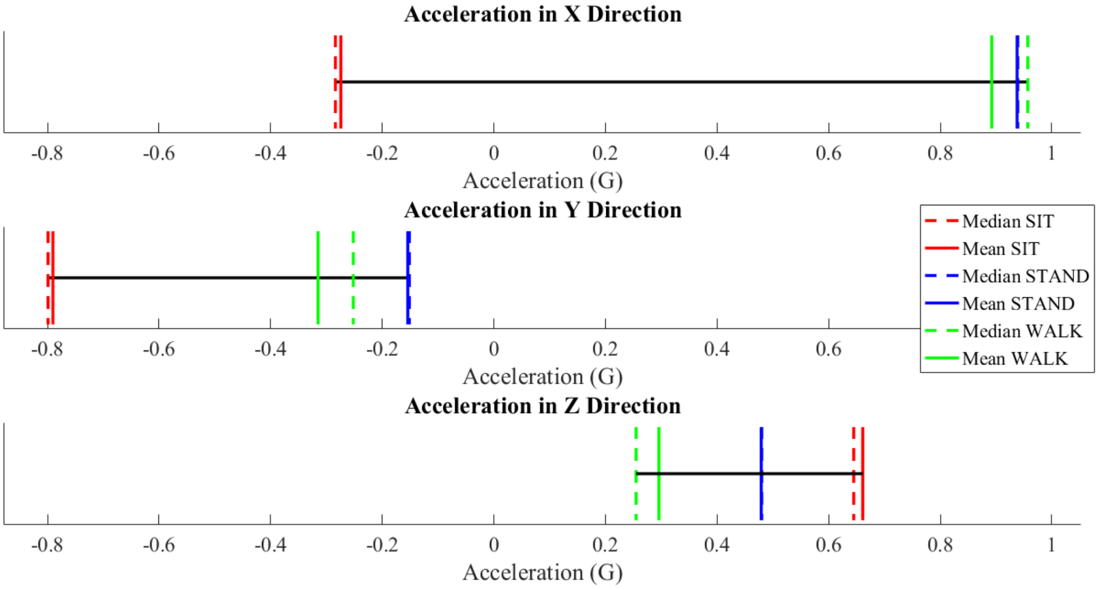

Figure 16: Comparison of "Sitting", "Standing" and "Walking" in the X-, Y-, and Z-axis.

3.4.1 Algorithm

To be able to differentiate these two states it is necessary to calibrate the system in each case, and save the different accelerations on each axis. Additionally, the incoming raw data from the accelerometer is filtered with a moving average filter.

After that, the decision whether the subject is “Sitting” or “Standing” is based on a simple comparison for each axis. Each filtered sample of the accelerometer is compared with the center value (decx, decy, decz) between the two calibrated points, to determine if it is within the range of “Sitting” or “Standing”.
The outcome of these three comparisons have to be united in one single state. To do this, it is necessary to consider the validity of each comparison.
Based on this outcome, it is possible to distinguish “Sitting” from “Standing” provided the sensor is turned in the transition between these two states. As seen in Figure 16, the change in the Z-axis is much smaller than in the X-axis. This makes it clear that a decision between “Sit” and “Stand” based on the Z-axis is less valid than the same decision based on the X-axis. Therefore, each decision is weighted with the difference in acceleration between the two states.

	Acceleration X	**Acceleration Y**	**Acceleration Z**
Mean SIT	-0.2742	-0.7903	0.6612
Mean STAND	0.9378	-0.1542	0.4792
Difference/weight	1.212	0.6361	-0.182
Center (dec)	0.3318	0.4723	0.570

Table 10: Difference between mean values of “Sitting” and “Standing” in all three axes and the Center value for detecting the different States

In addition to the weight of each comparison, the difference value in Table 10 is providing the information of which state has a higher acceleration value. This is necessary to interpret the data. If the difference is positive, the acceleration of the “Sitting” state is less than the acceleration of the “Standing” state. Thus, the difference has to be considered to decide if the comparison is positive or negative (Figure 17).

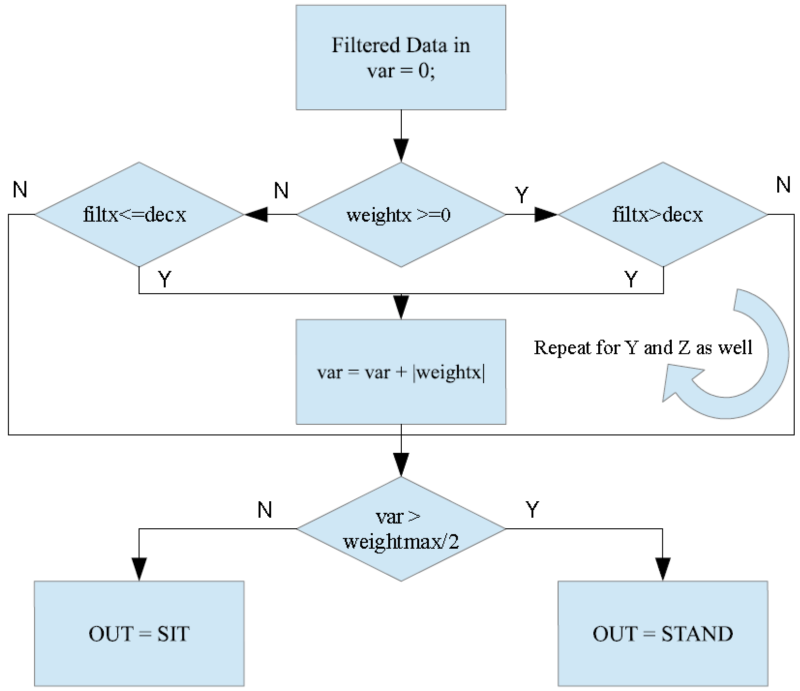

Figure 17: Algorithm Sitting vs. Standing; filtx, filty and filtz are the current filtered data point in each particular axis; weightx, weighty and weightz are the particular differences between the calibration data for Stand and Sit; weightmax is the sum of these differences.

```
function posture = calc_post(data,cal,ii)

posture = 0; %Initialize Posture Variable

% Sit Stand Axis X
if (cal.weightx >=0) %Decide if Sit or Stand has a higher Acceleration
    if data.filtx(ii)>cal.decx %If Signal is greater than calibration
for X
        posture = posture + cal.weightx; %Add weight to Posture
    end
else %Stand has a lower Acceleration
    if data.filtx(ii)<=cal.decx %If Signal is smaller than calibration
for X
        posture = posture + abs(cal.weightx); %Add weight to Posture
    end
end

% Sit Stand Axis Y
% Same procedure as for X
if (cal.weighty >=0)
    if data.filty(ii)>cal.decy
        posture = posture + cal.weighty;
    end
else %Stand has a lower Acceleration
    if data.filty(ii)<=cal.decy
        posture = posture + abs(cal.weighty);
    end
end
% Sit Stand Axis Z
% Same procedure as for X
if (cal.weightz >=0)
    if data.filtz(ii)>cal.decz
        posture = posture + cal.weightz;
    end
else
    if data.filtz(ii)<=cal.decz
        posture = posture + abs(cal.weightz);
    end
end

%Calculation of Posture 0 = Sit, 1= Stand
% if the sum of all weights is greater than the half of the maximum
weight it is "Stand" else it is "Sit"
if posture > ((abs(cal.weightx)+abs(cal.weighty)+abs(cal.weightz))/2);
    posture = 1;
else
    posture =0;
end
end
```

Code 2: Implementation of Figure 17 in MATLAB: This function is called for every new sample in the dataset and gives a prediction whether the person is sitting or standing based on the signal (data) and the calibration data (cal)

The result of using this code (Code 2) on the recent recorded data (Figure 14) is shown in Figure 18.

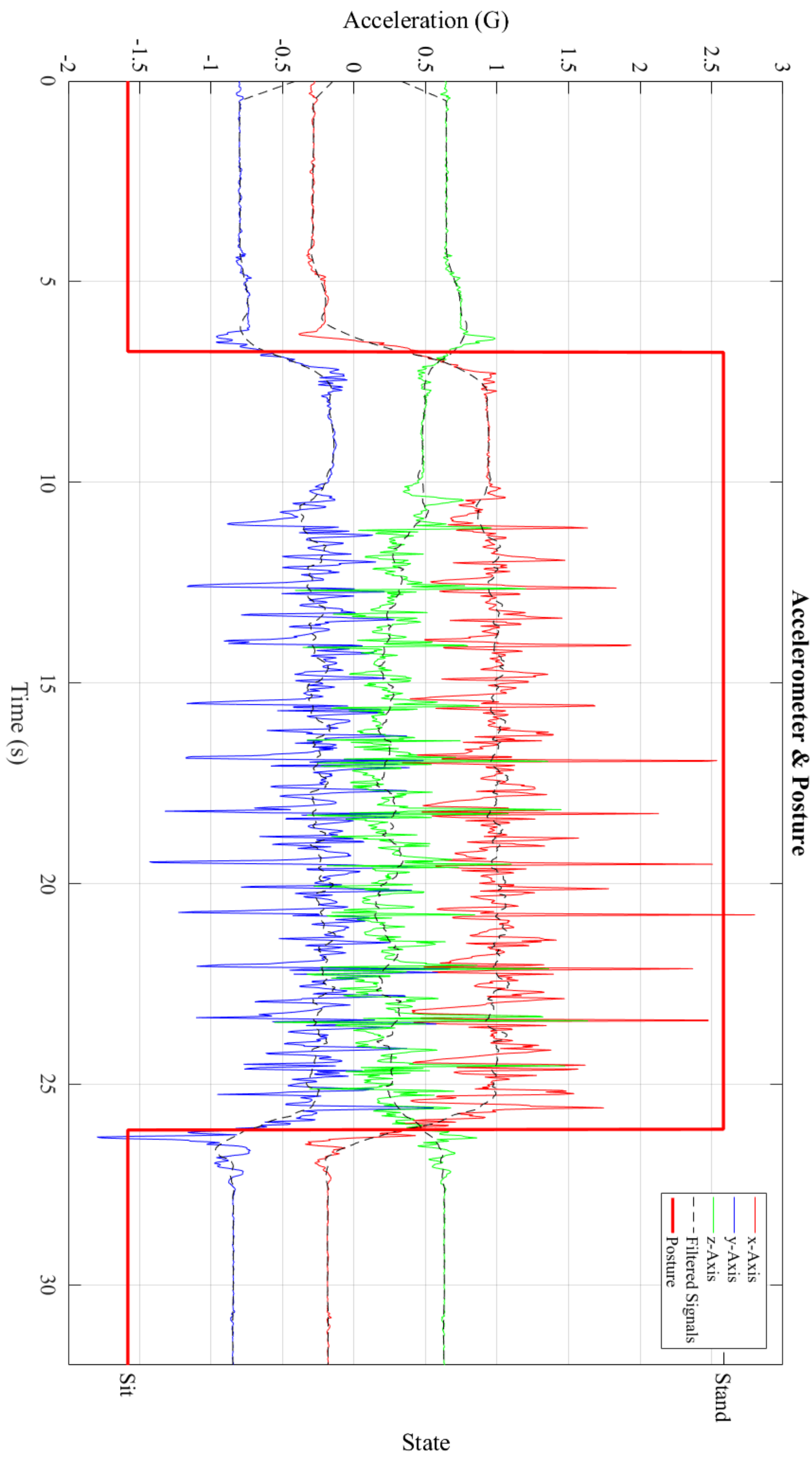

Figure 18: Distinguishing between “Sitting” and “Standing” after filtering the signal and the detection of “Sitting” vs. “Standing” as shown in Figure 17 and Code 2

3.5 Special Pattern Recognition

The Special Pattern Recognition is used for finding special events in the available sensor data. This includes the detection of steps for “Walking”, changes in the pressure for detecting “Stairs” and analysing β for the states “Biking” and “Perching”

3.5.1 Step Detection

The dataset used for distinguishing “Walking” (Figure 20) from other activities was previously used for the Distinguishing “Standing” vs. “Sitting” (3.4). Because of the acceleration during movements and the impact of the feet at initial contact during walking there have to be spikes on the RMS signal (3.2.1) of the accelerometer data [35]. To easily detect these spikes a threshold was chosen. If the signal is above this limit a flag is being raised and a timestamp is saved. Therefore, the timestamp of the last step can be taken in consideration during calculating the posture in the Combine Logic. The function for detecting steps (Code 3, Figure 19) is deactivated during the state “Sitting” and “Biking” do avoid unnecessary calculations.

There were several datasets recorded to test this detection, which included the states “Sitting”, “Standing”, “Walking”, “Running” and “Sprinting”. The value for the threshold was chosen by observation of these datasets subjects. The threshold is set to 0.25G.

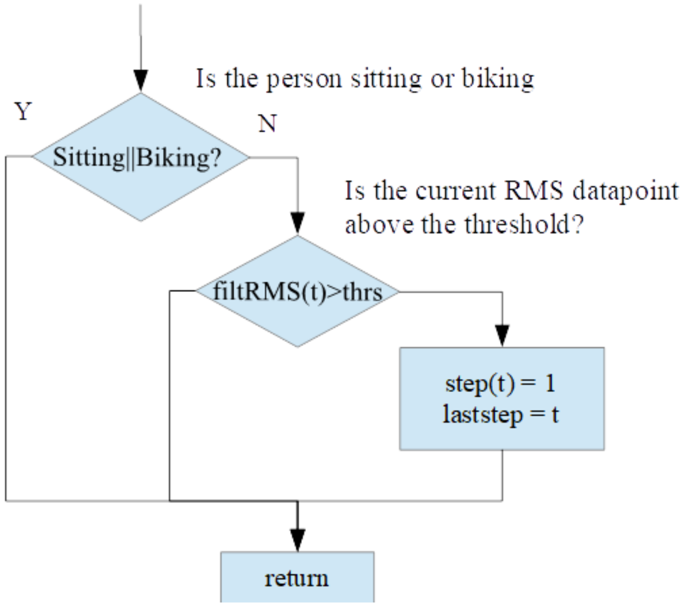

Figure 19: Flowchart Step-Detection. thrs is the threshold for the Step-Detection

```
%% Detecting steps on the signal
if data.standsit(ii)~=0||data.posture(ii-1)~=Biking
    if (data.filtstep(ii) >= cal.stepdet % Peaks on signal
        data.step(ii) = 1; % else it is 0
        data.laststep = data.t(ii); %Save Time of last Step
    end
end
```

Code 3: Detecting steps at RMS filtered signal. This function is called after the preparation of the data for every new sample of the accelerometer.

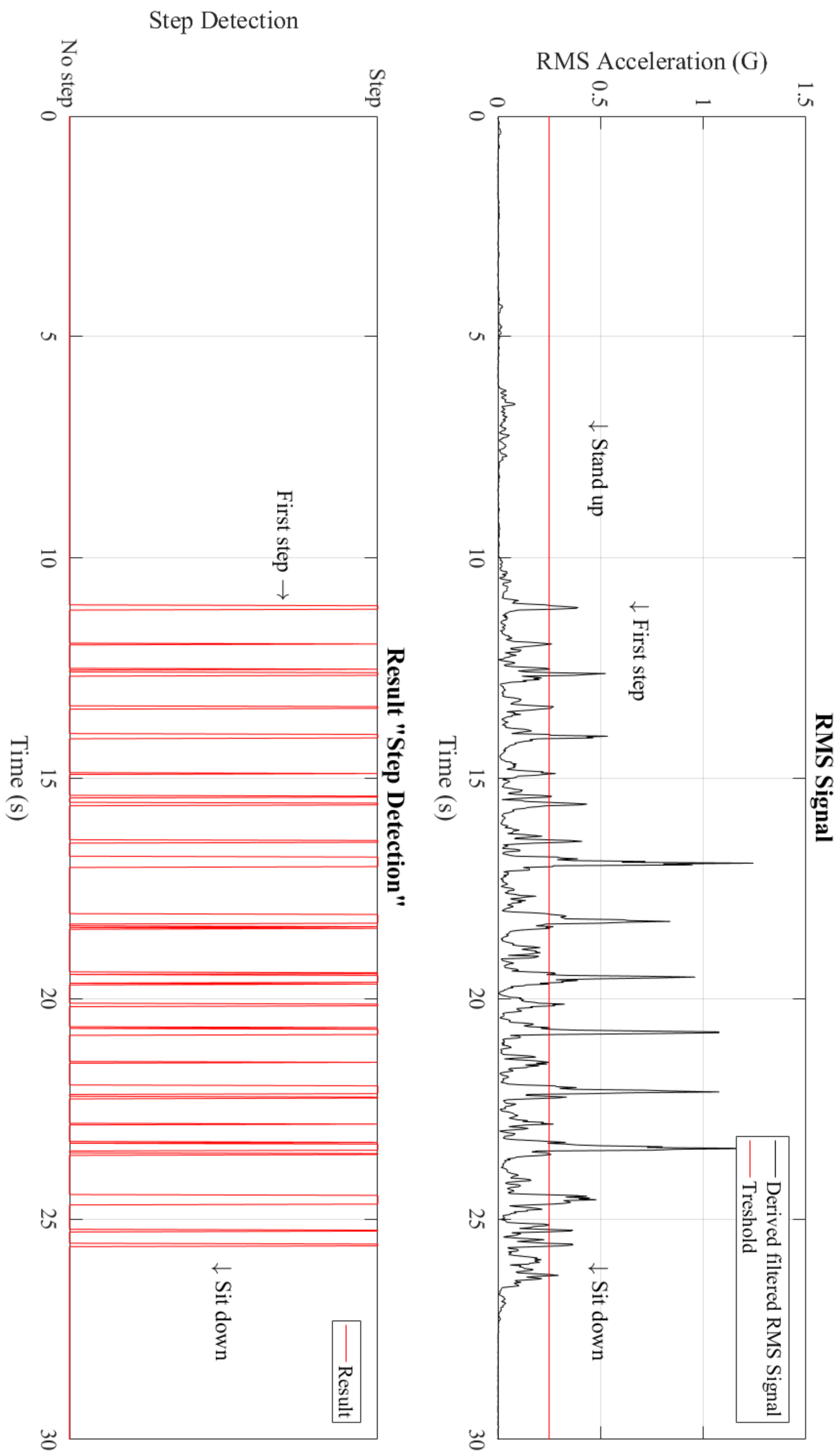

Figure 20: Dataset for detecting steps. This is the same data as used for 3.4 Distinguishing "Standing" vs. "Sitting" (Figure 14)

3.5.2 Stairs

The approach to detect “Stairs” is it to sense the difference of the barometric pressure, that changes during climbing stairs because the sensor is experiencing a change in height. The barometric formula is written as

$$p = p_0 * e^{\left(-\frac{\rho_0 * g}{p_0} * h\right)}$$

Equation 14: barometric formula [36]

with ρ_0 and p_0 as pressure and density of the air at sea level, respectively, which give an estimation of the pressure at every level of height. The BMP280 Digital Pressure Sensor can achieve 0.16Pa resolution [9], which is precise enough to detect changes in height of approximately 15cm.
The incoming pressure data of the sensor is derived and filtered to remove the offset and smooth the signal, which is influenced by accelerations due to physical limits of the hardware implementation in the microelectromechanical system (MEMS). There must also be a threshold to distinguish long-term changes in pressure due to changing of the weather or walking slightly ascending roads from actual stair climbing.

For this purpose, data were recorded while ascending stairs and descending stairs. The sensor was placed in the left pocket of the test subject. Due to different sample rates as the accelerometer a pressure sensor sampling rate of 13.51Hz was chosen (Figure 21). In this example it became obvious that even after filtering the signal and the derivation, it was not possible to distinguish the stairs with an easy comparison to thresholds. Therefore, it is necessary to collect 10 data points of the derived pressure signal, and calculate the mean pressure change in this period. If this mean pressure change is smaller than the “Stairs Up”-threshold, counter (a) is incremented and counter (b) is cleared. If it is over the “Stairs-Down”-threshold counter (b) is incremented and counter (a) is cleared. If one of these counters reaches a value of three or higher, “Stairs” is detected (Figure 22, Code 4).

This means that the algorithm is only able to detect “Stairs” when it takes the participant longer as 2.2s to climb or descend these stairs.

The delayed detection of “Stairs” is caused by a static debouncing (through assessment of counter values). Therefore, it would be possible to enhance the accuracy and reaction time with postprocessing of the result. This has not yet been implemented, and could be approached in an additional paper.

The result of this algorithm is shown in Figure 23.

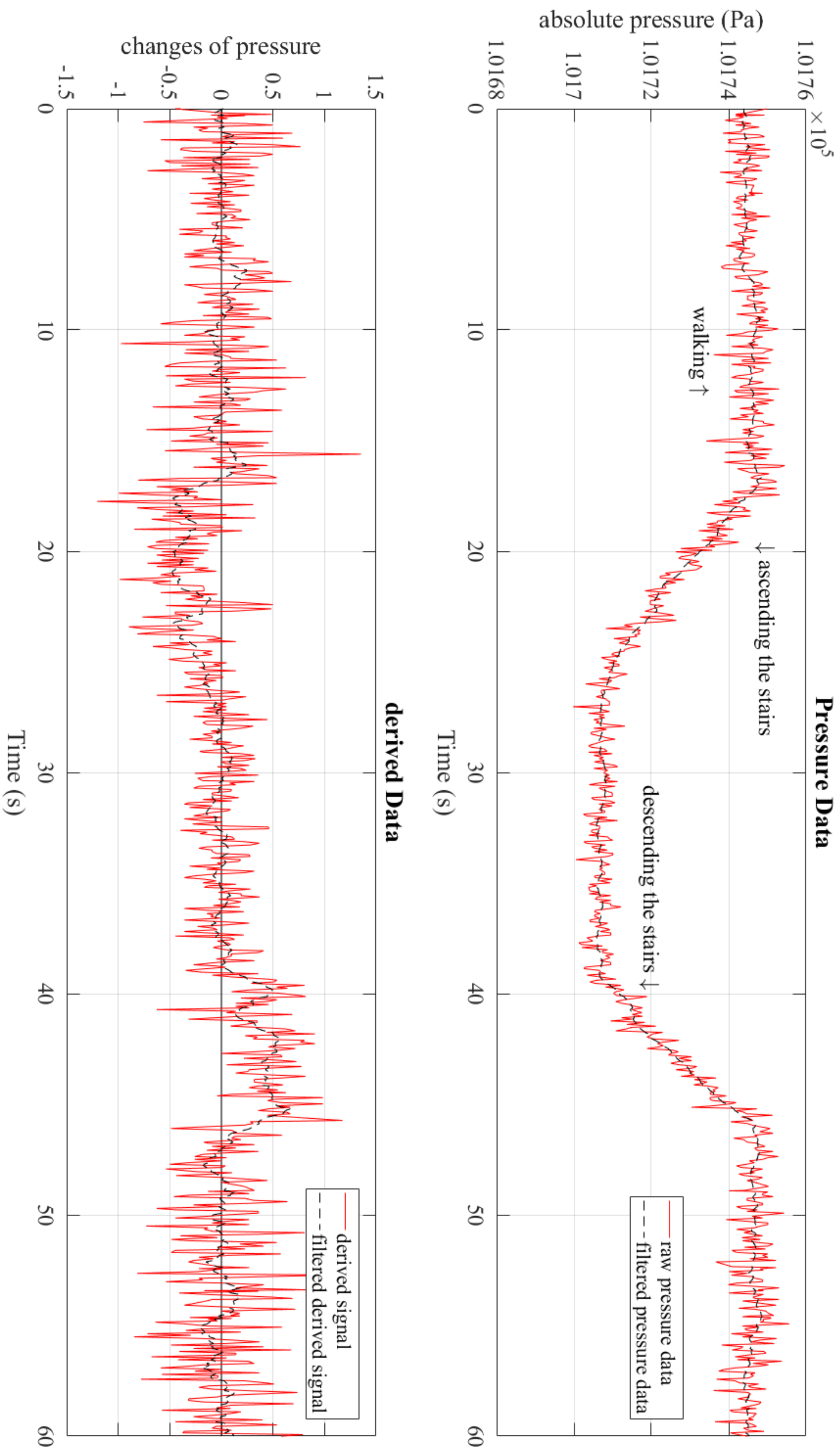

Figure 21: Recorded dataset for distinguishing stairs. The change in the pressure data is caused by climbing a stair.

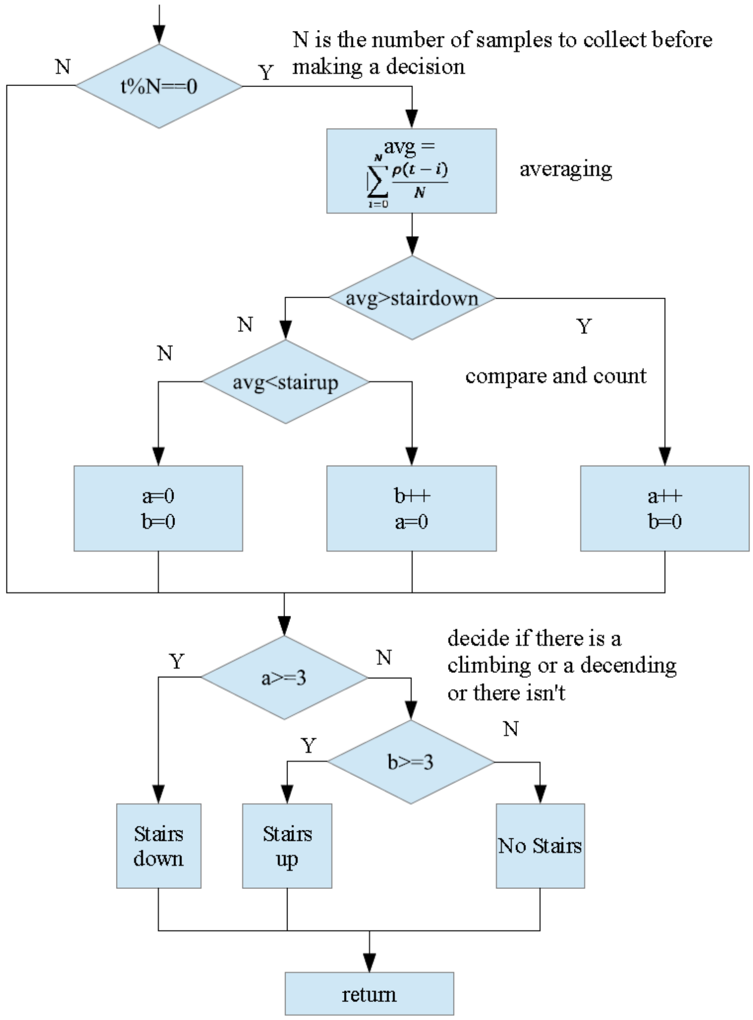

Figure 22: Flowchart detection of stairs, t is the actual sample #, N is the number of samples that are averaged. “a” is the counter for “Stairs up”. “b” is the counter for “Stairs down” stairup and stairdown are thresholds to determine if it there is enough change within a period

```
function data = stairdetection(data,cal,ii)
if mod(ii,cal.stairnum)==0;
    if mean(data.dpfilt(ii-cal.stairnum:ii))>cal.stairdown
        data.a= data.a+1;
        data.b=0;
    elseif mean(data.dpfilt(ii-cal.stairnum:ii))<cal.stairup
        data.b= data.b+1;
        data.a=0;
    else
        data.a=0;
        data.b=0;
    end
end
if data.a>=3% Define
    data.pstair(ii)=1; % Stairs down
elseif data.b>=3
    data.pstair(ii)=2; % Stair up
else
    data.pstair(ii)=0; % No Stair
end
end
```

Code 4: Implementation of Figure 22 in MATLAB, cal.stairnum is N, data.pstair is the output and ii is the actual sample. This code is executed with every new pressure data sample

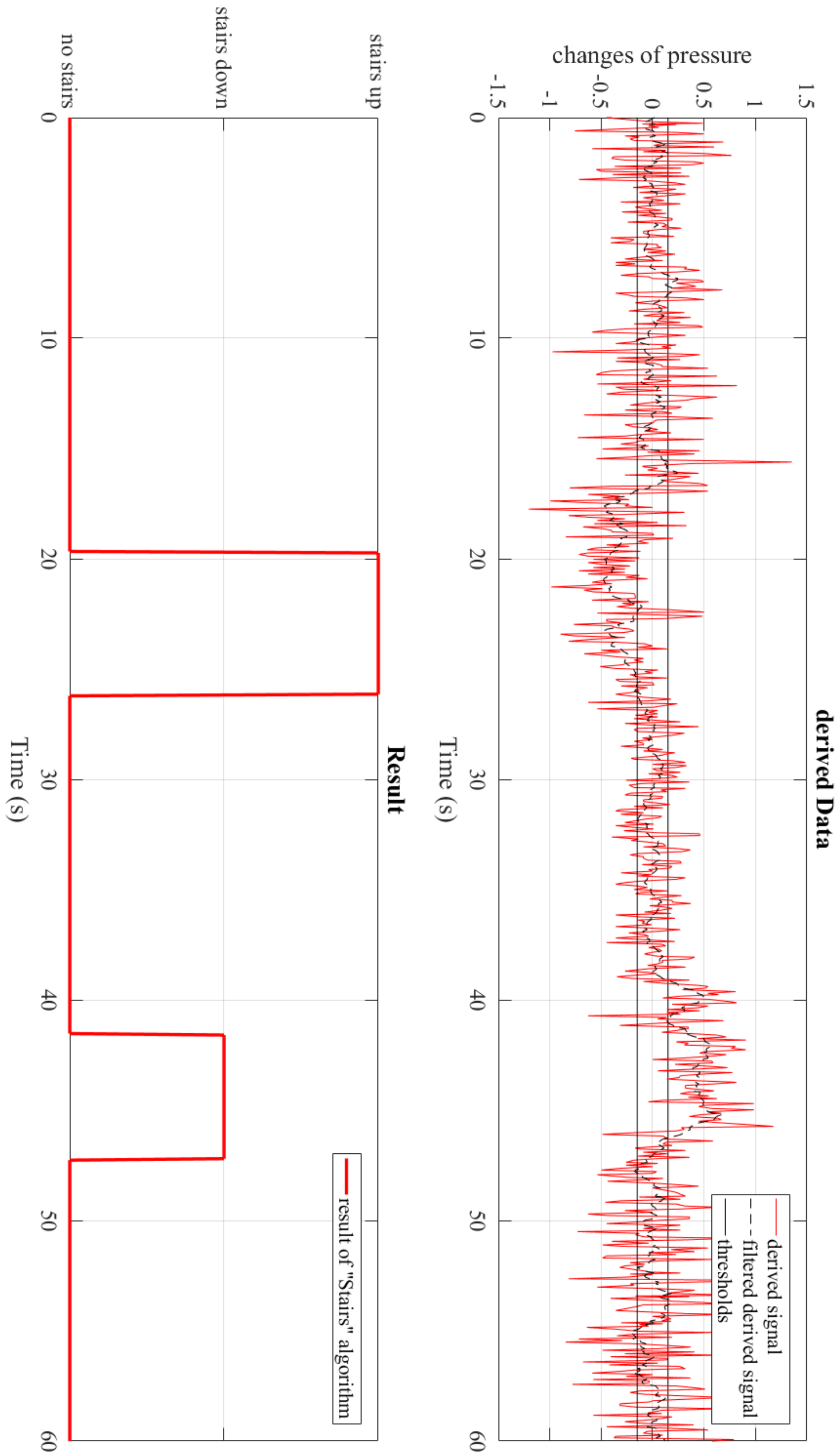

Figure 23: result of Code 4 on previously recorded dataset shown in Figure 21

3.5.3Biking

To enable the algorithm to detect the activity “Biking” it is necessary to examine the change of the angle of the upper leg. During “Biking” (sitting), the upper leg is performing a sinusoidal movement in a typical range of 38°, between 32° (hip extension) and 70° (hip flexion) (Figure 24) [37]. This range is changing between 20°-65° and 40°-80° depending on the position of the feet on the pedals and the position of the saddle [37] . The hip angle is not strongly affected by the workload (±3°) [37], [38]. It is assumed that the difference of the angle of the upper leg between “Sitting” and “Standing”, and therefore β, has a linear relationship with this angle, and the range of “Biking” is assumed in the field between 90% and 20% of β_{stand}.These two boundaries are used to enable the “Biking” detection (Figure 25). The angle must be within these boundaries for at least 1s to enable the algorithm. Furthermore, “Biking” is a moderate level activity. Therefore, the “Biking” detection is only enabled when there is at least this level of activity. This also means, that the LAL (3.3) must be calculated before calling the “Special Pattern Recognition”.

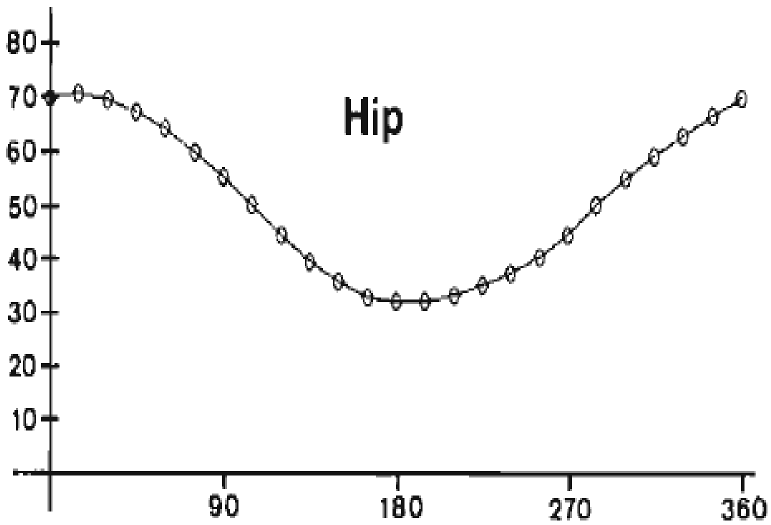

Figure 24: Typical hip joint angle motions in degrees during ergometer cycling (X-axis: Crank angle (°), Y-axis: Joint Angle (°))[37]

For this purpose, a dataset was recorded that included the states “Sitting”, “Standing”, and “Biking”. The sensor was placed in the left pocket of the test subject. An accelerometer sampling rate of 50Hz was chosen.

This data was imported into MATLAB (plotted in Figure 25).

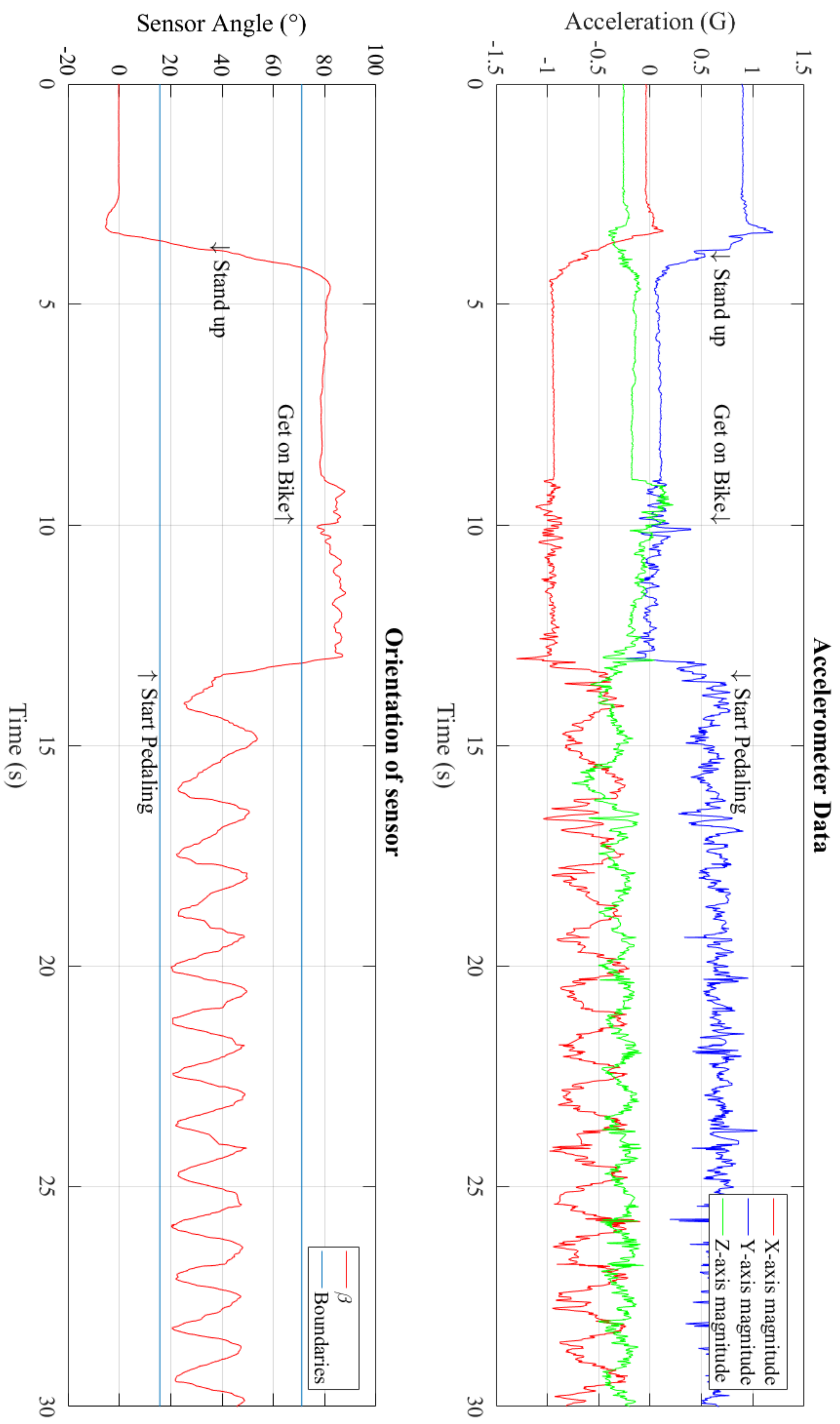

Figure 25: Dataset for distinguishing "Biking", Top, raw accelerometer data. Bottom, β and the boundaries for detecting "Biking"

3.5.3.1 Algorithm

Due to limited processing power in the sensor or the smartphone the analysis of the angle signal cannot currently be performed with a "Sine-Fit" or a Fast Fourier Transformation (FFT). Therefore, the algorithm is alternately detecting the lower and upper point of inflection of the signal. An additional boundary in form of a minimum change of degrees after detecting a turning point, is implemented to avoid unnecessary detections due to signal noise or peaks. This minimum change is, according to the previous results, around 15% of β_{stand}. It must be taken into consideration that fast transitions of the angle due to biking quickly will result in a lower change of β because of the low-pass filtering of the signal. After detecting an inflection point, the algorithm will search for the next opposite one. Due to noise in the signal, it is possible for the algorithm to detect a local maximum or minimum and erroneously classify it as the overall maximum. Due to alternating detection of inflection points, this could lead to an erroneous classification of a local maximum or minimum. To account for this, additional inflection points (local maximum or minimum) detected higher than the last top inflection point or lower than the last bottom inflection point are added and reclassified as the actual upper or lower inflection point.

Figure 26 shows an erroneously detected local maximum at 0.4s. After this point, the algorithm switches to searching for a local minimum which can only be found beneath the boundary. Whilst this, another local maximum (with a higher value) is detected (0.6s). Therefore, the boundary and the maxima are actualized. After detecting the minima (1.9s) the boundary is actualized, and the algorithm is searching for the next maximum above this boundary.

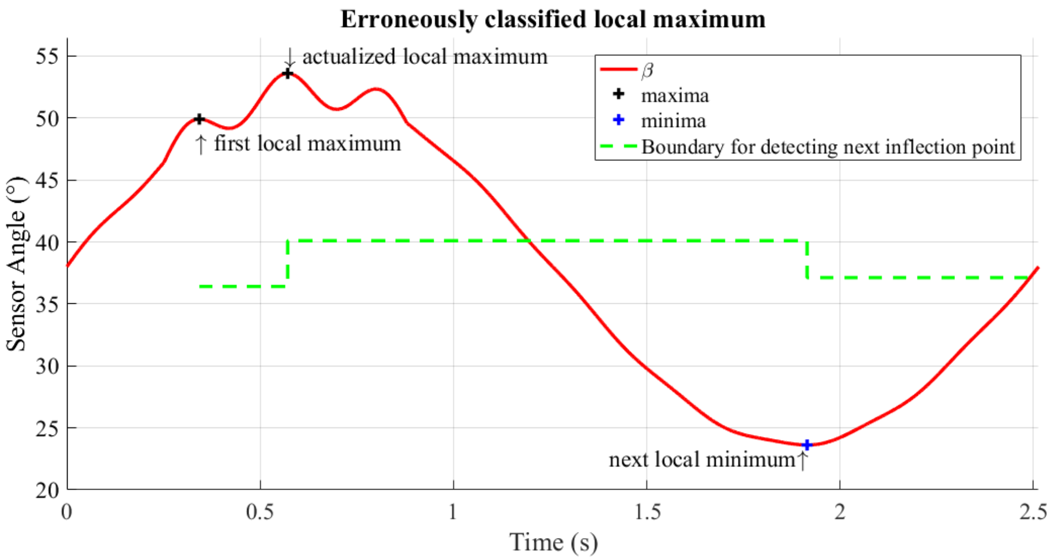

Figure 26: Erroneously classified local maximum and boundary for detecting next inflection point.

As previously described, the algorithm alternately detects a local maximum and minimum (Figure 27). These moments of switching between searching for a maximum or minimum is performed after the algorithm found what it searched for. After each detection (except the correction detections for erroneously classified inflection point) an additional function (runback) is called which is used to analyze the previously found maxima and minima (Figure 27).

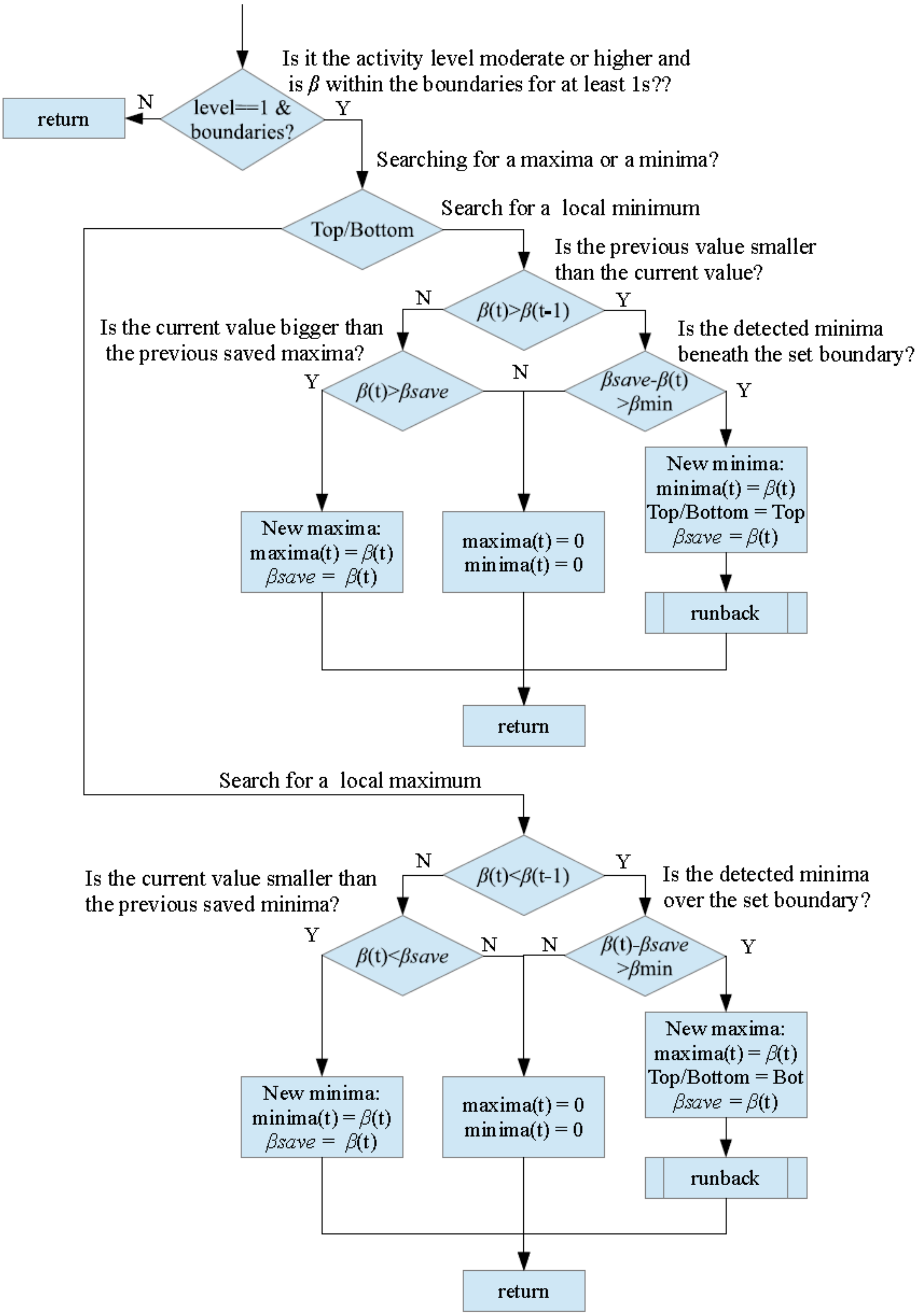

Figure 27: Flowchart local minima and maxima detection. This function is called at every new sample of the accelerometer

The “runback”-function (Figure 28, Code 5) is counting the previous minima and maxima that occurred in a defined period. It starts searching for the counterpart to the last detected inflection point. This makes sure that “runback” will only find the true minima and maxima. When there is at least an amount of a previously set threshold, detected inflection points it is detecting “Biking”. If this threshold set to (e.g.) three, it can detect biking after finishing the first round of the pedals (Figure

29). When "Biking" is detected the "runback"-function is rising a flag (bike) and saving the timestamp (biket) of this flag. These two variables will be used by the Combine Logic (3.6) for combining the outputs of the functions to a single conclusive perception.

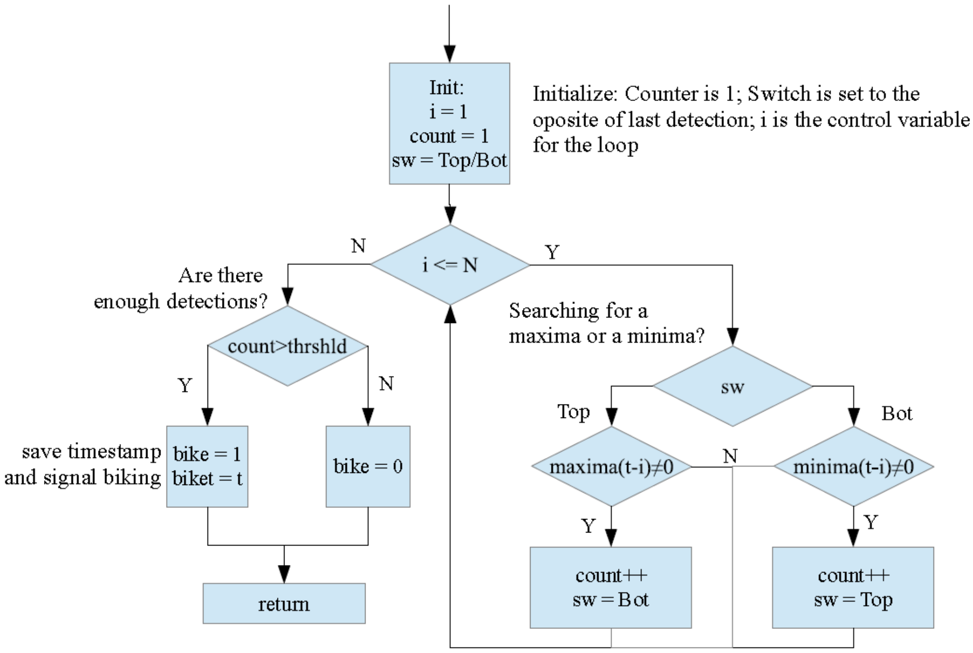

Figure 28: Flowchart for the "runback" function. N is the number of samples in the buffer. For 3 seconds of detection time N has to be 150 (50sps for 3seconds). thrshld is the minimum count of detected inflection points for detecting "Biking"

```
function data = bikerunback(data,cal,ii)
samples = 50*3; % Number of samples to search for inflection points
thrshld = 3; % Minimum found inflection points for signaling biking.
sw = data.topbot; % 0 = Last found was Bot, 1 = Last found was Top
num=1; % init counting variable

for jj = 1:samples
    if sw %Last was Top
        if data.minima(ii-jj)~=0 %Mark1 is Bot
            sw = 0;
            num = num+1;
        end
    else %last detection was Bot
        if data.maxima(ii-jj)~=0 %Mark2 is Top
            sw = 1;
            num = num+1;
        end
    end
end

if num>= thrshld % are there enough detections
    data.bike (ii) = 1; %Biking found
    data.biket = data.t(ii);
end
end
```
Code 5: Implementation of Figure 28 in MATLAB: This function is called for every new detection of an inflection point and gives a prediction whether the person is riding a bike based on the signal (data)

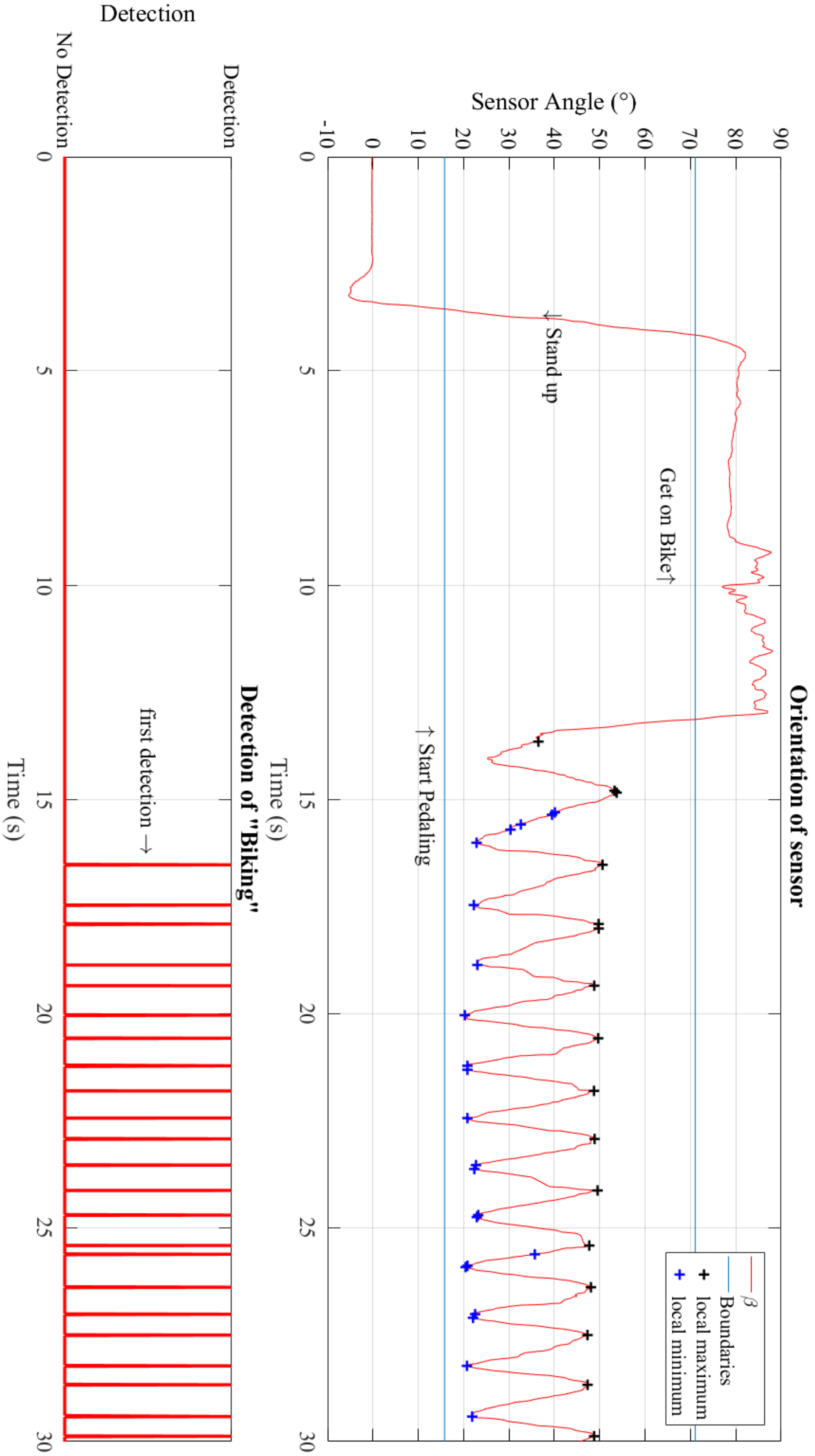

Figure 29: Detecting “Biking” on previously recorded dataset (Figure 25), Top, β and the boundaries for detecting “Biking” as well as detected local maxima and minima. Bottom, results of the “runback” function

3.5.4 Perching/Leaning

It is necessary to examine the difference in the angle of the upper leg to figure out if a person is "Leaning" or "Perching". Because of the minor difference in the angle "Leaning" is referred to as "Perching" in this Section. In Section 3.2.4, the coordinate system was rotated such that the angle β = 0 during "Sitting" and a defined angle (β_{stand}) during "Standing". Assuming that "Perching" is a state in between "Standing" and "Sitting", β is between 0 and β_{stand}. The angle in Figure 30 does not represent β, but instead defines the difference between these states. Furthermore, "Perching" and "Leaning" are light activities regarding the activity level detection (3.3). Detection of "Perching" is only activated if a Level 0 activity is detected (Code 6). The lower and the upper thresholds of "Perching" are set at 20% and 70% of β_{stand}, respectively. These values were chosen by observation of several subjects, and fit the expected values according to Figure 30.

There were several datasets recorded to test this detection, which included the states "Sitting", "Standing", and "Perching". One of them is represented in Figure 31.

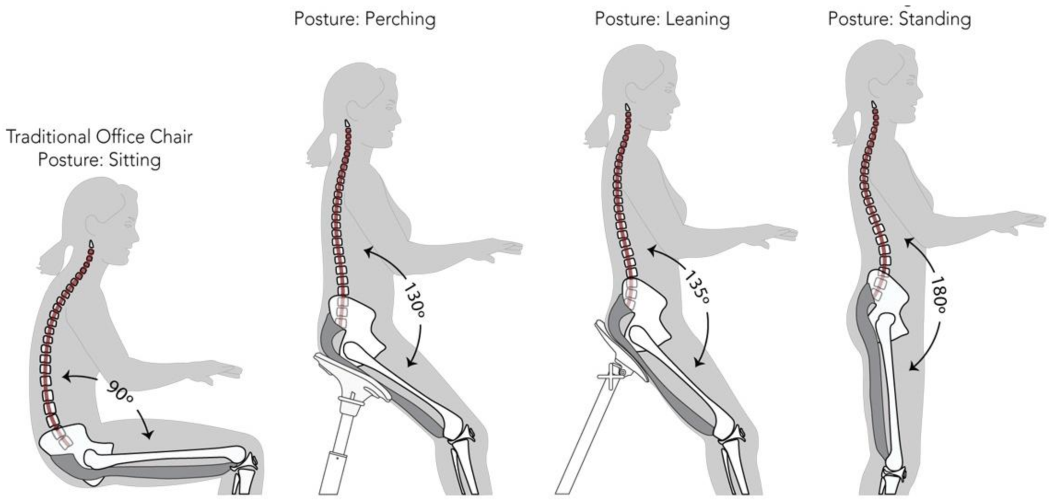

Figure 30: Typical different Sitting/Perching/Leaning positions during office work. (Source: Adapted from https://www.focalupright.com)

```
%% perching seat
if(data.level(ii)==0)
    if (data.tZ(ii)>cal.perchZbot)&&(data.tZ(ii)<=cal.perchZtop)
        data.perching(ii) = 1;
    else
        data.perching(ii) = 0;
    end
end
```

Code 6: Implementation of the perching detection as part of the Special Pattern Detection in MATLAB. If the activity level is light and β is within the upper (perchZtop) and lower (perchZbot) limits, "Perching" is detected. data.tZ is β

The result of using this code (Code 6) on recorded data is shown in Figure 31.

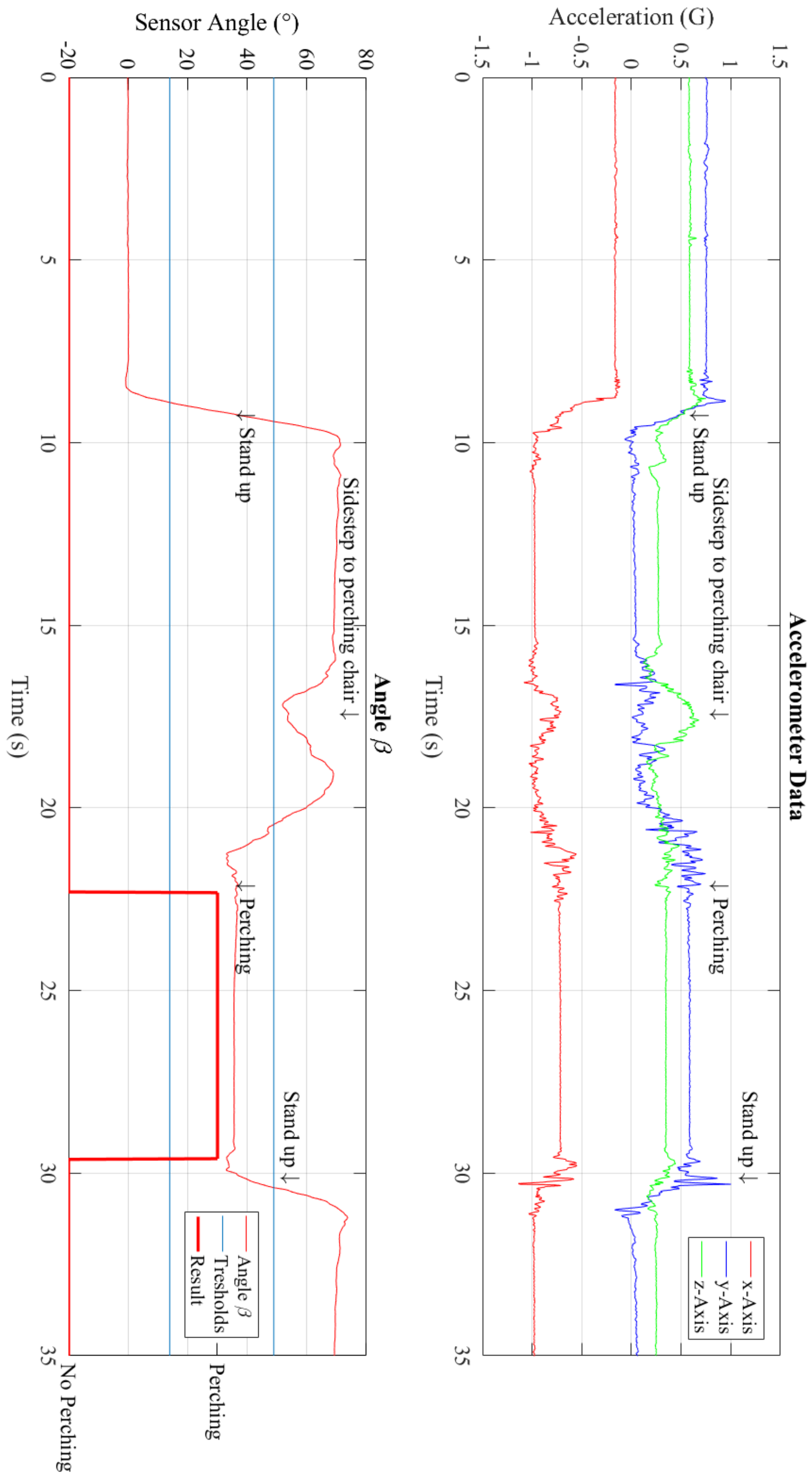

Figure 31: Dataset for distinguishing “Sitting”, “Standing”, and “Perching” “Leaning”. Top, raw accelerometer data. Bottom, β and the thresholds for detecting “Perching” and the result of Code 6.

3.6 Combine Logic

The Combine Logic is used to combine all the previous generated results in a single conclusive perception. So, the flags “Biking”, “Steps”, “Perching”, “Stairs”, “Standing/Sitting” and “Activity Level”(Table 11) are combined to generate different perceptions (Table 12) as shown in Figure 32.

Work stage	Calculated variable	Possible values
Sitting/Standing	standsit	0: Sitting 1: Standing
Special Pattern Recognition	stairs	0: No stairs 1: Stairs descending 2: Stairs ascending
	step	0: No step 1: Step
	laststep	Timestamp last detected “Step”
	bike	0: No biking 1: Biking
	biket	Timestamp last detection of “Biking”
	perching	0: no perching 1: perching
Level of Activity (LOA)	level	0: light/passive 1: moderate/active 2: vigorous/running 3: sprinting

Table 11: Combine Logic Inputs

At first the Combine Logic is used to divide the different activities based on their Level of Activity (Table 12). Due to this action the states “Running” and “Sprinting” are distinguished from each other and the states with “Light” and “Moderate” activities. An additional crosscheck with the value of the “Step Detection” is not necessary because there are no other activities who are associated with these LALs. Because of lack of movement, activities with a “Light” LAL are either “Perching”, “passive Sitting” or “passive Standing”. To distinguish these states, it the values of “Sitting/Standing” and “Perching” must be checked. “active Sitting”, “active Standing”, “Walking” “Stairs Up”, “Stairs Down” and Biking are activities with a “Moderate” LAL. Due to this diversity, it is necessary to start defining the current activity by checking the most certain flag. Due to the unique signal and method of the “Biking” – detection the “bike”-flag is considered the most certain flag and the one to start with. To set the state “Biking” only a single flag indicating a whole round of the pedals is required. After setting this state it will be maintained for 2.5s after the last detection of “Biking”. Due to similarities to “Biking” in the femur movement of smaller people and children while climbing stairs, it is necessary to check whether any kind of “Stairs” has been detected before setting the state “Biking”. The next signal to control is whether the person is “Sitting” or “Standing”. If the variable “standsit” is indicating the state “Sitting” the perception will be “active Sitting” while when it is indicating “Standing” there will be still four activities to differentiate. The next variable to check is if there are steps detected. If there is a flag “steps” or any kind of “Walking” has been detected within the last 2.5s, the outcome will be defined as “Walking”, “Stairs Up” or “Stairs Down” based on the state of the variable “stairs”. If there are no steps the perception will be “active Sitting”. This logic is shown in Figure 32.

Level of Activity	Different outputs
0: Light	“Perching”
	“passive Sitting”
	“passive Standing”
1: Moderate	“active Sitting”
	“active Standing”
	“Walking”
	“Stairs Up”
	“Stairs Down”
	“Biking”
2: Vigorous	“Running”
3: Sprinting	“Sprinting”

Table 12: Combine Logic Outputs sorted by activity level

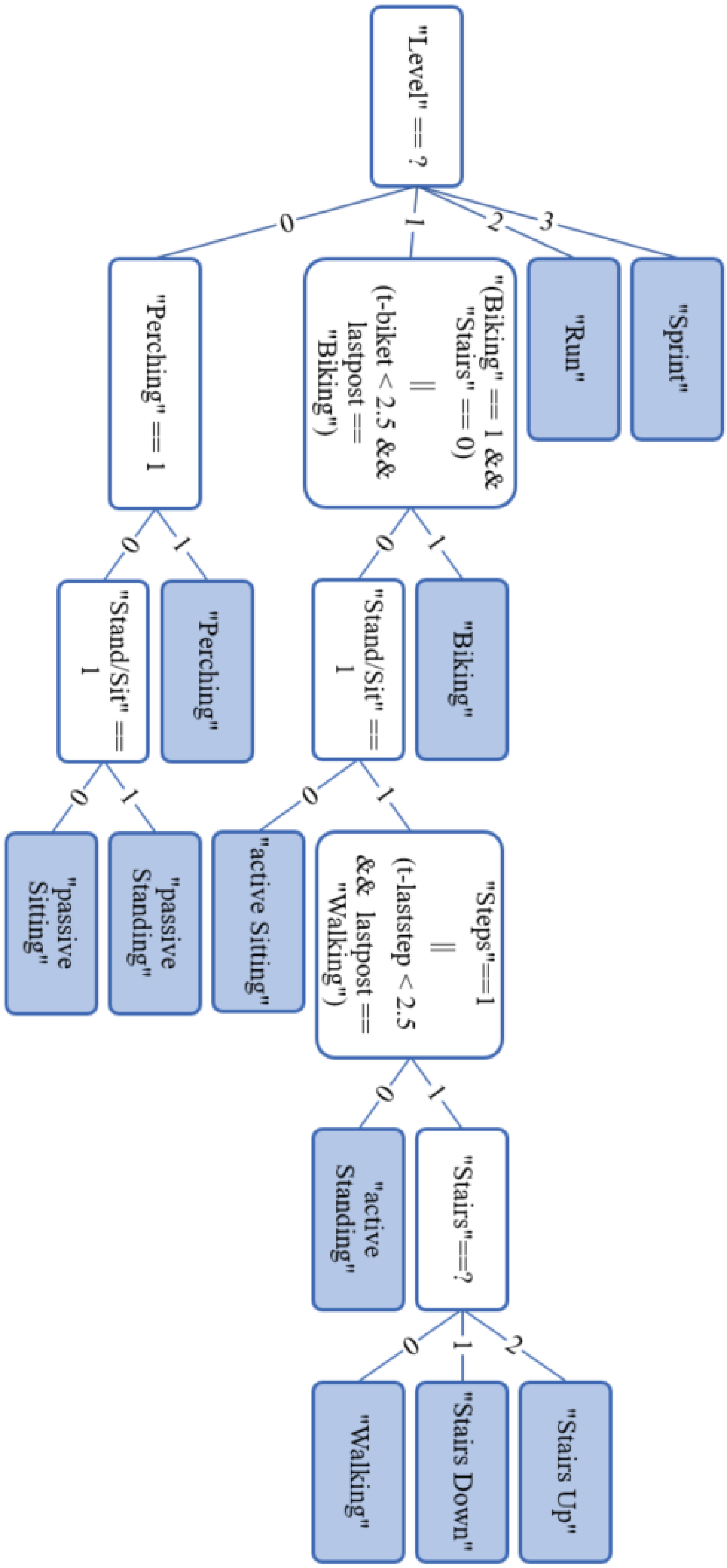

Figure 32: principal construction of the "Combine Logic", postures in the blue boxes are perceptions based on the inputs and questions of the white boxes. This logic is called with every new sample of the accelerometer or pressure sensor

4 Validation Study

The validation study is used to calculate the algorithms accuracy on two different positions of the sensor on the body (belt and pocket). While the development of the algorithm was based on data of a sensor in a pocket, it is also necessary to validate the algorithm on a position on the belt to see how a change in position affects its accuracy. The algorithm is designed for everyday use. Since there are many ways to wear pants (Figure 33 and Figure 34), the change in orientation of the sensor can vary and is different for each case.

To validate the algorithm, a dataset of 18 different participants was recorded. Each participant was wearing a sensor on the belt and on sensor in the pocket. The way in which the subject was wearing their pants was recorded.

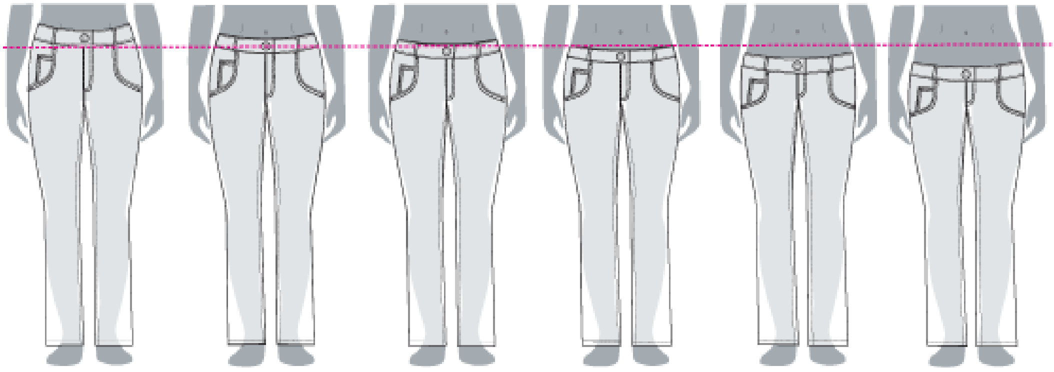

Figure 33: different ways in which pants can be worn (adapted from joyofclothes.com)

Figure 34: Different positions of sensors on different participants. The sensor is visibly lower on the hip on the left participant than on the right.

4.1 Methods

The videos used for this study were recorded using a Nikon Coolpix L820 at a resolution of 1920x1080 pixel and a framerate of 29.97Hz. The videos were analysed using Multimedia Video Task Analysis™ (MVTA) Software (Figure 35) of Ergonomics Analysis and Design Research Consortium of the University of Wisconsin-Madison. This analysis was performed on each recorded video by two independent students to increase validity. For each part of the algorithm there was an equivalent part in the analysis, to enable the comparison of each output to a golden standard based on that analysis. The MVTA output was a file (*.MDF3) which was imported into MATLAB for the comparison with the recorded dataset. The posture used for comparison was calculated in MATLAB and interpolated on the timestamps of the accelerometer to close the gap between the two different sample rates. The correctness of the analysis was controlled by superposing the result on the videos (Figure 36). The participants of the study physically tapped each sensor at the beginning and the end of each recording. This impact was manually identified and marked in the accelerometer data and in the video file to synchronise the video to the recorded accelerometer data and pressure data.

Figure 35: MVTA Analysis interface

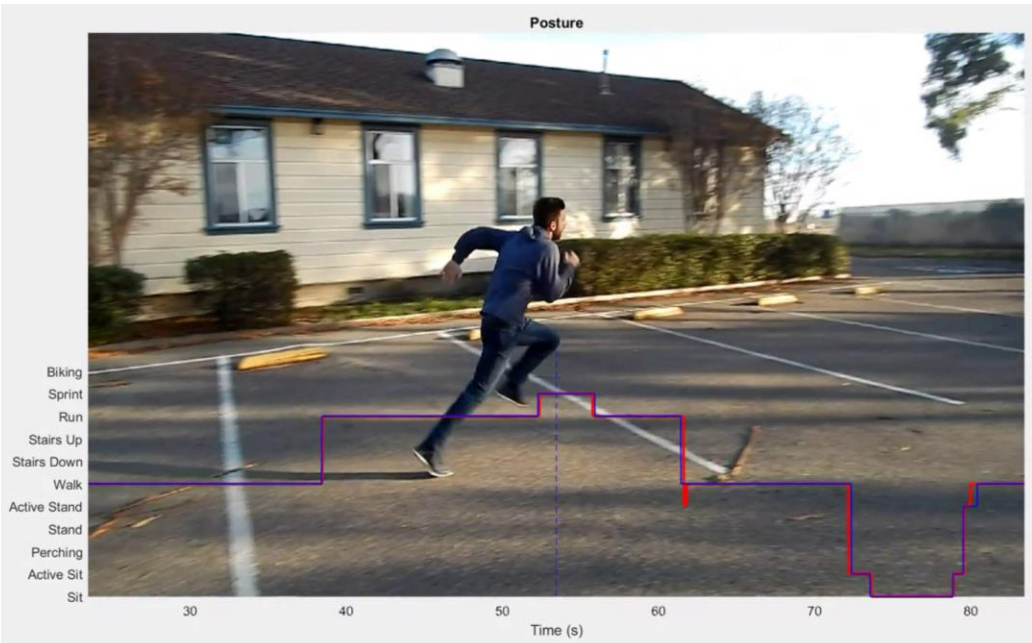

Figure 36: Analysis data overlay on video

4.2 Subjects

In this study, 18 subjects (7 females, 11 males, age range 23 - 51 years; Table 13) served as paid voluntary participants. The participants were told that the study would investigate various measures to assess physical activity and posture. Informed consent was obtained, and this study adhered to the tenets of the declaration of Helsinki.

Nr.:	Gender	Age	Height (cm)	Weight (kg)	Additional Information
1	F	24	158	53	
2	M	23	183	75	
3+	F	26	175	68	high waisted jeans
4	M	27	195	95	No Watch step count
5	M	26	186	84	No Watch step count
6	M	28	178	85	No Watch step count
7+	F	29	177	70	high waisted jeans
8-	M	33	180	60	Strong wind outside / very low pants
9+	F	23	162	52	No Biking / high waisted jeans
10	M	26	180	84	
11	M	23	175	75	
12	M	23	194	84	
13+	F	26	178	69	high waisted jeans
14	M	38	186	106	
15	M	26	183	86	
16	M	51	195	111	
17	F	27	172	81	
18+	F	44	170	61	high waisted jeans

Table 13: Data of Participants. + and – indicate the subject was wearing high- and low-waisted pants, respectively.

4.3 Duration of Activities

Table 14 shows the durations for each activity or posture of each participant. This data is based on the video analysis. The videos of the participants combined are two hours and 37 minutes long. In this time, the participants spent 22.3% sitting, 10.2% standing, 5.2% perching, 39.5% walking, 4.0% running and 15.2% biking. The remaining 3.7% could not be defined as any of the activities or postures.

Nr.:	Sitting active	Sitting passive	Perching	Standing passive	Standing active	Walking	Stairs UP	Stairs Down	Running	Sprint	Biking	Not defined	Duration
1	22,6	100,9	42,0	49,5	20,2	171,7	13,2	9,9	16,2	7,0	86,4	21,0	560,5
2	15,3	137,3	32,1	43,9	21,2	179,5	12,5	9,4	15,9	3,1	71,7	23,5	565,5
3+	13,7	93,6	25,4	34,5	24,3	158,8	11,4	8,5	14,6	3,1	92,9	15,6	496,3
4	26,2	78,8	19,0	37,0	14,2	153,4	11,8	8,7	18,6	0,0	77,7	27,0	472,3
5	10,9	92,5	21,3	32,9	11,0	179,5	11,8	10,6	19,4	3,6	78,1	26,6	498,3
6	18,5	121,5	27,5	40,6	14,1	169,7	12,0	8,5	15,4	0,0	70,2	28,6	526,8
7+	13,9	120,3	37,6	46,8	17,6	197,9	13,5	10,8	27,9	0,0	87,1	32,4	605,7
8-	7,2	86,4	19,8	28,6	21,4	197,4	14,1	10,0	21,1	0,0	88,7	29,3	524,0
9+	11,8	80,4	18,2	32,2	11,5	181,3	11,0	8,4	17,4	0,0	0,0	0,1	372,3
10	11,3	84,1	23,5	36,0	13,0	169,9	10,7	7,4	18,4	2,6	91,6	6,7	475,1
11	10,5	107,6	31,7	36,8	15,4	202,6	13,6	9,6	17,6	0,0	88,6	5,1	539,2
12	13,7	89,6	22,4	41,6	14,2	189,2	10,8	5,6	19,9	3,4	109,3	20,4	539,9
13+	13,1	105,3	26,9	33,1	17,3	199,0	12,8	8,9	12,7	0,0	79,4	24,7	533,1
14	13,7	118,0	34,1	27,8	13,4	199,3	9,2	8,0	21,5	0,0	73,7	22,4	541,0
15	11,5	127,8	28,2	41,7	11,1	188,7	11,7	10,2	20,1	3,4	95,8	6,5	556,7
16	12,4	112,9	32,8	32,6	13,7	187,7	12,0	10,8	20,5	3,0	98,4	5,9	542,6
17	17,4	95,1	29,1	36,1	22,1	227,7	12,9	11,6	23,9	0,0	89,7	17,8	583,4
18+	13,1	101,7	23,2	44,0	14,5	206,9	12,8	9,4	24,6	0,0	59,5	35,9	545,5
Sum	**256,8**	**1853,6**	**494,7**	**675,7**	**290,1**	**3360,2**	**217,8**	**166,3**	**345,6**	**29,2**	**1438,7**	**349,4**	**9478,0**
%	**2,7%**	**19,6%**	**5,2%**	**7,1%**	**3,1%**	**35,5%**	**2,3%**	**1,8%**	**3,6%**	**0,3%**	**15,2%**	**3,7%**	**100%**

Table 14: Overall duration of activities in seconds based on the video analysis

5 Results

At the end of my research stay in Berkeley, CA the results of the study were not analyzed. This project is being continued in my 6th semester at the University of Applied Science Upper Austria and will be finished in my Bachelor-Thesis in July 2018.

I want to thank the Austrian Marshall Plan Foundation for their support. Without it, it would not have been possible for me to take part in that research program.

References

[1] N. C. Barengo, R. Antikainen, K. Borodulin, K. Harald, und P. Jousilahti, „Leisure-Time Physical Activity Reduces Total and Cardiovascular Mortality and Cardiovascular Disease Incidence in Older Adults“, *J. Am. Geriatr. Soc.*, Bd. 65, Nr. 3, S. 504–510, März 2017.

[2] J. N. Morris, J. A. Heady, P. A. B. Raffle, C. G. Roberts, und J. W. Parks, „CORONARY HEART-DISEASE AND PHYSICAL ACTIVITY OF WORK“, *The Lancet*, Bd. 262, Nr. 6796, S. 1111–1120, Nov. 1953.

[3] K. M. Herman, C. L. Craig, L. Gauvin, und P. T. Katzmarzyk, „Tracking of obesity and physical activity from childhood to adulthood: The Physical Activity Longitudinal Study“, *Int. J. Pediatr. Obes.*, Bd. 4, Nr. 4, S. 281–288, Jan. 2009.

[4] P. Bellavista, Computer Society, und Communications Society, Hrsg., *Proceedings / 11th IEEE Symposium on Computers and Communications, ISCC 2006: 26 - 29 June 2006, Cagliari, Sardinia, Italy*. Los Alamitos, Calif.: IEEE Computer Society, 2006.

[5] M. Quwaider, A. Plummer Jr, J. Rao, M. Taghizadeh, und S. Biswas, „Real-time posture detection using body area sensor networks“, 2017.

[6] Rednic, R., Kemp, J., Brusey, J., und Gaura, E., „Posture determination using a body sensor network“, März 2012.

[7] F. Foerster, M. Smeja, und J. Fahrenberg, „Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring“, *Comput. Hum. Behav.*, Bd. 15, Nr. 5, S. 571–583, Sep. 1999.

[8] M. Kirwan, M. J. Duncan, C. Vandelanotte, und W. K. Mummery, „Using Smartphone Technology to Monitor Physical Activity in the 10,000 Steps Program: A Matched Case–Control Trial“, *J. Med. Internet Res.*, Bd. 14, Nr. 2, S. e55, Apr. 2012.

[9] MBIENTLAB INC, „MetaMotionR Wearable and connected devices product solution Product Specification v0.5“. 21-Nov-2016.

[10] B. Boulay, *Human Posture Recognition for Behaviour Understanding*. .

[11] H. Foroughi, B. S. Aski, und H. Pourreza, „Intelligent video surveillance for monitoring fall detection of elderly in home environments“, 2008, S. 219–224.

[12] I. Haritaoglu, D. Harwood, und L. S. Davis, „W/sup 4/: real-time surveillance of people and their activities“, *IEEE Trans. Pattern Anal. Mach. Intell.*, Bd. 22, Nr. 8, S. 809–830, Aug. 2000.

[13] S. Pellegrini und L. Iocchi, „Human Posture Tracking and Classification through Stereo Vision and 3D Model Matching“, *EURASIP J. Image Video Process.*, Bd. 2008, S. 1–12, 2008.

[14] L. Schwarz, D. Mateus, V. Castaneda, und N. Navab, „Manifold Learning for ToF-based Human Body Tracking and Activity Recognition“, 2010, S. 80.1-80.11.

[15] G. M. Lyons, K. M. Culhane, D. Hilton, P. A. Grace, und D. Lyons, „A description of an accelerometer-based mobility monitoring technique“, *Med. Eng. Phys.*, Bd. 27, Nr. 6, S. 497–504, Juli 2005.

[16] M. Quwaider und S. Biswas, „Body Posture Identification using Hidden Markov Model with a Wearable Sensor Network“. .

[17] C. G. Ryan, P. M. Grant, W. W. Tigbe, und M. H. Granat, „The validity and reliability of a novel activity monitor as a measure of walking“, *Br. J. Sports Med.*, Bd. 40, Nr. 9, S. 779, Sep. 2006.

[18] A. Godfrey, K. M. Culhane, und G. M. Lyons, „Comparison of the

performance of the activPAL^{TM} Professional physical activity logger to a discrete accelerometer-based activity monitor“, *Med. Eng. Phys.*, Bd. 29, Nr. 8, S. 930–934, Okt. 2007.

[19] A. Hecht, S. Ma, J. Porszasz, R. Casaburi, und for the COPD Clinical Research Netw, „Methodology for Using Long-Term Accelerometry Monitoring to Describe Daily Activity Patterns in COPD“, *COPD J. Chronic Obstr. Pulm. Dis.*, Bd. 6, Nr. 2, S. 121–129, Jan. 2009.

[20] D. Sumukadas, S. Laidlaw, und M. D. Witham, „Using the RT3 accelerometer to measure everyday activity in functionally impaired older people“, *Aging Clin. Exp. Res.*, Bd. 20, Nr. 1, S. 15–18, Feb. 2008.

[21] B. G Steele, B. Belza, K. Cain, C. Warms, J. Jeff Coppersmith, und J. Howard, *Bodies in motion: Monitoring daily activity and exercise with motion sensors in people with chronic pulmonary disease*, Bd. 40. 2003.

[22] N. A. Maffiuletti *u. a.*, „Concurrent validity and intrasession reliability of the IDEEA accelerometry system for the quantification of spatiotemporal gait parameters“, *Gait Posture*, Bd. 27, Nr. 1, S. 160–163, Jan. 2008.

[23] M. Grydeland, B. H. Hansen, M. Ried-Larsen, E. Kolle, und S. A. Anderssen, „Comparison of three generations of ActiGraph activity monitors under free-living conditions: do they provide comparable assessments of overall physical activity in 9-year old children?“, *BMC Sports Sci. Med. Rehabil.*, Bd. 6, Nr. 1, S. 26, Juni 2014.

[24] M. J. Mathie, B. G. Celler, N. H. Lovell, und A. C. F. Coster, „Classification of basic daily movements using a triaxial accelerometer“, *Med. Biol. Eng. Comput.*, Bd. 42, Nr. 5, S. 679–687, Sep. 2004.

[25] S. L. Murphy, „Review of physical activity measurement using accelerometers in older adults: Considerations for research design and conduct“, *Prev. Med.*, Bd. 48, Nr. 2, S. 108–114, Feb. 2009.

[26] C.-C. Yang und Y.-L. Hsu, „A Review of Accelerometry-Based Wearable Motion Detectors for Physical Activity Monitoring“, *Sensors*, Bd. 10, Nr. 8, S. 7772–7788, Aug. 2010.

[27] W. Schirotzek und S. Scholz, *Starthilfe Mathematik*. Wiesbaden: Vieweg+Teubner Verlag, 2005.

[28] K. Giokas, L. Bokor, und F. Hopfgartner, Hrsg., *eHealth 360°*, Bd. 181. Cham: Springer International Publishing, 2017.

[29] Han-Wook Lee, Ju-Won Lee, Won-Geun Jung, and Gun-Ki Lee*, „International Journal of Control, Automation, and Systems, vol. 5, no. 6, pp.701-706, 2007“, in *International Journal of Control, Automation, and Systems*, .

[30] G. Cai, B. M. Chen, und T. H. Lee, „Coordinate Systems and Transformations“, in *Unmanned Rotorcraft Systems*, London: Springer London, 2011, S. 23–34.

[31] W. Burger und M. J. Burge, *Principles of Digital Image Processing*. London: Springer London, 2009.

[32] The American Heart Association, „Moderate to Vigorous - What is your level of intensity?“, *The American Heart Association*. .

[33] The Nutrition Source, „Examples of Moderate and Vigorous Physical Activity“, *The Nutrition Source -Harvard University*, 23-Okt-2017. .

[34] D. M. Harrington, G. J. Welk, und A. E. Donnelly, „Validation of MET estimates and step measurement using the ActivPAL physical activity logger“, *J. Sports Sci.*, Bd. 29, Nr. 6, S. 627–633, März 2011.

[35] G. L. Smidt, Hrsg., *Gait in rehabilitation*. New York: Churchill Livingstone,

1990.

[36] J. Eichler, *Physik: für das Ingenieurstudium - prägnant mit knapp 300 Beispielaufgaben ; mit 48 Tabellen*, 4., Aktualisierte und überarbeitet Auflage. Wiesbaden: Vieweg + Teubner, 2011.

[37] M. O. Ericson und R. Nisell, „Joint Motions of the Lower Limb during Ergometer Cycling“. American Physical Therapy Association, 1988.

[38] Roman Farana, Adam Motykal, Jaroslav Uchytill, und Gerda Strutzenberge, „EFFECT OF DIFFERENT CYCLING CONDITIONS ON SAGITALL AND FRONTAL LOWER LIMB KINEMATICS OF COMPETITIVE AND RECREATIONAL CYCLISTS“. Human Motion Diagnostic Center, University of Ostrava, Czech Republic Department of Sport Science and Kinesiology, University of Salzburg, Austria, 18.72016.