

# Supplementary file 1 - posture and mobility detection algorithm

Fabian Marcel Rast<sup>a,b,c</sup>

<sup>a</sup>Swiss Children's Rehab, University Children's Hospital Zurich, Mühlebergstrasse 104, 8910, Affoltern am Albis, Switzerland

<sup>b</sup>Children's Research Center, University Children's Hospital of Zurich, University of Zurich, Zurich, Switzerland

<sup>c</sup>Rehabilitation Engineering Laboratory, Department of Health Sciences and Technology, ETH Zurich, Zurich, Switzerland

---

## Abstract

The algorithm can be divided into three independent parts using different sensor setups:

1. The posture detection algorithm detects lying, sitting, and standing positions based on data of the trunk and thigh sensors.
2. The wheeling detection algorithm detects wheeling periods with data of the wheelchair sensor and discriminates between active and passive wheeling with data of the wrist sensor of the dominant hand.
3. The walking detection algorithm detects walking periods and differentiates between level walking and stair climbing with data of a single ankle sensor. Further, the algorithm discriminates between free and assisted walking with data of the sensor attached to a walking aid.

---

## 1. Raw data

We used the data of inertial measurement units containing a 3-axis accelerometer, a 3-axis gyroscope, and a barometric pressure sensor, as well as Bluetooth Low Energy for time synchronization (see Figure 1).[1] However, the algorithm can be applied to any measurement unit containing the required sensor modalities. The raw data needs to be resampled to 50 Hz, and the signals must be measured in or converted to the following units:

- acceleration  $a \Rightarrow m/s^2$
- angular rate  $\omega \Rightarrow \text{°}/s$
- barometric pressure  $p \Rightarrow Pa$

## 2. Posture detection algorithm

This part of the algorithm detects lying, sitting, and standing positions based on data of the trunk and thigh sensors.

---

Email address: [fabian.rast@gmail.com](mailto:fabian.rast@gmail.com) (Fabian Marcel Rast)



Figure 1: The ZurichMOVE sensor and its coordinate system (created by Rehabilitation Engineering Laboratory, ETH Zurich).

13 *2.1. Sensor placement*

14 The trunk sensor needs to be placed on the sternum with the x-axis facing toward the belly  
 15 button. The thigh sensor needs to be placed mid-thigh on the lateral side of the less-affected leg.  
 16 Here, the x-axis faces toward the knee (see Figure 2).

17 *2.2. Orientation estimation*

18 Before estimating the orientation of the sensor, the algorithm corrects the offset and drift of the  
 19 gyroscope signal.[2] First, still phases are detected by applying a 2nd order high-pass filter (cut-off  
 20 frequency =  $0.5\text{ Hz}$ ), a low-pass filter (cut-off frequency =  $2\text{ Hz}$ ), and a threshold of  $1^\circ/s$ . [3] Then,  
 21 the drift of the gyroscope signal is estimated by piecewise low-pass filtering of each axis, linearly  
 22 interpolating between the still phases, and limiting the slew rate of the signal to  $500\ \mu^\circ/s$ . And  
 23 finally, this drift is subtracted from the raw gyroscope measurements.

24  
 25 To estimate the orientation of the sensor, the acceleration and the corrected gyroscope signals  
 26 are fused with the open-source algorithm of S. Madgwick.[4] The filter gain  $\beta$  was set to 0.03 which  
 27 provided optimal performance in previous experiments.[4]. The output is a vector containing the  
 28 quaternions of each sample  $\vec{q} = [q_0\ q_1\ q_2\ q_3]^T$ . In the neutral position ( $\vec{q} = [1\ 0\ 0\ 0]^T$ ), the z-axis  
 29 points towards the floor.

30  
 As the last step, the pitch angle of the sensor's orientation is derived from the quaternions. The  
 pitch angle is defined as the deviation of the sensor's orientation from its neutral position around a  
 new y-axis after rotating the sensor around its z-axis. It is calculated with the following equation:

$$\varphi = \arcsin\left(2(q_0q_2 - q_3q_1)\right) * \frac{180}{\pi}$$

31 This angle is filtered using a 5th order low-pass filter (cut-off frequency =  $0.1\text{ Hz}$ ). An angle of  $0^\circ$   
 32 represents a horizontal orientation, while an angle of  $\pm 90^\circ$  represents a vertical orientation. Nega-  
 33 tive values result when the x-axis of the sensor points downward, and positive values result when  
 34 the x-axis points upward. The former corresponds to a standing position, while the latter would  
 35 correspond to a handstand position. Signals of which the mean of the whole measurement period  
 36 exceeds  $0^\circ$  are multiplied by -1 since we assume that the sensor was placed upside down rather than

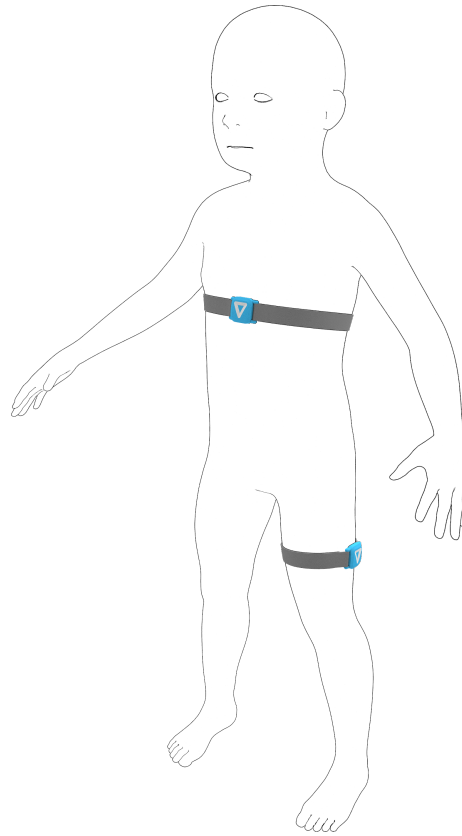


Figure 2: Sensor placement of the posture detection algorithm (created by Rehabilitation Engineering Laboratory, ETH Zurich).

37 the participant being in this position for a prolonged time.

38

### 39 *2.3. Classification of lying, sitting, and standing*

40 In lying, both sensors are horizontal while they are vertical during standing. In sitting, however,  
 41 the thigh sensor is horizontal and the trunk sensor is vertical. A vertical thigh sensor and a hori-  
 42 zontal trunk sensor is uncommon and probably reflects a standing position with bending forward.  
 43 Hence, the algorithm classifies this scenario as standing. The thresholds to distinguish between a  
 44 horizontal and vertical orientation were trained with labeled data of children with mobility impair-  
 45 ments and a decision tree by minimizing the Gini's Diversity Index. The resulting thresholds are  
 46  $T_{trunk} = -35.9^\circ$  and  $T_{thigh} = -48.4^\circ$ .

### 47 *2.4. Outcome measures*

48 After detecting lying, sitting, and standing positions, the algorithm determines the duration the  
 49 participant spent in each position throughout the measurement period. Moreover, the number of  
 50 transitions between a sitting and a standing position are counted. The minimal duration between

51 two consecutive sit-to-stand transitions was set to 2 *min* to avoid an overestimation in noisy data  
52 or during cycling periods.

### 53 **3. Wheeling detection algorithm**

54 This part of the algorithm detects wheeling periods with data of the wheelchair sensor and  
55 discriminates between active and passive wheeling with data of the wrist sensor of the dominant  
56 hand.

#### 57 *3.1. Sensor placement*

58 The wheelchair sensor needs to be placed on the spokes of the wheelchair, with the z-axis being  
59 parallel to the axis of the wheel. The direction of the z-axis does not matter since the algorithm  
60 assumes that the participant more frequently wheels forward than backward. The wrist sensor is  
61 worn on the dominant hand as a watch. The x-axis faces toward the fingers (see Figure 3). The  
62 user selects the dominant hand by placing a single sensor on the corresponding wrist. If data of  
63 both wrist sensors are available, the algorithm uses the side which reveals a higher acceleration  
magnitude during wheeling periods.



Figure 3: Sensor placement of the wheeling detection algorithm (created by Rehabilitation Engineering Laboratory, ETH Zurich).

64

#### 65 *3.2. Classification of non-wheeling activities, active wheeling and passive wheeling*

66 This part of the algorithm is an adapted version of a previously published algorithm that was  
67 developed for patients with a spinal cord injury.[5] The feature selection process was repeated with  
68 data of children with mobility impairments and the resulting algorithm is described in the following  
69 sections.

70 *3.2.1. Detection of wheeling periods*

71 This part depends solely on the z-axis of the wheelchair sensor  $a_{wheel,z}$  and  $\omega_{wheel,z}$ . As a first  
72 step, it is verified if the z-axis of the sensor is in a horizontal orientation which is the case if the  
73 sensor is fixed to the spokes of the wheel. In this case, the acceleration signal due to gravity is close  
74 to zero. In contrast, the signal is close to  $9.81 \text{ m/s}^2$  if the sensor is lying around in neutral position.  
75 Therefore, periods in which  $a_{wheel,z}$  is  $> 0.5 * 9.81 \text{ m/s}^2$  for longer than  $1 \text{ min}$  are classified as  
76 non-wheeling activities and ignored during the following steps. Before applying this cut-off, the  
77 signal is processed with a low-pass filter (cut-off frequency =  $0.05 \text{ Hz}$ ).

78  
79 Then, plateaus in the gyroscope signal of five samples in a row or longer are set to zero to  
80 remove gyroscope data with low quality. The resulting signal is used to detect wheeling periods in  
81 three steps: identifying preliminary wheeling periods, classifying them as valid and invalid wheeling  
82 periods, and fusing valid wheeling periods by analyzing the rest phases between two consecutive  
83 wheeling periods. First, a threshold ( $|\omega_{wheel,z}| > 0.4^\circ/s$ ) is applied to identify preliminary wheeling  
84 periods. Second, for each period, the following heuristic rules are used to detect valid wheeling  
85 periods:

- 86 •  $\max |\omega_{wheel,z}| > 10^\circ/s$
- 87 •  $Var(\omega_{wheel,z}) > 1^\circ/s$
- 88 •  $\int |\omega_{wheel,z}| dt > 80^\circ$

89 Third, valid wheeling periods that are less than  $2 \text{ s}$  apart are fused. In addition, rest phases between  
90 two valid wheeling periods that contain more than  $80\%$  of preliminary wheeling periods as well as  
91 those that are shorter than  $0.8 \text{ s}$  are also classified as a valid wheeling period. Finally, valid wheeling  
92 periods that are less than  $2 \text{ s}$  apart are fused again.

93 *3.2.2. Discrimination between active and passive wheeling*

94 As a first step, the raw data of the wrist sensor is filtered. The acceleration signal is passed  
95 through an infinite impulse response eight order elliptic low-pass filter with a cut-off frequency of  
96  $0.3 \text{ Hz}$ , a passband ripple of  $0.02 \text{ dB}$ , and a minimum stopband attenuation of  $200 \text{ dB}$  in order  
97 to separate the static acceleration component due to gravity  $a_{static}$  from the dynamic acceleration  
98 component resulting from wrist movement  $a_{dynamic}$ . [6]

99  
100 Wheeling periods lasting longer than  $5.12 \text{ s}$  are divided into segments with a window length of  
101  $5.12 \text{ s}$  and an overlap of  $75\%$ . Each segment in which the wrist sensor is not able to communicate  
102 with the wheelchair sensor via Bluetooth Low Energy is classified as non-wheeling activity. Here,  
103 it is assumed that the wheelchair is far away from the participant. The remaining segments are  
104 either classified as active or passive wheeling. The same features of the original publication of this  
105 algorithm [5] were calculated and the feature selection process was repeated with data of children  
106 with mobility impairments. It revealed just a single relevant feature:  $P_{10th}(a_{wrist,static,x})$ . This  
107 feature is a surrogate for the orientation of the wrist. It is  $9.81 \text{ m/s}^2$  if the z-axis is parallel to  
108 gravity and zero if the z-axis is perpendicular to gravity. Movement related features did not improve  
109 classification accuracy since we encouraged children to do hand activities while they were pushed  
110 around in their wheelchair. The threshold to distinguish between active and passive wheeling was  
111 trained with a decision tree by minimizing the Gini's Diversity Index. The resulting threshold  
112 is  $T_{wrist} = -0.61 * 9.81 \text{ m/s}^2$ . Wheeling periods with the hand facing down towards the wheel

113 are classified as active wheeling while periods with the hand facing more horizontal or upward are  
114 classified as passive wheeling.

### 115 3.3. Outcome measures

116 The algorithm derives the total duration of active and passive wheeling separately.

## 117 4. Walking detection algorithm

118 This part of the algorithm detects walking periods and differentiates between level walking and  
119 stair climbing with data of a single ankle sensor. Further, the algorithm discriminates between free  
120 and assisted walking with data of the sensor attached to a walking aid.

### 121 4.1. Sensor placement

122 The ankle sensor is worn on the less-affected ankle. The x-axis faces toward the floor and points  
123 to the lateral malleolus (see Figure 4). If applicable, a sensor is placed firmly on the walking aid of  
the participant. The position and orientation of this sensor are irrelevant.

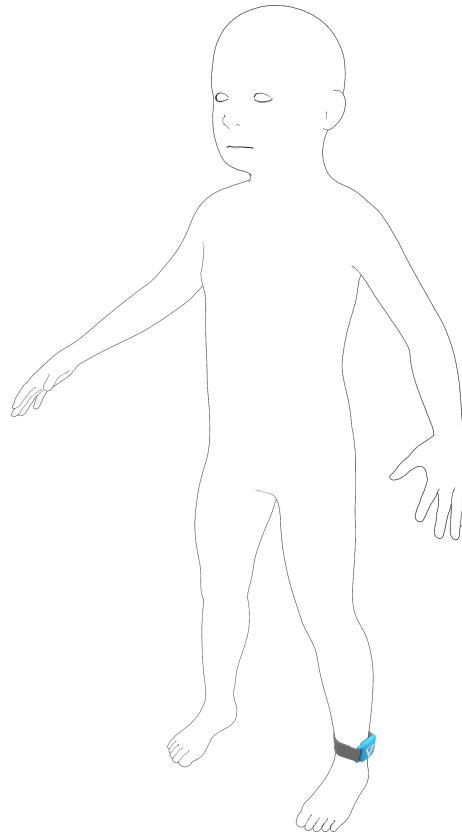


Figure 4: Sensor placement of the walking detection algorithm (created by Rehabilitation Engineering Laboratory, ETH Zurich).

124

125 *4.2. Detection of walking bouts*

126 *4.2.1. Preprocessing I*

127 As a first step, the algorithm verifies the placement of the ankle sensor. If the average accel-  
 128 eration signal of the x-axis is greater than zero, it is assumed that the sensor was placed upside  
 129 down. In this case, the sensor is rotated 180° around its z-axis by multiplying the acceleration and  
 130 gyroscope signals of the x- and y-axis by  $-1$ . Then, the bias and drift of the gyroscope signal are  
 131 corrected as described in chapter 2.2

132 *4.2.2. Segmentation and preprocessing II*

A 5th order low-pass filter (cut-off frequency = 3 Hz) is applied to the gyroscope signal. Then,  
 the signal is segmented into windows of 30 s and an overlap of 15 s. In each segment, the angular  
 rate around the mediolateral axis  $\omega_{ml}$  is determined by correcting for misalignment around the  
 x-axis of the ankle sensor and by the assumption that the majority of leg movement occurs in the  
 sagittal plane:

$$\begin{pmatrix} \cdot \\ \omega_{ml} \end{pmatrix} = \mathbf{v} \begin{pmatrix} \omega_y \\ \omega_z \end{pmatrix},$$

133 with  $\mathbf{v}$  being the eigenvector of  $cov(\omega_y, \omega_z)$  with the largest eigenvalue. Since the eigenvector can  
 134 point in both directions, the signal  $\omega_{ml}$  has to be multiplied with  $-1$  whenever it is upside down.  
 135 To verify this, the algorithm uses the fact that the angular rate is larger during the swing phase  
 136 compared to the stance phase. It compares the means of the upper and lower envelopes of the signal  
 137 and multiplies  $\omega_{ml}$  with  $-1$  whenever the mean of the lower envelop is larger than the the mean of  
 138 the upper envelop. Consequently, positive values of  $\omega_{ml}$  correspond to a backward rotation of the  
 139 shank as during the swing phase and vice versa. An exemplary signal is shown in Figure 5

140 *4.2.3. Step detection*

The algorithm detects steps by finding local maxima in  $\omega_{ml}$ , corresponding to the peak angular  
 rate during mid-swing of each step (see Figure 5). The amplitude of these maxima, as well as the  
 duration between two consecutive maxima, must exceed the thresholds  $T_{peakheight}$  and  $T_{peakdistance}$ ,  
 respectively. These thresholds are adapted to the underlying data of each segment and, thus, to  
 individual gait patterns.

$$T_{peakheight} = \max(50^\circ/s, 0.2P_{99th}(\omega_{ml}))$$

A minimum of 50°/s was chosen to exclude maxima of non-walking data.[7]

$$T_{peakdistance} = \frac{0.5}{\tilde{f}_{walking}}$$

141 with  $\tilde{f}_{walking}$  being the median estimated step frequency (steps per second) of each segment. An  
 142 initial step frequency  $f_{initial}$  is estimated for the whole segment by applying a fast Fourier trans-  
 143 formation to  $\omega_{ml}$  and taking the first main frequency component. An adapted step frequency is  
 144 estimated by repeating this step in a sliding window of  $\frac{3}{f_{initial}}$  and an overlap of  $\frac{2}{f_{initial}}$ . Eventu-  
 145 ally, a moving average filter with a span of  $\frac{9}{f_{initial}}$  is applied to determine the time-dependent step  
 146 frequency  $f_{walking}$ . Eventually, only steps of the middle 15 s of each segment are considered to  
 147 avoid duplicates in overlapping segments.

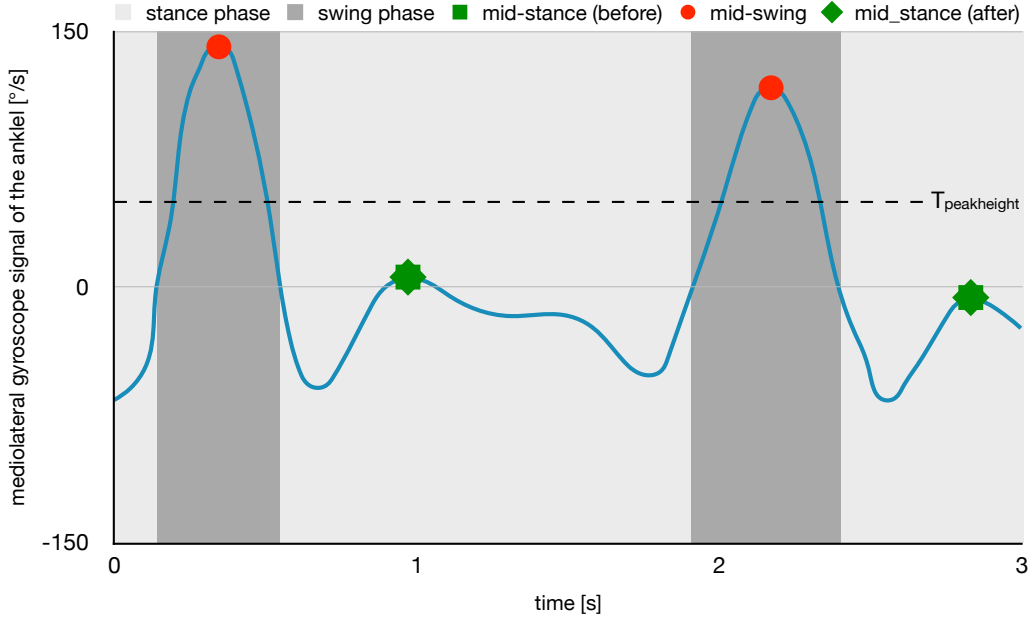


Figure 5: Exemplary illustration of the gyroscope signal of two steps as well as the corresponding gait phases and gait events detection.

148 **Removing unreasonable steps** The swing phase of each step is defined as the time  $t_{swing}$  be-  
 149 tween the first zero-crossings of  $\omega_{ml}$  before and after mid-swing. The stance phase is defined  
 150 as the time  $t_{stance}$  between the first zero-crossing of  $\omega_{ml}$  after the preceding mid-swing and  
 151 the beginning of the current swing phase. Steps with  $t_{swing} < 100 \text{ ms}$  or  $t_{stance} < 200 \text{ ms}$   
 152 are not considered valid steps.[8]

#### 153 4.2.4. Mid-stance detection and classification of walking periods

154 The algorithm detects the mid-stance before and after each mid-swing detected above, and the  
 155 period between the two mid-stance is classified as walking. During continuous gait, the mid-stance  
 156 after one mid-swing is equal to the mid-stance before the subsequent mid-swing and the whole  
 157 period is classified as walking. During interrupted gait, the two mid-stance do not overlap, and the  
 158 period between is classified as non-walking (see Figure 6). Mid-stance is defined as the time of the  
 159 largest local maximum in  $\omega_{ml}$  during the stance phase (see Figure 5 and Figure 6). The occurrence  
 160 of local maxima is a typical characteristic of gait. If there is no local maximum (e.g., during cycling  
 161 periods), the algorithm removes the corresponding step. Moreover, the angular rate of the local  
 162 maxima is usually negative which corresponds to a forward progression of the shank. However, the  
 163 angular rate can reach positive values during walking on uneven surfaces or stair climbing. Still,  
 164 the local maxima during the stance phases are considerably smaller than those during the swing  
 165 phases. Therefore, the algorithm removes steps whenever the local maximum during the stance  
 166 phase exceeds half of the maximum during the swing phase. Eventually,  $t_{stance}$  needs to be smaller  
 167 than  $\frac{3}{f_{walking}}$ . Otherwise, the mid-stance after the preceding step is set to the end of the preceding  
 168 swing phase, and the mid-stance before the subsequent step is set to the beginning of the subsequent



step. Consequently, the stance phase will be classified as a non-walking period.

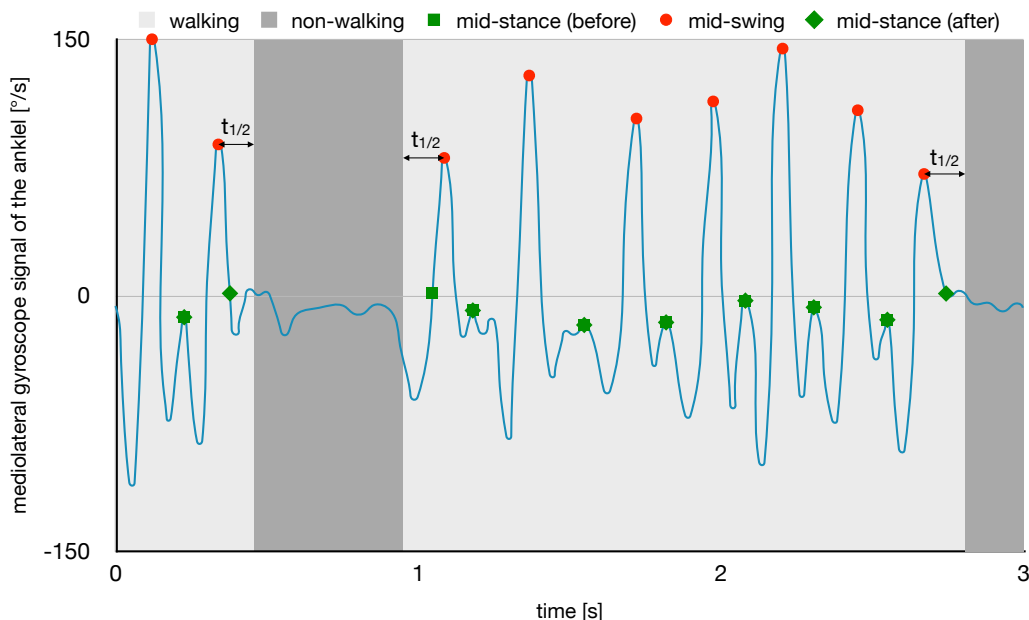


Figure 6: Exemplary illustration of the gyroscope signal of interrupted gait as well as the corresponding gait events detection and the resulting walking classification.  $t_{1/2}$  represents half of the mean step duration of each walking period to determine the corresponding start and end points

169

**Removing unreasonable steps** The algorithm determines the orientation of the ankle sensor  $\varphi_{ankle}$  (see chapter 2.2) between the mid-stance before and after each step. Then, steps with unreasonable orientation and range of motion are classified as non-walking periods if one of the following criteria is fulfilled:

$$\begin{aligned}
 \min(\varphi_{ankle}) &> -45^\circ \\
 \max(\varphi_{ankle}) &> 0^\circ \\
 \max(\varphi_{ankle}) - \min(\varphi_{ankle}) &< 5^\circ \\
 \max(\varphi_{ankle}) - \min(\varphi_{ankle}) &> 90^\circ
 \end{aligned}$$

#### 170 4.2.5. Break detection

171 This part of the algorithm detects breaks within each walking period and classifies them as  
 172 non-walking. It is assumed that the step duration between two consecutive mid-swings remains  
 173 relatively constant during continuous gait. Therefore, breaks are detected with long and irregular  
 174 step durations. The specific criteria depend on the number of steps within each walking period and  
 175 are defined as follows:

176  **$\geq 4$  steps** First, the algorithm calculates the median step duration of four consecutive steps. If  
 177 the step duration of one of these steps is greater than one and a half times the median step  
 178 duration, the corresponding stance phase is classified as non-walking. This part is repeated  
 179 for each set of four consecutive steps.

180 **3 steps** If one of the two step durations is more than twice as long as the other, the whole period  
181 is classified as non-walking.

182 **2 steps** If the step duration is longer than 5 s, the whole period is classified as non-walking.

183 **1 step** Walking periods with a single step are ignored and classified as non-walking.

#### 184 4.2.6. Start and end point

185 At the beginning and end of each walking period, there is no typical mid-stance. Hence, each  
186 walking period begins half of the mean step duration before the mid-swing of the first step and  
187 ends at half of the mean step duration  $t_{1/2}$  after the mid-swing of the last step (see Figure 6).

#### 188 4.3. Use of walking aids

The algorithm classifies each walking period as either free walking or assisted walking. If the participant does not use a walking aid and there is no data available, all walking periods are classified as free walking. Walking periods in which the ankle sensor is not able to communicate with the sensor on the aid via Bluetooth Low Energy are classified as free walking, too. Here, it is assumed that the walking aid is far away from the participant. To determine whether the walking aid was used or not, the acceleration signal of the sensor placed on the walking aid is processed with a high-pass filter and a cut-off frequency of 0.3 Hz to remove the gravity component in the signal. Then, for each walking period, the algorithm verifies if the 95th percentile of the magnitude of the filtered signal  $a_{aid}$  is above a predefined threshold  $T_{aid} = 0.05 * 9.81 m/s^2$  to determine whether the walking aid was moved around or not:

$$P_{95th} \left( \sqrt{a_{aid,x}^2 + a_{aid,y}^2 + a_{aid,z}^2} \right) \begin{cases} \leq T_{aid} & \implies \text{free walking} \\ > T_{aid} & \implies \text{assisted walking} \end{cases}$$

#### 189 4.4. Detection of stair climbing

190 The algorithm detects stair climbing periods based on the altitude change per step. Previously  
191 detected walking periods (independent of the use of walking aids) are classified as level walking,  
192 going upstairs, or going downstairs.

##### 193 4.4.1. Altitude estimation

The pressure signal  $p$  is transformed to the altitude above sea level  $h$  with the following formula<sup>1</sup>:

$$h = \log \frac{1013}{p} * 7990$$

194 Then, a median filter with a window length of five samples is applied. The resulting signal is filtered  
195 with an 8th order decomposition, heuristic, automatic 1-D de-noising filter using a soft threshold  
196 and symlet8 wavelet.[9]

---

<sup>1</sup>a simplification of the international barometric formula

197 *4.4.2. Expected altitude change per step*

198 to 42 cm since data of a single ankle sensor is used and participants can walk in a step-by-step  
199 or a step-over-step pattern. The algorithm adds a margin of 7 cm, which corresponds to the half  
200 of the smallest expected step height. Therefore, the lower border for discriminating between level  
201 walking and stair climbing was set to 7 cm/step, and the upper border was set to 49 cm/step. An  
202 upper border is needed as large altitude changes can occur when the environmental temperature  
203 changes rapidly (e.g., when walking out of a heated building).

204 *4.4.3. Classification of going upstairs and going downstairs*

Walking periods containing less than four steps are always classified as level walking. The  
remaining walking periods are segmented into windows of four consecutive steps and an overlap of  
three steps. For each window, the algorithm determines the altitude change and compares it to the  
expected altitude change described above:

$$\begin{aligned} 7 \text{ cm/step} < \frac{\Delta h}{4 \text{ steps}} < 49 \text{ cm/step} &\implies \text{going upstairs} \\ -49 \text{ cm/step} < \frac{\Delta h}{4 \text{ steps}} < -7 \text{ cm/step} &\implies \text{going downstairs} \end{aligned}$$

205 *4.5. Outcome measures*

206 Eventually, the algorithm derives the free and assisted walking duration and estimates the  
207 covered altitude change during stair climbing periods.

208 **5. Acknowledgments**

209 This algorithm is an extension of previous algorithms developed at the Rehabilitation Engineer-  
210 ing Laboratory, ETH Zurich (RELab). Therefore, I want to express my gratitude to three former  
211 and current members of the RElab for providing me with the codes of their work. Specifically, I  
212 want to thank Kaspar Leuenberger for providing his codes regarding the preprocessing of sensor  
213 data, Werner Popp for providing his wheeling detection algorithm for adults with a spinal cord  
214 injury, and Charlotte Werner for providing her codes regarding gait detection. Moreover, I want to  
215 thank Charlotte for revising this manuscript.

216 **References**

- 217 [1] W. L. Popp, S. Schneider, J. Bär, P. Bösch, C. M. Spengler, R. Gassert, A. Curt, *Wearable*  
218 *Sensors in Ambulatory Individuals With a Spinal Cord Injury: From Energy Expenditure Es-*  
219 *timation to Activity Recommendations*, *Frontiers in neurology* 10 (2019) 1092.
- 220 [2] K. D. Leuenberger, R. Gassert, A. Luft, *Long-term activity and movement monitoring in neu-*  
221 *rological patients*, Ph.D. thesis, ETH Zürich (2015).
- 222 [3] J. Lötters, J. Schipper, P. Veltink, W. Olthuis, P. Bergveld, *Procedure for in-use calibration*  
223 *of triaxial accelerometers in medical applications*, *Sensors and Actuators A: Physical* 68 (1-3)  
224 (1998) 221–228.
- 225 [4] S. O. H. Madgwick, A. J. L. Harrison, R. Vaidyanathan, *Estimation of IMU and MARG orien-*  
226 *tation using a gradient descent algorithm*, in: *2011 IEEE International Conference on Rehabil-*  
227 *itation Robotics*, 2011, pp. 1–7.

- 228 [5] W. L. Popp, M. Brogioli, K. Leuenberger, U. Albisser, A. Frotzler, A. Curt, R. Gassert, M. L.  
229 Starkey, A novel algorithm for detecting active propulsion in wheelchair users following spinal  
230 cord injury, *Medical Engineering & Physics* 38 (3) (2016) 267–274.
- 231 [6] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, B. G. Celler, Implementation of a  
232 real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring,  
233 *IEEE Transactions on Information Technology in Biomedicine* 10 (1) (2006) 156–167.
- 234 [7] A. Salarian, H. Russmann, F. J. G. Vingerhoets, C. Dehollain, Y. Blanc, P. R. Burkhard,  
235 K. Aminian, Gait assessment in Parkinson’s disease: toward an ambulatory system for long-  
236 term monitoring, *IEEE transactions on bio-medical engineering* 51 (8) (2004) 1434–1443.
- 237 [8] D. Trojaniello, A. Cereatti, E. Pelosin, L. Avanzino, A. Mirelman, J. M. Hausdorff,  
238 U. Della Croce, Estimation of step-by-step spatio-temporal parameters of normal and impaired  
239 gait using shank-mounted magneto-inertial sensors: application to elderly, hemiparetic, parkin-  
240 sonian and choreic gait, *Journal of NeuroEngineering and Rehabilitation* 11 (2014) 152.
- 241 [9] K. Leuenberger, R. Gonzenbach, E. Wiedmer, A. Luft, R. Gassert, Classification of stair ascent  
242 and descent in stroke patients, 2014 11th International Conference on Wearable and Implantable  
243 Body Sensor Networks Workshops (Bsn Workshops) (2014) 11–16.