Pontificia Universidad Católica del Perú

Escuela de Posgrado

A method of estimation of relevant trunk control parameters for cerebral palsy rehabilitation in children during independent walking based on inertial measurement unit data

Tesis para obtener el grado académico de Maestra en Ingeniería Mecatrónica que presenta:

Maria Alejandra Guzman Alfaro

Asesor PUCP (PUCP): *Dr. Dante Angel Elias Giordano*

Co-Asesor de la Universidad no PUCP: *Prof. Dipl. -Ing. Dr.Hartmut Witte*

Lima, 2024

Informe de Similitud

Yo, Dante Angel Elias Giordano, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor(a) de la tesis titulada(o) A method of estimation of relevant trunk control parameters for cerebral palsy rehabilitation in children during independent walking based on inertial measurement unit data, de la autora Maria Alejandra Guzman Alfaro, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 7%. Así lo consigna el reporte de similitud emitido por el software Turnitin el 11/12/2024.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de investigación, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lima, 12 de noviembre del 2024.

<table>
<tr><td colspan="2">Apellidos y nombres del asesor: Elias Giordano, Dante Angel</td></tr>
<tr><td>DNI: 10142907</td><td rowspan="2">Firma  </td></tr>
<tr><td>ORCID: https://orcid.org/0000-0001-5920-9608</td></tr>
</table>

Resumen

La parálisis cerebral (CP) es una discapacidad del desarrollo infantil que genera un impacto en la capacidad de moverse y caminar. Algunos estudios biomecánicos han desarrollado en el análisis de marcha de niños con CP que caminan de forma independiente, estimando parámetros que cuantifican el rendimiento del movimiento usando unidades de medición inercial (IMU). No obstante, solo una investigación ha proporcionado información acerca de la comparación antes y después del tratamiento fisioterapéutico en niños con CP.

Un estudio de viabilidad se está llevando a cabo por la DRK-Kinderklinik en Siegen para dar prueba del concepto acerca de la mejora del control postural después de un bloque intensivo de terapia Vojta. Para este estudio, seis niños con CP y seis niños con desarrollo típico (TD) realizaron pruebas de marcha de 5 metros ida y vuelta, tres veces, usando sensores inerciales en el esternón, lumbar y en el dorso de los pies.

El objetivo de esta investigación es identificar los cambios en el rendimiento del balance del tronco en la marcha de niños con CP mediante un análisis de los datos inerciales. Esta información fue analizada en los dominios de tiempo y frecuencia, y con ella, parámetros de control del tronco fueron identificados y cuantificados. El algoritmo de MATLAB desarrollado consiste en una detección automática de caminata y giros, una detección parcialmente automática de los ciclos de marcha, y la estimación de parámetros de la marcha en los dominios de tiempo y frecuencia.

Cuatro grupos de parámetros fueron evaluados: espaciotemporal, suavidad, armonicidad y movimiento angular. Los parámetros espaciotemporales son la velocidad, cadencia, tiempo de paso y número de pasos. Los parámetros de suavidad son la media cuadrática (RMS) y el coeficiente de atenuación (AC) de las aceleraciones por cada eje. La relación armónica (HR) evalúa la armonicidad. El movimiento angular fue determinado por los ángulos absolutos del esternón y la columna lumbar por eje, así como los ángulos relativos entre ambas posiciones.

Después de la terapia Vojta, el único parámetro de la marcha que varía para todos los niños con CP evaluados es el mediolateral (ML) AC. Asimismo, cinco de los seis niños presentaron una variación en el ángulo roll del esternón. Cuatro niños mostraron una variación en la ML RMS del esternón, ML HR del esternón y lumbar, anteroposterior (AP) RMS del lumbar, vertical (VT) AC, y el ángulo pitch relativo. Sin embargo, la evaluación de solo un parámetro provee información incompleta, por lo tanto, un grupo de parámetros debe ser analizado.

Abstract

Cerebral palsy (CP) is an infantile development disability that has an impact on the ability to walk or move. Some biomechanical studies concentrated on the gait analysis of children with CP have been conducted and estimated parameters, which quantify the performance of the movement using inertial measurement units (IMU) during independent walking. However, only one investigation has given information about the comparison before and after physiotherapeutic treatment by children with CP.

A feasibility study is conducted by the DRK-Kinderklinik Siegen to provide proof of the concept that improvement in trunk postural control following an intensive block of Vojta therapy. For this thesis, six children with CP and six in age and gender matched typically developing (TD) children performed a 5 m walking test back and forth three times, while wearing inertial sensors on sternum, lumbar and feet.

The aim of this thesis is to identify changes in the performance of the trunk balance in the gait of children with CP by analysis of the IMU data. This data was analyzed in time and frequency domain and trunk control parameters were identified and quantified. The MATLAB-algorithm consists of an automatic walk and turn detection, a partially automatic gait cycle detection and the estimation of gait parameters in time and frequency domain. Four groups of parameters were evaluated: spatiotemporal, smoothness, harmonicity and angular motion. Spatiotemporal parameters are speed, cadence, step time, number of steps. Smoothness parameters are root mean square (RMS) and the attenuation coefficient (AC) of accelerations per axis. The harmonic ratio (HR) rates the Harmonicity. Angular motion was determined by the absolute angles per axis of the sternum and lumbar spine, as well as the relative angles from both positions.

Following Vojta therapy, the mediolateral (ML) AC is the sole parameter that varies for all the evaluated children with CP. In addition, five of the six children exhibited a variation in the sternum roll angle. Meanwhile, four children exhibited a variation in the sternum ML RMS, sternum and lumbar ML HR, lumbar AP RMS, vertical (VT) AC, and the relative pitch angle. However, the evaluation of only one parameter provides incomplete information, thus, a group of parameters must be analyzed.

Kurzfassung

Zerebralparese (CP) ist eine frühkindliche Erkrankung, welche die Bewegung und das Gleichgewicht beeinträchtigt. Es wurden verschiedene biomechanische Studien durchgeführt, die sich auf die Ganganalyse bei Kindern mit CP konzentrierten und Parameter schätzten, die die Leistung der Bewegung unter Verwendung von Inertialmesseinheiten (IMU) während des eigenständigen Gehens quantifizieren. Es gibt jedoch nur eine Untersuchung, die Aufschluss über den Vergleich vor und nach einer physiotherapeutischen Behandlung bei Kindern mit CP gibt.

Die DRK-Kinderklinik Siegen führt eine Machbarkeitsstudie durch, um den Nachweis zu erbringen, dass sich die Rumpfhaltungskontrolle nach einem intensiven Block Vojta-Therapie verbessert. Für diese Arbeit absolvierten sechs Kinder mit CP und sechs alters- und geschlechtsäquivalente Kinder mit normaler Entwicklung (TD) dreimal hin und her einen 5 m-Gehtest, während IMU am Sternum, im Lendenbereich und auf den Füßen getragen werden.

Das Ziel dieser Masterarbeit besteht darin Veränderungen des Rumpfgleichgewichts im Gang von Kindern mit CP anhand von IMU-Daten zu identifizieren. Dafür wurden die Daten im Zeit- und Frequenzbereich analysiert und Parameter für die Bewertung der Rumpfhaltungskontrolle identifiziert und quantifiziert. Der MATLAB-Algorithmus besteht aus einer automatischen Lauf- und Drehungserkennung, einer teilautomatischen Gangzykluserkennung und der Schätzung von Gangparametern im Zeit- und Frequenzbereich. Die Bewegung wurde anhand von vier Gruppen von Parametern bewertet. Räumlich-zeitliche Parameter sind die Geschwindigkeit, Gleichschritt, Schrittzeit und Anzahl der Schritte. Glättungsparameter sind der quadratische Mittelwert (RMS) und der Dämpfungskoeffizient (AC) der Beschleunigungen pro Achse. Mittels Oberschwingungsverhältnis (HR) bewertet die Harmonizität. Die Rotationsbewegung wird durch die absoluten Winkel pro Achse des und der Lendenwirbelsäule sowie durch die relativen Winkel aus beiden Positionen bestimmt.

Nach der Vojta-Therapie ist der mediolaterale (ML) AC der einzige Parameter, der sich bei allen untersuchten Kindern mit CP verändert. Außerdem wiesen fünf der sechs Kinder eine Veränderung des Sternum-Rollwinkels auf. Vier Kinder wiesen eine Veränderung des ML RMS des Sternums, des ML HR des Sternums und der Lendenwirbelsäule, des AP RMS der Lendenwirbelsäule, des vertikalen (VT) AC und des relativen Nickwinkels auf. Die Auswertung nur eines Parameters liefert jedoch nur unvollständige Informationen, so dass eine Gruppe von Parametern analysiert werden muss.

Acknowledgements

Dedico con todo mi corazón esta tesis a mi familia, a mis padres Rosa Alfaro y Orlando Guzmán, quienes me han acompañado, siempre me han dado un apoyo incondicional, fuerza y amor que traspasa todas las fronteras y kilómetros, y porque sin ellos no sería la persona que soy ahora.

A mis abuelitas que me cuidan y acompañan desde el cielo y mi abuelito a quién espero volver a abrazar. Asimismo, dedico mi trabajo a mis hermanos, Claudia y Daniel Guzmán, quienes a su vez me apoyaron totalmente y siempre me sacaron una sonrisa a pesar de la distancia.

I would like to express my deepest gratitude to Prof. Hartmut Witte and M.Sc. Sabine Bruchmüller for providing the opportunity to embark on a novel investigative journey, the wholesome support they gave me and all the knowledge I have learnt by doing this master thesis. Moreover, I would like to thank Prof. Dante Elias for his great support throughout the years and for always having the right and concise answer to my questions.

I would like to express my deepest gratitude to my dearest friend, who is the one that always has my back and supports me through every step of the way. Also, I am grateful to all the friends that helped me with this process including those who accompanied me from Peru and those I have met in Germany.

Contents

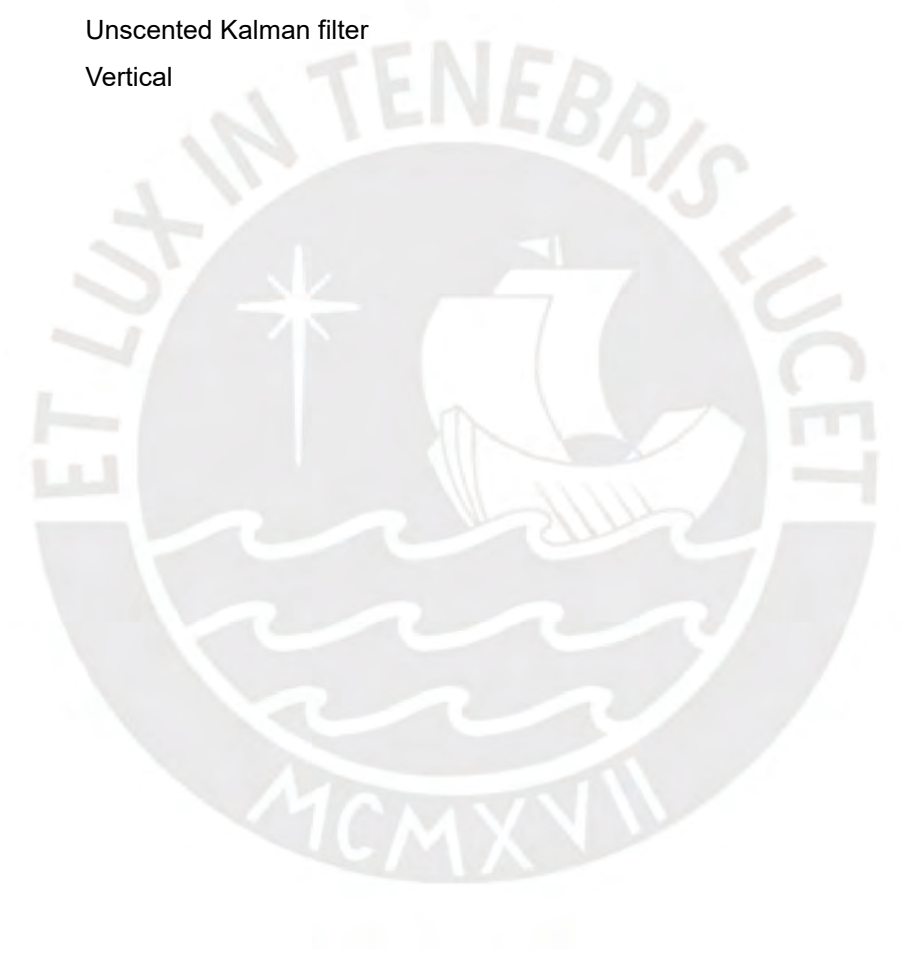
List of figures

List of tables

List of abbreviations

AC	Attenuation Coefficient
AGG	Average Gait Graph
AP	Anteroposterior
AR	Axial rotation
CDC	Centers for Disease Control and Prevention
CF	Complementary filter
CGG	Characteristic Gait Graph
CIMT	Constraint-Induced Movement Therapy
COM	Center of mass
CWT	Continuous Wavelet Transform
CP	Cerebral Palsy
DP	Diplegia
DOF	Degrees of Freedom
DFT	Discrete Fourier Transform
DRK	Deutsches Rotes Kreuz
DS	Double support phase
EKF	Extended Kalman filter
EMG	Electromyography
FE	Flexion or extension
FFT	Fast Fourier Transform
FSM	Finite State Machine
GC	Gait Cycle
GMFCS	Gross Motor Function Classification System
hAFO	Hinged ankle-foot orthosis
HO	Heel off
HP	Hemiplegia
HR	Harmonic Ratio
HS	Heel strike
iHR	Improved Harmonic Ratio
IMU	Inertial Measurement Unit
KF	Kalman filter
LB	Lateral bending
midSt	Midstance phase
midSw	Midswing phase
MEMS	Micro-electro-mechanical
ML	Mediolateral
mWT	Meter walking test
PCA	Principal component analysis
RI	Ratio Index
ROM	Range of Movement

RMS	Root Mean Square
RMSE	Root Mean Square Error
RMSR	Root Mean Square Ratio
sAFO	Solid ankle-foot orthosis
SB	Spina bifida
SCPE	Surveillance of Cerebral Palsy in Europe
TCMS	Trunk Control Measurement Scale
TIS	Trunk Impairment Scale
TD	Typically developing
TO	Toe off
TS	Toe strike
TUG	Timed Up and Go test
UKF	Unscented Kalman filter
VT	Vertical



List of symbols

a_{min}	Minimum peak of accelerations
a_{x-IMU}	Raw acceleration from the IMU in the x axis
a_{y-IMU}	Raw acceleration from the IMU in the y axis
a_{z-IMU}	Raw acceleration from the IMU in the z axis
A_{i*2}	Even harmonics from a signal
A_{i*2-1}	Odd harmonics from a signal
AC_{PS}	Attenuation coefficient from the pelvis to sternum
AC_{PH}	Attenuation coefficient from the pelvis to head
AC_{SH}	Attenuation coefficient from the sternum to head
$C_{PS}\%$	Attenuation coefficient from the pelvis to sternum from Summa et al. (2016)
$F(t)$	State transition model
f_0	Fundamental frequency
f_c	Cutoff frequency
$\mathbf{f}(\mathbf{x}(t), t)$	State model continuous and differentiable function
F_c	Sensed force
g	Gravity force
$\bar{g}$	Gravity vector
$G(s)$	Low-pass Laplace transform
$G(t)$	Process noise state model
GC_i	Gait Cycle Nr. *i*
$\mathbf{h}$	Measurement model
H_k	Observed model
H_x	Magnetic vector in function of the displacement in x direction
H_y	Magnetic vector in function of the displacement in y direction
HR_{AP}	Harmonic ratio from the AP axis
HR_{ML}	Harmonic ratio from the ML axis
HR_{VT}	Harmonic ratio from the VT axis
$IMU_{filtered}$	Filtered IMU data in the global frame
IMU_{global}	IMU data in the global frame
IMU_{raw}	Raw IMU data
IMU_{W_i}	Processed IMU data per walk W_i
$IMU_{W_{iGC_j}}$	Processed IMU data per walk W_i and per GC_j
K	Number of elements from the signal array *x*
m	Resonating mass
M	Number of points to points for the averaging process of the moving average filter
$\tilde{m}_{x,y,z}$	Raw magnetic field in the x, y and z directions
M_x	Global frame magnetic field in the x axis
M_{x-IMU}	Raw magnetic field from the IMU in the x axis
M_y	Global frame magnetic field in the y axis
M_{y-IMU}	Raw magnetic field from the IMU in the y axis
M_z	Global frame magnetic field in the z axis
M_{z-IMU}	Raw magnetic field from the IMU in the z axis
$\bar{M}_{IMU}$	Raw magnetic field vector from the IMU
N	Order of the Butterworth filter
PP_v	Peak-to-peak value of angular velocities
q_{lumbar}	Quaternion of the motion of the lumbar spine
q_{rel}	Relative quaternion from the lumbar spine to the sternum
q_{roll}	Roll angle quaternion
q_{RPY}	Final rotation quaternion
$\bar{q}_{RPY}$	Conjugate of q_{RPY}

$q_{sternum}$	Quaternion of the motion of the sternum
q_{pitch}	Pitch angle quaternion
q_{yaw}	Yaw angle quaternion
$Q(t)$	Spectral density
R_k	Covariance
R_ϕ	Rotation matrix of the ϕ angle
R_θ	Rotation matrix of the θ angle
R_ψ	Rotation matrix of the ψ angle
$RMS_{acc-lumbar}$	RMS magnitude of the lumbar acceleration
$RMS_{acc-sternum}$	RMS magnitude of the sternum acceleration
R_{XYZ}	Rotation matrix in the XYZ
RMS_{AP}	RMS magnitude in the AP axis
RMS_{lower}	RMS magnitude of a lower sensor
RMS_{ML}	RMS magnitude in the ML axis
RMS_{total}	Total magnitude of RMS for every direction
RMS_{upper}	RMS magnitude of an upper sensor
RMS_{VT}	RMS magnitude in the VT axis
t	Time array
$\mathbf{v}$	Drive axis velocity
$\mathbf{v}(t)$	System noise
v_{global}	Global frame IMU quaternion matrix
v_{IMU}	Raw IMU information quaternion matrix
$\Delta W_{i_{time}}$	Time duration of the W_i
$\mathbf{x}$	State vector
x	Input signal of the Moving average filter
x_n	Element *n* from the signal array *x*
y	Ouput signal of the Moving average filter
$\mathbf{w}(t)$	White Gaussian measurement noise
W_i	Walk Nr. i
$x(n)$	Discrete signal sequence
$X(k)$	Fourier transform output
ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
ψ_m	Yaw angle calculated by the magnetometer only
α	Tuning parameter for the complementary filter from Kok et al. (2017)
β	Step length constant for the Gauss-Newton Optimization
$\Theta(s)$	Laplace transform of the estimated orientation
$\Theta_c(s)$	Combined Laplace transform of the orientation by the accelerometer and magnetometer
$\Theta_g(s)$	Laplace transform of the orientation estimate by the angular velocity
$\theta_c(t)$	Combined orientation estimate by the accelerometer and magnetometer
$\theta_g(t)$	Orientation estimate by the angular velocity
γ	Complementary filter tuning parameter
$\mathcal{N}(\mathbf{0}, Q(t))$	Gaussian distribution of the spectral density
$\mathcal{N}(\mathbf{0}, R_k(t))$	Gaussian distribution of the covariance
ω	Angular velocity of the drive
Δt_{GC_j}	Time duration of the GC_j

1. Introduction

1.1. Motivation

Cerebral palsy is the most prevalent motor disability among children, affecting 1 to 4 children per 1000 worldwide, according to the Centers for Disease Control and Prevention (CDC Archive, 2022). In developing countries like the USA, the affected proportion is 3 per 1000 8-year-old children. In Germany, 2 out of 1000 children suffer from cerebral palsy, according to the Charité Hospital in Berlin (Schaumann, 2024)

This disability can affect the child's ability to walk depending on the type of cerebral palsy the child is diagnosed with. According to the CDC Archive (2022), 58.9 % of the children can walk independently, while 7.8 % can walk using assistive devices and 33.3 % exhibit limited walking ability or are unable to walk. Some of the primary motor disabilities include loss of selective motor control, muscle spasticity and weakness, and secondary incapacities such as bony deformities and muscle contractures (Armand et al., 2016).

These statistics indicate that infantile cerebral palsy is a very common disability that affects the capacity of children to walk or move. Therefore, there have been many biomechanical studies focused primarily on gait analysis in children with cerebral palsy and parameters which quantify the performance of the movement using inertial measurement units (IMU) during independent walking (Sæther et al., 2014; Sæther et al., 2015; Iosa et al., 2012; Iosa et al., 2013; Summa et al., 2016; Mutoh et al., 2016; Tramontano et al., 2017; Mutoh et al., 2018; Paraschiv-Ionescu et al., 2019; Carcreff et al., 2020; Piitulainen et al., 2021).

Notwithstanding the investigations, the majority of them concentrate only on the analysis of the lumbar spine, incorporating feet data and estimating gait parameters through the analysis of stride patterns. It is notable that only two of the investigations in question, Summa et al. (2016) and Paraschiv-Ionescu et al. (2019), evaluate two distinct regions of the spine, sternum and lumbar spine. However, the calculated parameters were the Root Mean Square (RMS) accelerations and spatiotemporal parameters.

Conversely, the research conducted by Mutoh et al. (2016; 2018) are the sole study to make a comparison of the gait of CP children before and after Hippotherapy sessions. Besides them, there is no information about comparison before and after physiotherapeutic treatment in CP children.

Due to the mentioned complications, this work proposes to evaluate the changes made in the children's gait after the Vojta therapy using IMU sensors. Nevertheless, the ultimate goal of this thesis is to identify the optimal parameters for trunk stability and smoothness, and to determine how these parameters change following the implementation of blocks

of Vojta therapy. This is due to the fact that when there is a lack of movement on the feet, the children tend to use the trunk to stabilize themselves better.

1.2. Objectives

1.2.1. General objective

Identify gait cycle parameters in children with CP to quantify trunk control parameters by analyzing data in frequency and time domains.

1.2.2. Specific objectives

- Conduct comprehensive research on Cerebral Palsy (CP) and its effect on the gait cycle, and the use of inertial measurement units (IMU) in gait analysis.
- Elaborate an algorithm which provides the automatic detection of walk and turn segments.
- Elaborate an algorithm which provides the partially automatic detection of gait cycles per walk, by detecting the Heel strike event.
- Identify relevant spatio-temporal gait parameters and variables that reflect smoothness and trunk symmetry.
- Define, identify, and select parameters that differ between the signals observed before and after therapy.

1.3. Scope

This thesis will concentrate on the creation of signal processing algorithms utilizing IMU sensor data from four regions of the children's bodies (sternum, lumbar and feet). However, the majority of the processing will be focused on the trunk sensors, as the primary objective of this work is to identify and estimate parameters that assess the trunk smoothness.

The gait inertial data is provided by a feasibility study carried out in Deutsches Rotes Kreuz (DRK) Kinderklinik Siegen in Germany, that recorded the gait before and after performing Vojta therapy on six children with diverse ages and condition of cerebral palsy. The raw inertial data will be later processed using MATLAB R2024a software with a student license.

The development of this master’s thesis will be monitored on a monthly basis through meetings with Mrs. Fionn Bayley, physiotherapist responsible for the study, and the investigators overseeing the project from the Biomechatronics group on the TU Ilmenau.

2. Theoretical framework

2.1. Human locomotion

2.1.1. Gait cycle

The act of walking consists mainly in the alternation between static and dynamic phases, which involves not only the movement of the legs or lower body but also the movement of the upper body. Specifically, the human gait consists of a periodic movement of both feet while they exchange support, requiring also sufficient ground reaction forces. (Vaughan et al., 1999)

The human gait is composed of two important and main phases: stance and swing phase (see Figure 2.1). During the stance phase, the foot remains in contact with the ground, while during the swing phase, the same leg takes off and swings forward into the next stance phase. (Vaughan et al., 1999)

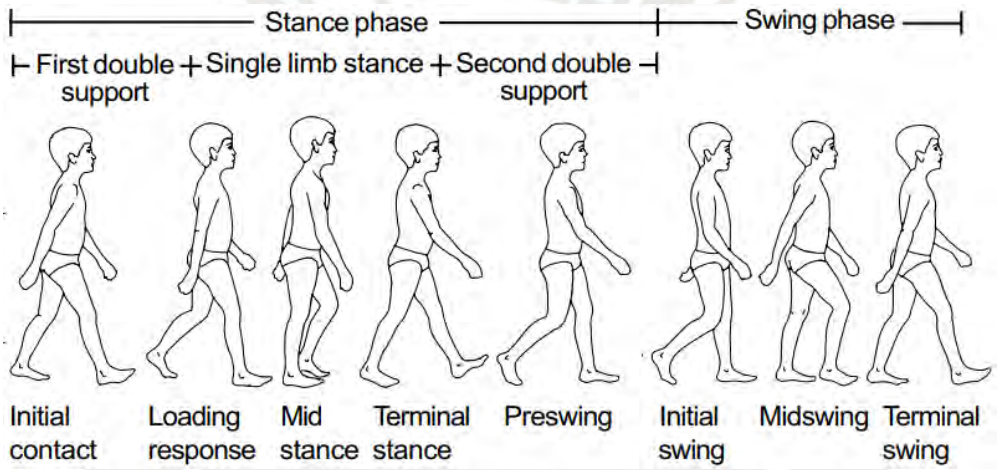


Figure 2.1 Phases and sub phases of the gait cycle (Vaughan et al., 1999).

In the stance phase, there are sub phases known as double support and single stance, as well as events know as heel strike (HS), toe strike (TS), heel off (HO), toe off (TO), as is illustrated on Figure 2.2.

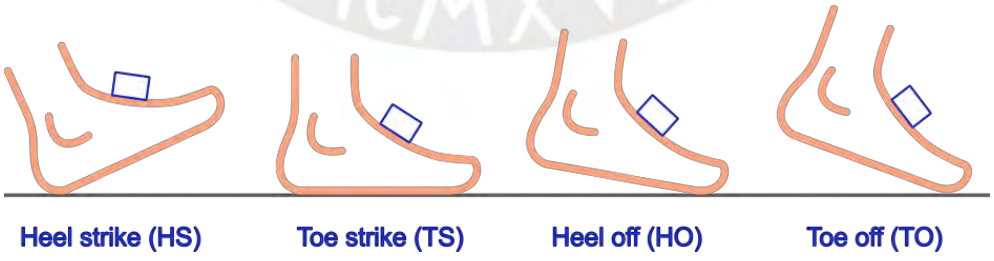


Figure 2.2 Gait events during stance phase.

The HS refers to the initial contact of the foot, while TS generates the foot flat event. In the Midstance event, the observed foot is fully in contact with the ground, while the other foot passes the foot in single limb stance. The HO marks the end of the double stance and initiates the pre swing event, which ends with the TO event.

2.1.2. Pathological gait cycle

The stance and swing phase represent the 62% and 38% of the gait according to Vaughan et al. (1999) for a typical developed adult. However, the gait cycle of people with musculoskeletal diseases or conditions differs from the normal standard. This illustrates Figure 2.3, for the gait of a patient with avascular necrosis and a patient with osteoarthritis. It shows that the percentage of gait per phase differs for both cases and, also, the duration of the gait cycle.

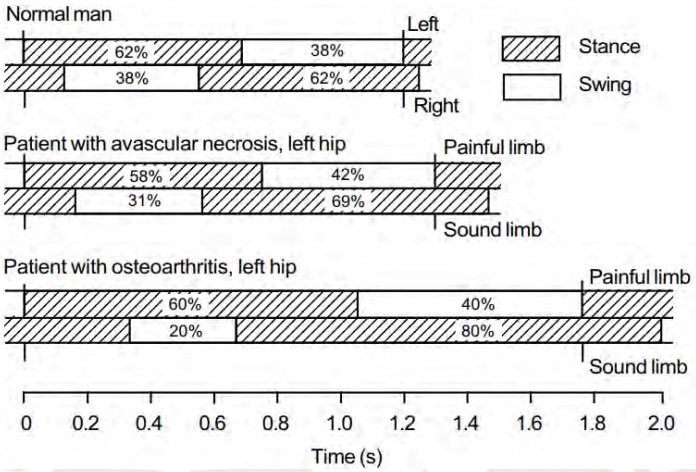


Figure 2.3 Comparison of normal and pathological gait (Vaughan et al., 1999).

As a conclusion, the gait duration and the gait percentage per phase and subphase differs depending on the condition of each person. There is not a pattern we can establish for each condition.

2.1.3. Gait parameters

Spatiotemporal parameters

Gait can be quantified in the time and frequency domains by certain parameters. The most common parameters shown in the literature are spatiotemporal parameters, which of course refer to the gait signals on the time domain.

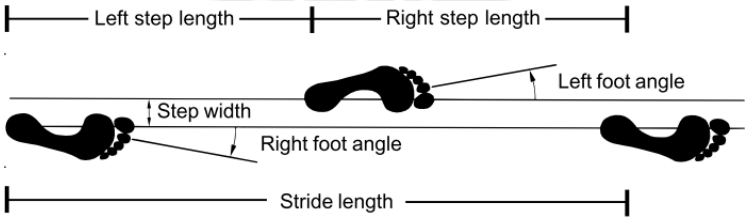


Figure 2.4 Spatial parameters of the gait (Vaughan et al., 1999).

Figure 2.4 presents some parameters, such as stride and step lengths, foot angles and step width. The length of a stride is considered as the sum of both footsteps' length. The angle of each foot is the curvature made between the imaginary line of progression of the

walk and the foot in observation. Finally, the step width is the distance perpendicular to the center line between both footsteps.

Other temporal parameters refer to the gait speed, cadence, the duration of the gait cycle and its phases, subphases and events. These include the duration of the stance phase and the double support subphase, among others.

Joint Kinematics

Additionally, the linear and angular displacement throughout the walks gives relevant information of movement of the lower and upper body joints. In the same line, the linear acceleration in the gait in all three axis provides important information in shape of patterns not only for body parts such as feet or knees, but also the different parts of the spine, shoulder girdle and hips. In some investigations, researchers tend to analyze the maximum, mean or minimum value of linear accelerations and angular displacement.

Root Mean Square (RMS)

The RMS of a signal x is given by Eq. 2.1 (MathWorks, 2024).

$$x_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|x_n|^2} \qquad \textit{(Eq. 2.1)}$$

This parameter is calculated from the acceleration and angular velocities obtained from IMU sensors to calculate the magnitude of the evaluated signal in a certain direction.

Attenuation Coefficient (AC)

The attenuation coefficient parameter provides a quantifiable manner to differentiate the acceleration data between an upper and a lower sensor (Paraschiv-Ionescu et al., 2019). The AC is defined in Eq. 2.2.

$$AC = \left(1 - \frac{RMS_{upper}}{RMS_{lower}}\right) \times 100\% \qquad \textit{(Eq. 2.2)}$$

Symmetry

Gait can be considered as a cyclic and rhythmic movement, where the events repeat for each footstep, right or left. Therefore, it is considered that both steps should have the same duration and patterns in the gait of typical developed people. That is because they present a natural symmetry between both sides, meanwhile, pathological gait presents different duration per gait phase depending on the condition (see Figure 2.3).

Fast Fourier Transform (FFT)

The FFT is the computation of the Discrete Fourier Transform (DFT) of a discrete sequence $x(n)$ given by the Eq. 2.3 (Proakis and Manolakis, 1996).

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi}{N}}, 0 \leq k \leq N-1 \qquad \textit{(Eq. 2.3)}$$

This parameter illustrates the frequencies decomposition of a signal on a time-domain, as illustrated on Figure 2.5. A periodic signal exhibits a prevalence of frequencies on the fundamental frequency f_0. Conversely, a nonperiodic signal, such as the second and third signals illustrated above, is represented on the fundamental frequency and its harmonics.

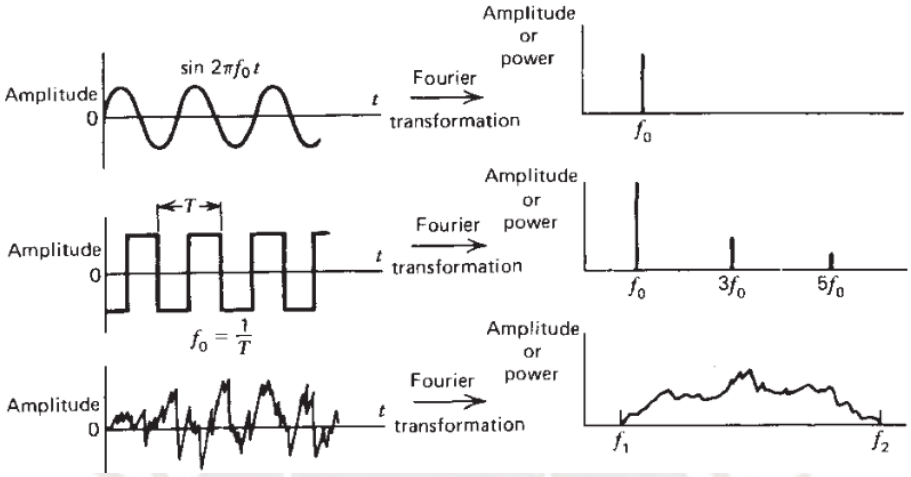


Figure 2.5 Fourier transformation of three different signals (Winter, 2009).

Harmonic ratio (HR)

The harmonic ratio enables the quantification of the rhythm and smoothness of the analyzed signal. Most of the gait is developed under a frequency of 10 Hz, therefore, the HR is calculated considering the first 20 harmonics of the signal. Since the act of walking consists of a stride divided by two steps and then, it would resemble in two patterns in a single stride in the anteroposterior (AP) and vertical (VT) axes, the HR_{AP} and HR_{VT} is estimated as the ratio of the sum of amplitudes of ten even harmonics (A_{i*2}) and the sum of ten odd harmonics (A_{i*2-1}). In contrast, the HR_{ML} would be the ratio of odd divided by even harmonics. (Castiglia et al., 2021)

$$HR_{AP} = HR_{VT} = \frac{\sum_i A_{i*2}}{\sum_i A_{i*2-1}}, HR_{ML} = \frac{\sum_i A_{i*2-1}}{\sum_i A_{i*2}} \qquad \textit{(Eq. 2.4)}$$

2.2. Cerebral palsy

Cerebral palsy (CP) is a group of gait and motor disorders affecting the nervous system that manifests as gait and motor impairments, resulting in a loss of balance and posture, according to the Center for Disease Control and Prevention of the United States (CDC, 2024).

In addition to the motor impairments that are the main symptoms of CP, other impairments are frequently observed in individuals with this condition. Such impairments encompass those affecting sensation, perception, cognition, communication, and behavior, in addition

to epilepsy, secondary musculoskeletal disorders, metabolic disorders, and non-progressive disorders. (Sadowska et al., 2020)

Its identification is based on motor impairments that usually develop in early childhood and endure until the end of life. However, CP diagnosis consists of a comprehensive assessment that includes interviews, clinical tests to identify motor impairments, and other additional examinations. Therefore, a long period of clinical observation is required. (Sadowska et al., 2020)

2.2.1. Classification

Cerebral palsy is classified in a number of ways, with each of them reflecting or providing insight into different characteristics of this condition.

The classification of CP in accordance with the Gross Motor Function Classification System (GMFCS) categorizes patients based on their level of performance regarding motor disorders (see Table 2.1). In addition, the scale comprises several categories pertaining to the age of the patient.

Table 2.1 Level of performance classification of patients with CP given the Gross Motor Function Classification System (Sadowska et al., 2020).

Performance Level	Characteristics
I	Can walk freely
II	Can walk but with certain limitations
III	Walk using auxiliary equipment
IV	Can move on their own, but with certain limitations (could use an electric wheelchair)
V	Lack of independent movement. Patient transported in wheelchair by caregiver

On the other hand, with regards to the Surveillance of Cerebral Palsy in Europe (SCPE), the CP clinical classification consists of three types (Christine et al., 2007).

- Spastic CP: Exhibit increased tone and pathological or increased reflexes. The tone can be characterized as an increase in resistance that depends on velocity.
- Dyskinetic CP: Display involuntary, uncontrolled, and recurrent movements. In addition, there is a variation in muscle tone, yet primitive reflex patterns remain predominant. It is divided into 2 subgroups.
    - Dystonic CP: Characterized by the presence of abnormal postures and hypertonia. Sustained muscle contractions are the underlying cause of involuntary movements, distorted voluntary gestures and irregular postures.
    - Choreo-athetotic CP: Characterized by the presence of hyperkinesia and hypotonia.

- Ataxic CP: Present a deficiency of muscular coordination resulting in an abnormal force, rhythm and precision of movement.

Regarding the Mixed CP, the SCPE indicates that the child should be classified in accordance with the most prominent clinical feature.

2.2.2. Gait disorders

The motor disorders on the gait of the children differs on the type of spastic CP: unilateral or bilateral (Armand et al., 2016).

Unilateral spastic CP

Gait disorders in unilateral spastic CP can be grouped into 4 categories by observing the movement on the sagittal plane (Armand et al., 2016), as illustrated in Figure 2.6.

- ***Group I:*** Distinguished by drop foot during swing phase.
- ***Group II:*** Distinguished by drop foot in the swing phase and permanent plantarflexion in the stance phase.
- ***Group III:*** Characterized by drop foot and limited knee flexion in the swing phase and permanent plantarflexion in the stance phase. As well as hip hyperflexion and intensified lumbar lordosis.
- ***Group IV:*** Characterized by every gait pattern described above and, additionally, restricted knee and hip motion.

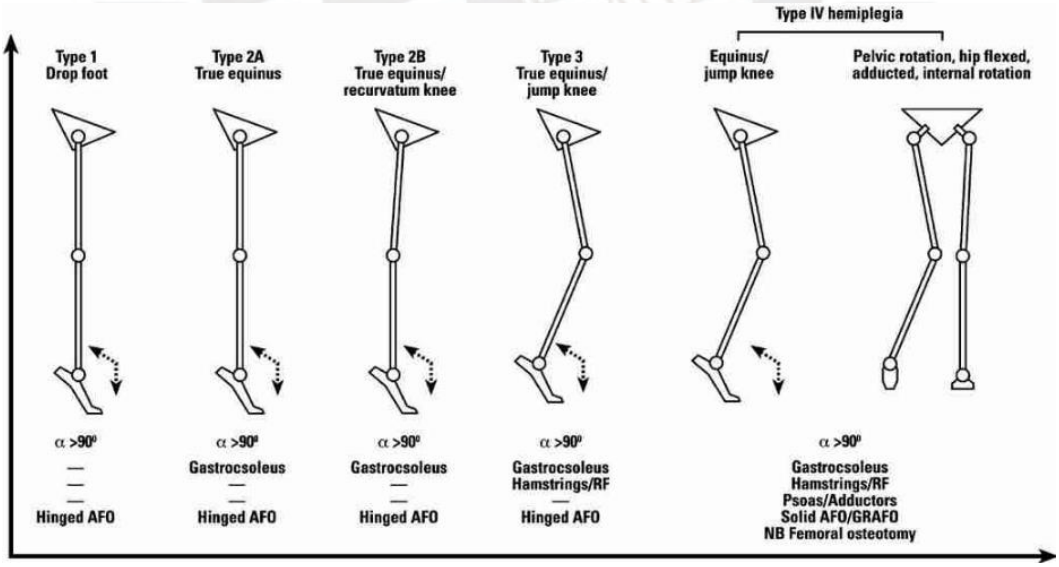


Figure 2.6 Unilateral spastic CP gait patterns, showing the variation of α as the angle between the forefoot and the shanks (Armand et al., 2016).

Bilateral spastic CP

Figure 2.7 illustrates that bilateral spastic CP gait patterns can also be categorized into four groups (Armand et al., 2016).

- ***True equinus:*** Distinguished by having the ankle in plantarflexion in the stance phase and, also, extended knees and hips.
- ***Jump knee:*** Characterized by having the ankle in plantarflexion, knee and hip flexion and, also, lumbar tilted with increased lordosis.
- ***Apparent equinus:*** Distinguished by an excessive flexion in the hip and knee in the stance phase, which provides an impression of walking on the toes. However, the range of ankle dorsiflexion is within the normal range.
- ***Crouch gait:*** Characterized by an excessive ankle dorsiflexion combined with extreme knee and hip flexion.

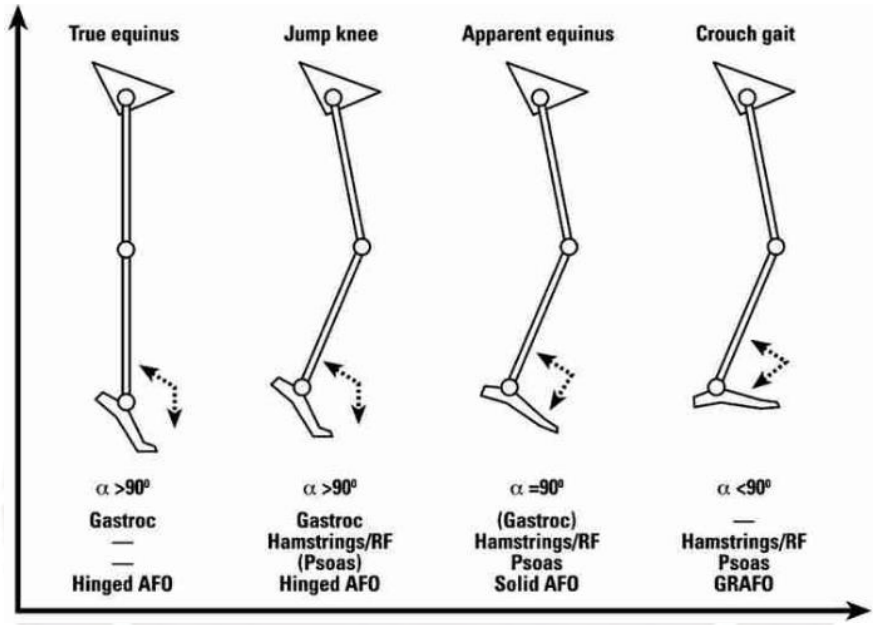


Figure 2.7 Bilateral spastic CP gait patterns, showing the variation of α as the angle between the forefoot and the shanks (Armand et al., 2016).

2.2.3. Physiotherapy techniques

Among the most famous or notorious therapies available for children with CP, is the Constraint-Induced Movement Therapy (CIMT), the Bimanual training, electrical stimulation and hippotherapy (Physiopedia, 2024). Nevertheless, little has been said or researched about Vojta Therapy in cerebral palsy.

2.2.4. Vojta Therapy

Vojta therapy is distinguished by its application of pressure to certain areas of the body while the patient is in a supine position. This approach differs from other methods since it facilitates access to the individual patterns of motion for a particular movement through the therapeutic activation of specific places of the body. (Internationale Vojta Gesellschaft e.V., 2024)

2.3. Inertial sensors

Inertial sensors, alternatively known as inertial measurement units (IMU), are a group of sensors that permit the estimation of acceleration, force and angular velocity. Such measurements are referred to as inertial measurements of an object or body and are made with respect to a frame of reference.

Nevertheless, contemporary IMUs incorporate non-inertial sensors such as magnetometers or barometers, which are designed to measure the surrounding magnetic field and atmospheric pressure, rather than the body's inertial measurements. However, both sensors provide additional information that assists the sensor fusion process, thereby avoiding drift.

In this section the most commonly used sensors on the inertial measurements units will be presented and discussed.

2.3.1. Accelerometer

Operating principles

An accelerometer is a device that provides one dimensional linear acceleration based on the mechanical principle of the moving proof mass. This mass is constrained to a single axis and held by springs, which is actuated by a given accelerating force. Therefore, the final displacement of the mass will be proportional to the acceleration applied to the device case (see Figure 2.8).

However, an important factor to consider in this sensor is the gravitational force, which in this case is exerted directly into the proof mass. In the absence of non-gravitational acceleration forces, in other words when the sensor is in a static state, the accelerometer is capable of measuring the effect of the gravity as weight and, thus, resulting in a measured acceleration opposite to the effect of the gravity on its axis (Groves, 2015)

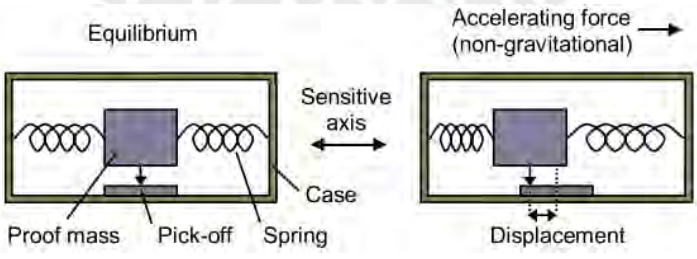


Figure 2.8 Mechanical principle of the accelerometer (Groves, 2015).

Technology

Nowadays, these sensors use the mechanical principle explained above and use Micro-electro-mechanical systems (MEMS) technology. This technology allows smaller devices and low-cost silicon or quartz sensors while maintaining the desired performance. (Groves, 2015)

2.3.2. Gyroscope

A diversity of gyroscopes exists, such as mechanical or optical, however, their performance does not meet the market expectations, and they tend to be expensive. In spite of this, the vibratory MEMS gyroscopes offer a superior performance-to-cost ratio and thus represent an advantageous solution. The fabrication of these devices is mostly conducted using silicon micro-machining techniques, which enables the housing of the three axes of the gyroscope in a confined space and results in the reduction of manufacturing costs. (Woodman, 2007)

Operating principles

The operational principle of these devices is given by the Coriolis effect, which enables the measurement of the angular velocity. This is achieved by the vibrating elements present on the sensor and the resonating mass. Figure 2.9 provides a simple example of this working principle. The mass *m* represents a resonating mass which, when rotated, generates a velocity *v* in the drive axis and a *Fc* force in the sense axis. (Woodman, 2007)

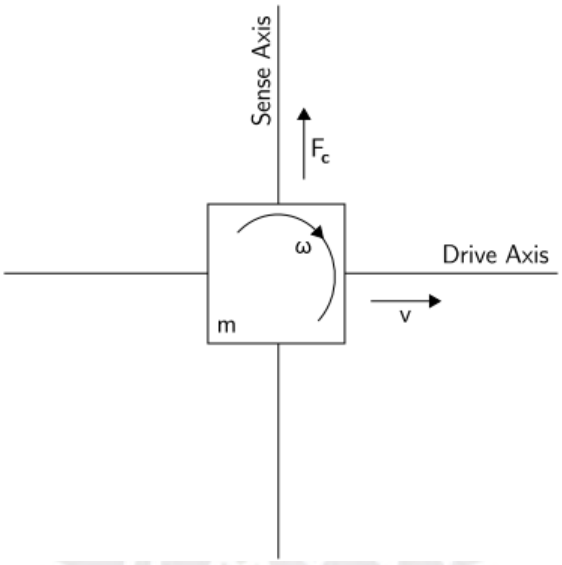


Figure 2.9 Operational principle of MEMS gyroscopes, where m is the mass, v is the drive axis velocity, ω is the angular velocity and F_c is the sensed force (Woodman, 2007).

2.3.3. Magnetometer

This type of device typically comprises a set of three magnetometers, which measure the Earth's magnetic field magnitude and direction in the three distinct and orthogonal axes. This geomagnetic field is self-sustaining and exhibits a magnetic dipole-like configuration, although with a subtle tilt in both north and south poles as illustrated in Figure 2.10 (VectorNav, 2024b).

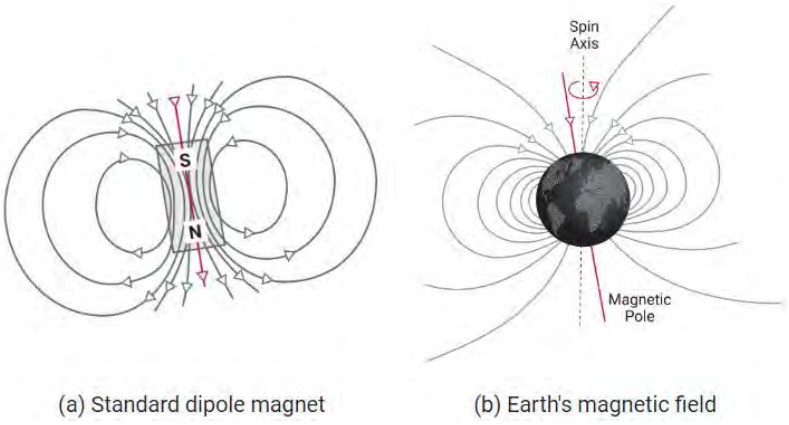


Figure 2.10 Resemblance of the Earth's magnetic field with a dipole magnet (VectorNav, 2024b).

Operating principles

The most used MEMS types of magnetometers are the Hall-Effect and magnetoresistive. The Hall sensor is designed to quantify only the magnetic flux perpendicular to the sensing plate, mostly made of FeNi alloy, as illustrated in Figure 2.11a. In order to measure the planar fluctuations, the magnetic field rotation is accomplished through the use of integrated magnetic concentrators. (Fraden, 2016)

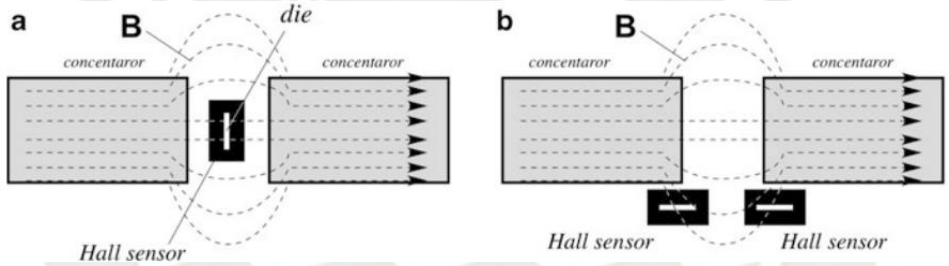


Figure 2.11 (a) Perpendicular Hall sensor. (b) Array of Hall sensors in an integrated magnetic concentrators (Fraden, 2016).

On the other hand, magnetoresistive sensors operate on a completely different principle, since a permanent magnet is an integral component of the sensor (see Figure 2.12).

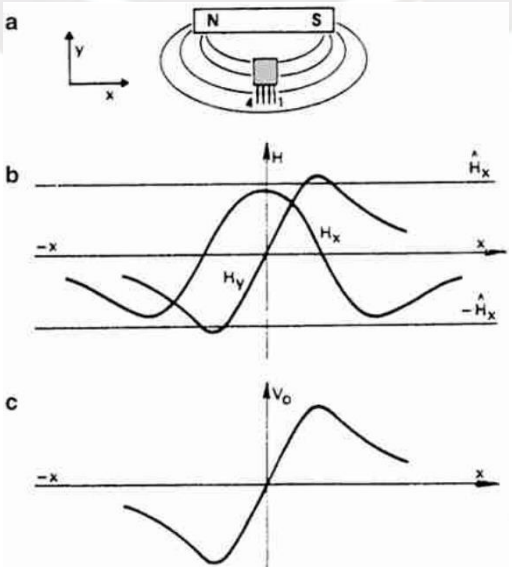


Figure 2.12 (a) Sensor attached with permanent magnet, (b) Magnetic vectors generated, (c) Output magnetic vector (Fraden, 2016).

Consequently, when the sensor is situated within under an external magnetic field (Figure 2.12a), it generates the magnetic vectors H_x and H_y which are functions of the sensor's displacement x, considering the external magnetic field influence (Figure 2.12b). If the magnetometer is oriented parallel to the displacement axis (x-axis in this case), then H_x provides an supplementary magnetic field, while H_y provides the output signal (Figure 2.12c), which resembles to the displacement on x-axis. (Fraden, 2016)

Data processing

Magnetometer data is widely used to calculate the inclination or the heading of the whole IMU sensor. Since the accelerometer data enables the roll and pitch angles calculation, with the geomagnetic field measured by this device the estimation of the heading or yaw angle can be obtained by the Eq. 2.5 adapted from Groves (2015).

$$\tilde{\psi}_m = \arctan_2\left(\frac{\tilde{m}_y}{\tilde{m}_x}\right) = \arctan_2\left(\frac{\tilde{m}_y \cos\hat{\phi} - \tilde{m}_z \sin\hat{\phi}}{\tilde{m}_x \cos\hat{\theta} + \tilde{m}_y \sin\hat{\phi} \sin\hat{\theta} + \tilde{m}_z \cos\hat{\phi} \sin\hat{\theta}}\right) \qquad \textit{(Eq. 2.5)}$$

where $\tilde{m}_{x,y,z}$ represents the measured magnetic field by the magnetometer, with the measurement taken with respect to the three axes.

Measurement errors

The data provided by this type of sensor is subjected to several sources of error given by the presence of external ferromagnetic materials like electronic devices, vehicles, power devices and so on. Consequently, there is a presence of the hard- and soft-iron magnetism effect on the device (Groves, 2015).

The effects in question relate to the magnetic field generated by magnets or electrical equipment, which create an offset of the Earth's magnetic field in its true state, as well as the distortion or variation of the field produced by certain materials, such as iron or nickel. Since both effects distort the real data, hard and soft iron calibration must be carried out. This calibration process consists of three-dimensional measurements through the shaping of a sphere in the designated working environment, moving and rotating the sensor in order to obtain an extensive quantity of data from all axes. (VectorNav, 2024a)

2.4. Inertial data fusion algorithms

The orientation estimation based on signals from a single sensor, such as accelerometer, gyroscope or magnetometer, is prone to drift and therefore may provide inaccurate results with significant errors.

Consequently, over the years, this problem has been the subject of research, resulting in the development of numerous algorithms as potential solutions. Among these algorithms

are the Madgwick filter, Complementary filter (CF) and Kalman filter (KF). This thesis will present the two remaining methods, information about the Madgwick filter can be found in Madgwick et al. (2011).

2.4.1. Complementary filter

Each of the three sensors under consideration has advantages and disadvantages. The estimated orientation from the accelerometer and magnetometer data tends to be more accurate over shorter time periods, whereas the orientation calculated from gyroscope data is more accurate over longer time periods.

Furthermore, the orientation derived from acceleration and magnetic field data exhibit desirable properties at low frequencies, whereas angular velocity demonstrates favorable properties at high frequencies. In addition, gyroscope orientation estimations tend to exhibit greater drift than other sensor data estimates, while accelerometer and magnetometer orientation is more susceptible to noise. (Kok et al., 2017)

To overcome the disadvantageous properties of the sensors, the combination of two filters should be used, a low-pass $G(s)$ and a high-pass filter $1 - G(s)$. Then, the Laplace transform of the complementary filter to compute the orientation $\Theta(s)$ is

$$\Theta(s) = G(s)\Theta_c(s) + \big(1 - G(s)\big)\Theta_g(s) \qquad \textit{(Eq. 2.6)}$$

$$\Theta(s) = G(s)\Theta_c(s) + \big(1 - G(s)\big)\frac{1}{s}\omega_g(s) \qquad \textit{(Eq. 2.7)}$$

where $\Theta_c(s)$ and $\Theta_g(s)$ are the Laplace transforms of the orientation estimates calculated by the acceleration and magnetic field data, and by the gyroscope, respectively.

Then, the complementary filter formula in discrete time, after doing a backward discretization considering the low-pass filter as one with first order, is represented by:

$$\hat{\theta}(t) = (1 - \gamma)\theta_c(t) + \gamma\theta_g(t) \qquad \textit{(Eq. 2.8)}$$

where γ is the tuning parameter of this filter. As this parameter approaches one, the contribution of the gyroscope data $\theta_g(t)$ increases. Conversely, when it is close to zero, the contribution of the accelerometer and magnetometer data $\theta_c(t)$ is the greatest. Additionally, this $\theta_c(t)$ can be calculated through trigonometric functions while $\theta_g(t)$ is calculated by Eq. 2.9.

$$\theta_g(t) = \hat{\theta}(t-1) + \theta_g(t-1)dt \qquad \textit{(Eq. 2.9)}$$

2.4.2. Kalman filter

The Kalman filter (KF) offers a real-time estimation of specific parameters within a linear system, incorporating both the prior state of the system and the update of the actual state.

The filter is comprised of five fundamental elements: the system model, the state vector and its covariance, the measurement vector and its covariance, the measurement system model, and the algorithmic procedure itself (Groves, 2015).

There are many types of Kalman filter, however, the most common are Extended Kalman filter (EKF) and Unscented Kalman filter (UKF).

Extended Kalman Filter

The EKF is known for its capability to linearize nonlinear systems, something that the standard KF lacks, as it is designed for estimating linear systems only (Baghdadi et al., 2018). The model can be represented by Eq. 2.10 – 2.11.

$$\dot{\mathrm{x}}(t) = \mathrm{f}(\mathrm{x}(t), t) + G(t)\mathrm{w}(t),\ \mathrm{w}(t) \sim \mathcal{N}\big(0, Q(t)\big) \qquad \textit{(Eq. 2.10)}$$

$$\tilde{\mathrm{y}}_k = \mathrm{h}(\mathrm{x}_k) + \mathrm{v}_k,\ \mathrm{v}(t) \sim \mathcal{N}(0, R_k) \qquad \textit{(Eq. 2.11)}$$

where $\mathrm{f}(\mathrm{x}(t), t)$ is the state model continuous and differentiable function, $G(t)$ is the process noise state model, x is the state vector, h is the measurement model, $\mathrm{w}(t)$ and $\mathrm{v}(t)$ are white Gaussian measurement and system noises, respectively. Additionally, $Q(t)$ is the spectral density and R_k is the covariance. (Baghdadi et al., 2018)

Since EKF does the differentiation of the state and the measurement model, then, the state transition model $F(t)$ and observed model H_k can be obtained by:

$$F(\hat{\mathrm{x}}(t), t) = \frac{\partial[\mathrm{f}(\mathrm{x}(t), t)]}{\partial[\mathrm{x}(t)]}|_{\hat{\mathrm{x}}(t)} \qquad \textit{(Eq. 2.12)}$$

$$H_k(\hat{\mathrm{x}}_k^-) = \frac{\partial \mathrm{h}}{\partial \mathrm{x}}|_{\hat{\mathrm{x}}_k^-} \qquad \textit{(Eq. 2.13)}$$

Unscented Kalman Filter

The UKF was created to overcome the instability generated by the linearization of nonlinear models. Then, instead of calculating the Jacobian, the UKF estimates sigma points and these points represent the state of the system mean and covariance with a Gaussian distribution. (Baghdadi et al., 2018)

2.4.3. Filter comparison

Kok et al. (2017) have evaluated the CF and EKF algorithms with different parameters. The following Table 3.2 presents the results of an analysis of both algorithms with different parameters, conducted using a Monte Carlo simulation.

Table 2.2 Comparative table of the mean RMSE from sensor fusion algorithms (Kok et al., 2017).

Algorithm	Roll (°)	Pitch (°)	Heading (°)
Complementary filter ($\alpha = 0.07$)	1.44	1.43	4.39
Complementary filter ($\alpha = 0.7$)	0.47	0.47	**12.98**
EKF quaternions	0.45	0.45	3.57
EKF orientation deviation	0.45	0.45	3.55

The table displays the mean root mean square error (RMSE) for each algorithm. At plain sight, the best algorithm to use is EKF with quaternions or orientation deviation. In the case of the CF, the parameter can be compared to the tuning parameter presented above as $\alpha = (1 - \gamma)\beta$, where β is a step length constant for the Gauss-Newton Optimization. In order to obtain a deeper understanding of this constant, it is important to note that it has an inverse proportion to the γ parameter. Consequently, a low value of α is associated with a high value of γ.

2.5. Digital filters

Digital filters are of significant importance due to their ability to attenuate frequencies. Figure 2.13 presents the most common filters (Smith, 1999). Low-pass and high-pass filters serve to filter and pass, as implied by their designation, low and high frequencies, respectively, within a specified cutoff frequency. In contrast, band-pass filters (Figure 2.13c) allow the transmission of frequencies within a specified range, whereas band-reject filters (Figure 2.13d) block the transmission of a range of frequencies.

The most used filters in gait analysis research use low-pass or band-pass filters, mainly using Butterworth filters and a different type of filter, known as the Moving average filter. Thus, these filters will be presented to provide information for a better understanding and selection of filters on the preprocessing of the data in this work.

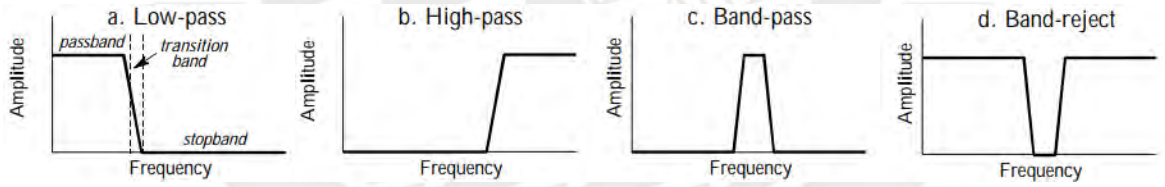


Figure 2.13. Commonly used filters (Smith, 1999).

2.5.1. Butterworth filter

This type of filter has a monotonic frequency response, and it is characterized by its sharpness when shifting from the desired cutoff frequency to the rejected frequency. This last factor is given by the order N of the filter, as can be illustrated on Figure 2.14.

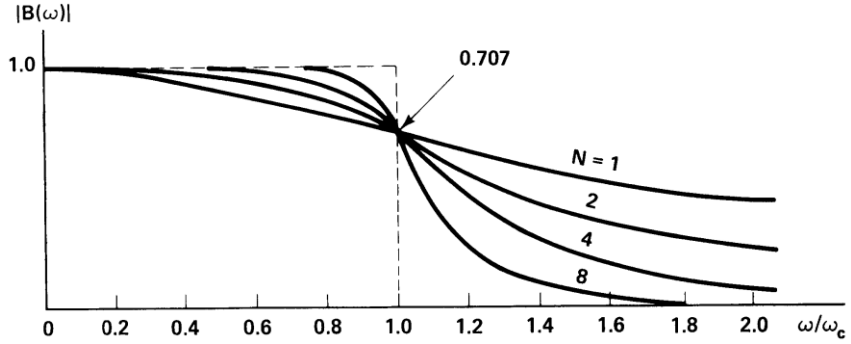


Figure 2.14. Low-pass Butterworth filter frequency response (Oppenheim, 2011).

This filter has only two parameters: the order and the cutoff frequency ω_c. The squared frequency response $|B(j\omega)|^2$ is given by the following formula. (Oppenheim, 2011)

$$|B(j\omega)|^2 = \frac{1}{1 + \left(\frac{j\omega}{j\omega_c}\right)^{2N}} \quad \textit{(Eq. 2.14)}$$

2.5.2. Moving average filter

This filter is known for averaging a certain number of points from a signal to generate a smooth output signal. It is ruled by the following equation (Smith, 1999).

$$y[i] = \frac{1}{M}\sum_{j=0}^{M-1} x[j-i] \quad \textit{(Eq. 2.15)}$$

where M is the number of points to consider of the averaging procedure, x is the input and y is the output signal.

3. State of technology

3.1. State of the art of IMUs

Table 3.1 summarizes the most common IMUs used in industrial or research applications, including the sensors used and their degrees of freedom (DOF).

Table 3.1 List of commercial IMUs.

IMU	DOF	Sensors
MPU6050	6	Triaxial accelerometer Triaxial gyroscope
FreeSense	6	Triaxial accelerometer Triaxial gyroscope
MVP-RF8-GC	6	Triaxial accelerometer Triaxial gyroscope
MPU9250	9	Triaxial accelerometer Triaxial gyroscope Triaxial magnetometer
XSens	9	Triaxial accelerometer Triaxial gyroscope Triaxial magnetometer
Trigno Avanti	10	Triaxial accelerometer Triaxial gyroscope Triaxial magnetometer EMG
APDM Opal	10	Triaxial accelerometer Triaxial gyroscope Triaxial magnetometer Barometer
Physilog4	10	Triaxial accelerometer Triaxial gyroscope Triaxial magnetometer Barometer
NGIMU	10	Triaxial accelerometer Triaxial gyroscope Triaxial magnetometer Barometer

3.2. Gait analysis

Almost all the gait analyses conducted on children with CP these days are centered on the estimation of gait parameters derived from feet data. However, recent investigations have begun to analyze the upper body in order to obtain gait information data from this region, due to the compensatory movements caused by the trunk on CP gait. For that reason, some of these investigations are presented in the following sections. Section 3.2.6 provides an overview with multiplexed information about the studies, such as type of sensors user, number of participants, and so on.

3.2.1. Analyses of the whole body in CP

Paraschiv-Ionescu et al. (2019) carried out an experiment consisting of a 10-meter walking test (10mWT) indoors and daily-life activities outdoors of the Geneva University

Hospitals, with 15 children with CP with 11 age matching TD-children. Six sensors were located on the children's sternum, lumbar (L5 vertebrae), thighs and shanks.

They developed a step or locomotion detection algorithm based on the trunk acceleration norm, either sternum or lumbar sensor, enhancing and detecting peaks that correspond to the heel-strike events and selecting the steps corresponding to a gait segment.

With these gait cycles, parameters such as spatiotemporal, attenuation coefficient (AC), root mean square (RMS) acceleration were calculated. As a result, they determined that the AC showed more variability in children with CP and a GMFCS of II and III, in comparison to the children with CP GMFCS I. Regarding the algorithm, it showed a similar performance for both trunk sensors, having as a ground truth the shanks sensor data.

Carcreff et al. (2020) performed a 10-meter walk test and gait monitoring for three days of ten hours each. Sensors were placed on the chest, thighs and shanks of 14 children with CP and 14 matching TD-children. The processing centered on the selection of walking segments from the monitoring test and gait characterization.

The involved evaluated domains were rhythm, pace, knee angle amplitude, stability, coordination, smoothness, variability and asymmetry. The estimated parameters from the mentioned domains were computed from the spatiotemporal parameters. Consequently, they estimated that the pace and variability ranges significantly varied depending on where the test was conducted: in the laboratory or elsewhere. This variability was greater and the pace slower when the gait was performed in daily life activities. A comprehensive overview of these findings is provided in Figure 3.1.

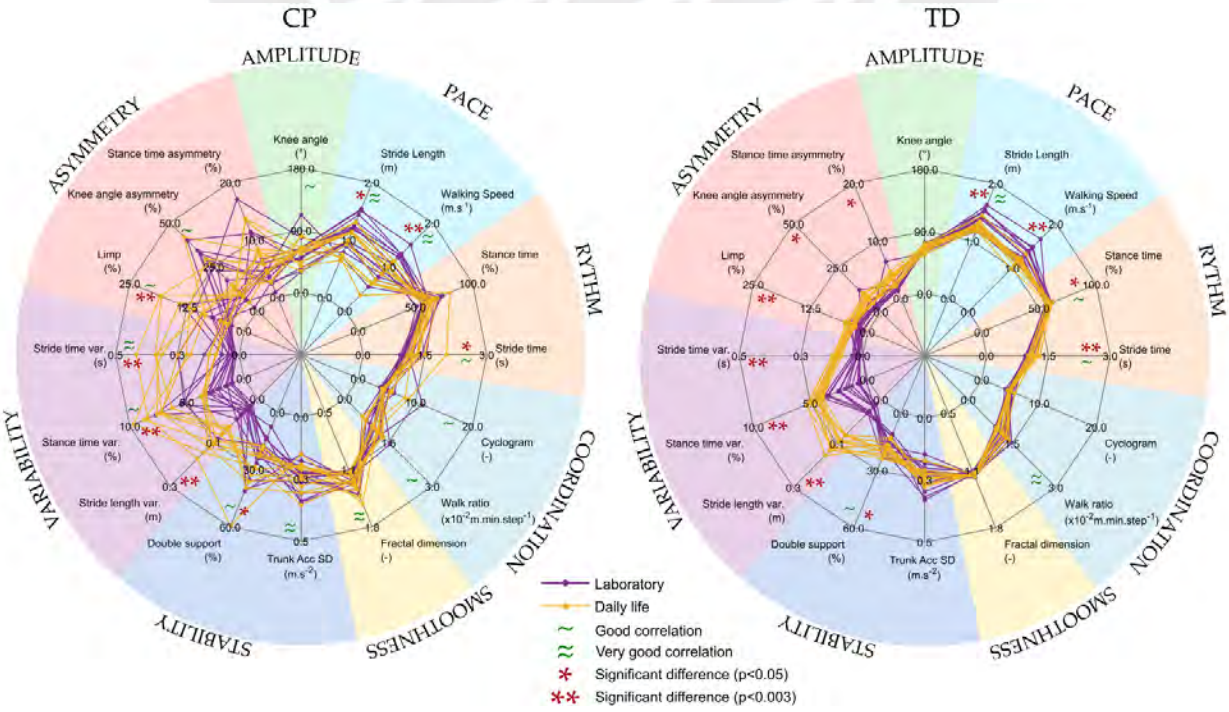


Figure 3.1 Summary of results from processed data in radar plots (Carcreff et al., 2020).

Furthermore, children with CP had increased asymmetry and rhythm parameters compared to their matched TD-children. The double support phase was increased in children with CP and no difference was found when it came to the other domains.

3.2.2. Analyses of the upper body in CP

Few researchers have focused on examining the performance of the spine or the upper body on CP gait analysis with IMUs. The previously mentioned Paraschiv-Ionescu et al. (2019) did that by comparing the gait not only on standard walking tests, but also with considering other sequence of daily-life activities.

Contini et al. (2019) conducted a 10-meter walking test on ten children with diplegic CP. This experiment focused on three measurements in different conditions: barefoot, after 20 days using a hinged ankle-foot orthosis (hAFO) and after 20 days using a solid ankle-foot orthosis (sAFO). Each participant had an attached sensor at sternum, near L3 vertebrae and ankles. The parameters evaluated were spatiotemporal, gait stability and symmetry.

The spatiotemporal parameters that were evaluated in this research included gait speed, step length and cadence. The range of motion (ROM) of the pelvic tilt and the trunk lateral bending, the RMS of AP pelvic and trunk ML acceleration, and the trunk's attenuation coefficient as stability measurements. Finally, the gait symmetry is measured with the improved harmonic ratio (iHR) of anteroposterior (AP) and mediolateral (ML) axes. As results they obtained that after the use of hAFOs reported a little increase on the ROM of the trunk. Additionally, after the use of sAFOs the average number of step lengths and the attenuation coefficient increased significantly.

On the other hand, Summa et al. (2016) developed a 10-meter walking test involving 20 children with CP and their matched TD-children. They focused mainly on the upper body, in comparison to the other two mentioned studies, which measured also the lower body with sensors placed on the thighs, shanks or ankles. This study had sensors placed on the head, sternum and lumbar spine (near L5 vertebrae).

This investigation focused on evaluating and computing the average of the spatiotemporal parameters, RMS acceleration and the attenuation coefficient (AC). Therefore, as result, there was no noticeable difference in walking speed but a significant smaller step length and greater step frequency between CP and TD groups. There was a higher difference in the acceleration RMS for all the axes comparing the CP to the TD children. The AC from the pelvis to sternum (AC_{PS}), the pelvis to head (AC_{PH}) and the

sternum to the head (AC_{SH}) were also calculated and resulted on a larger AC_{PS} and smaller AC_{SH} on CP children compared to the TD.

Finally, Sivarajah et al. (2018) conducted a feasibility study with 15 children with spina bifida (SB) or CP and 15 matched TD-children. The study consisted of a 10-meter walk test, a Timed Up and Go (TUG) test, an obstacle test and the Curb test. The participants had sensors placed on the upper chest, lower back and one per wrist and per ankle.

To conclude with, this research proved discrimination in both types of tests and between both SB and CP groups, recognizing the high-functioning walkers from each group. The spatiotemporal parameters corroborate a significant difference on the frontal and horizontal ROM and walking velocity during walking tests and the 180° turn duration during the TUG test. Additionally, there were observed some differences in limb asymmetry between both TD and SB/CP groups.

3.2.3. Analyses of the trunk in TD

Furthermore, it is essential to gather information about the current gait analysis studies conducted on healthy adults. Therefore, this section will provide some of the research findings, giving insight into recent developments in gait parameters.

Digo et al. (2019) conducted a six-meter walking test five times, placing three IMUs on the cervical (C7 vertebrae), thoracic (T12 vertebrae) and sacrum (S1 vertebrae) of a young healthy adult. The main objective of the pilot study was to validate the assessment of spinal angles using the tilt-twist method.

To ascertain this, the researchers established mathematical formulas that enable the estimation of the tilt (φ), tilt-azimuth (θ) and twist (τ) angles from one coordinate frame with respect to another. The raw rotation matrices obtained from the sensors were used for the calculation of relative rotation matrices between each spine segment through matrix calculations.

This also permitted the estimation of tilt-twist angles and then, the calculation of relative flexion/extension (FE), lateral bending (LB) and axial rotation (AR) from every sensor. This method is interesting since it gives three relative angles that are not sequential, like Euler or Cardan angles are. Nevertheless, the data must be related from one sensor to another in order to provide relative angles.

Furthermore, the investigation also employed optoelectronic systems to validate the performance and efficacy of the method on inertial sensors. As a result, the comparison of the two measuring systems delivered a range of difference ROM values between 0.2° and 1.1°, which were demonstrated to be within an acceptable range.

In their subsequent study, Digo et al. (2020) developed a study explicitly focused on evaluating the performance of the trunk posture during three types of gaits: normal, slow and self-selected speed. For this, 18 young and healthy adults were recruited, who had to use five IMU sensors distributed on the spine. These sensors were placed on the cervical (C7 vertebrae), thoracic (T6 vertebrae), lower thoracic (T12 vertebrae), lumbar (L3 vertebrae) and sacrum (S1 vertebrae). They utilized the previously tested tilt-twist method, whose angles φ, θ, τ allow the calculation of FE, LB and AR. Figure 3.2 provides more detailed information about the tilt-twist angles between two coordinate systems.

As a result, they observed that this methodology was suitable for spinal posture evaluation during different types of gaits. This new method effectively identifies a pattern of trunk compensation during gait, providing a valuable indicator of efficiency in both clinical and rehabilitative settings.

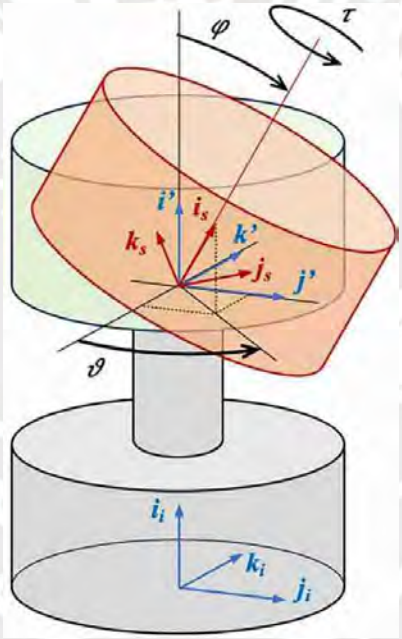


Figure 3.2 Tilt, tilt-azimuth and twist angles of the upper coordinate system with respect to the lower coordinate system (Digo et al., 2020).

3.2.4. Analyses of the lumbar spine in CP

In this section, the investigations focused exclusively on getting the gait parameters using the lumbar spine data are presented.

Iosa et al. (2012; 2013) were two of the first investigations regarding gait analysis using IMU sensors in CP children. They performed a 10-meter walking test on 17 children with spastic hemiplegic CP and their matched TD-children in the first study, and a 10-meter walking or running test on 20 children with CP and their matched 20 TD-children in the second study. Both groups of participants had one sensor placed on the lumbar spine (L2-L3 vertebrae).

Gait stability parameters were the RMS acceleration, the minimum peak of accelerations (a_{min}) and the peak-to-peak value of angular velocities (PP_v). The RMS represented the dispersion of the acceleration, a_{min} described the abrupt decelerations that affected the conservation of momentum during walking; and PP_v measured the generated rotation affecting the linear momentum.

Gait harmony parameters were the Harmonic Ratio (HR), the Ratio Index (RI) of a_{min} and of PP_v. The first parameter indicated how smooth and rhythmic the acceleration patterns were. Meanwhile, both ratios indicated asymmetries in the decelerations in absolute values of each step and each stride, respectively.

As a result of this, Iosa et al. (2012) concluded that there was no meaningful difference in the mean walking speed of both CP and TD groups. However, there was a clear reduction in both gait stability and harmony in at least one axis. They also found out that CP children had higher negative a_{min} along the AP and VT axes and per stride in the ML axis, despite the reduced walking speed.

Additionally to the mentioned parameters, Iosa et al. (2013) determined Gait ability parameters. Those related to spatiotemporal parameters essential for walking and running comparison, such as walking speed, step length and step duration.

Furthermore, Iosa et al. (2013) concluded that the mean running speed was more than twice the mean walking speed, leading to a higher step length and lower step time during the running tests. Gait smoothness and rhythmicity as measured by HR and RI were significant for each one of the groups evaluated. However, the type of movement only affected the RMS acceleration for all, with no significant difference between groups.

Sæther et al. (2014) conducted a 5-meter walking test on 41 children with spastic CP and 29 TD-children, with an IMU sensor place on the lumbar spine (L3 vertebrae). Involved parameters were spatiotemporal variables, RMS acceleration, interstep and interstride regularity and trunk asymmetry.

The results obtained involved no significant difference on the spatiotemporal parameters, greater mean accelerations and more asymmetry along all axes for children with CP, and interstride regularity was notably lower only on the AP axis for children with CP. They also concluded that age produced a noticeable change in stride regularity on the AP axis as well. Significant variations in asymmetry were observed between children with CP GMFCS I level and TD-children, but there was no contrast between children with different GMFCS levels. Lastly, it was observed higher accelerations in unilateral CP only on VT axis and along the three axes for bilateral CP in comparison with the TD group.

Their subsequent investigation, Sæther et al. (2015), focused on the assessment of trunk control in sitting with the Trunk Impairment Scale (TIS) and the Trunk Control Measurement Scale (TCMS), additional to the trunk control during walking with a 5-meter walking test. Participants involved were 26 children with spastic CP who carried an IMU on the lumbar spine (L3 vertebrae). The estimated parameters were the same as in the previous studies in addition to the trunk scale scores. They observed that trunk control during sitting was strongly associated with gait AP acceleration and regularity, concluding on the higher importance of the trunk during the gait.

In addition, Tramontano et al. (2017) concentrated on analyzing the lumbar spine (L2-L3 vertebrae) in a group of 80 people: 15 children with CP, 10 TD-children, 20 healthy young adults, 20 healthy elderly and 20 adults with stroke. To do this they performed a 7-meter walking test during dual-task, such as avoiding obstacles or carrying objects. The calculated spatiotemporal parameters and RMS accelerations during the gait. This resulted in a noticeably reduction in walking speed and a significant increase of RMS acceleration while performing dual tasks in comparison to a single task for all groups.

Chen et al. (2017) evaluated the gait of 22 children with CP, 10 TD-children and 14 healthy adults using a 30-strides gait test. The IMUs were placed on the lumbar spine (L2-L3 vertebrae) and thighs. In this case, the lumbar spine information was employed to find the following parameters: Pearson coefficient, variance ratio, extreme points number, harmonic ratio and symmetry. In order to analyze the acceleration pattern data, the grey relational analysis method was used. Figure 3.3 portrays the results of the gait parameters and estimated score for all groups.

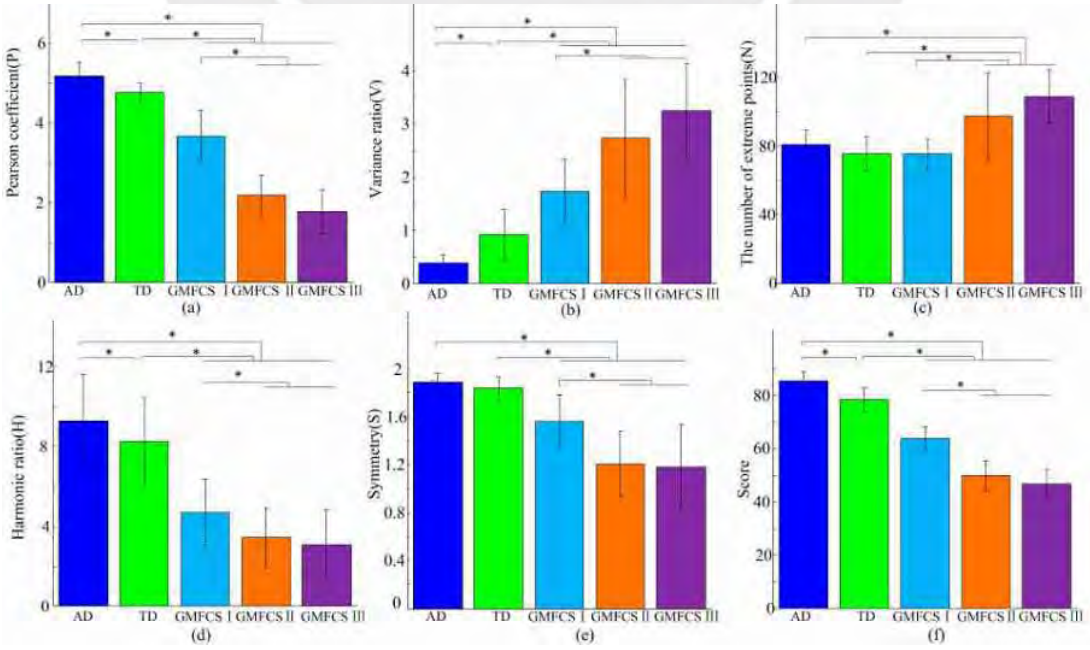


Figure 3.3 Results on the five gait parameters and the final score (Chen et al., 2017).

They focused on the comparison of the mean accelerations per patient representing the data in Average Gait Graph (AGG) and Characteristic Gait Graph (CGG). Therefore, they compared the CGG of healthy adults with the AGG of children with CP with different levels of GMFCS, finding out not only the difference on the gait pattern but also the variance in the gait parameter values and the gait score, as shown on the figure above. As a conclusion, this algorithm was able to successfully detect and evaluated abnormal gait in children with CP.

Piitulainen et al. (2021) conducted a 10-meter walking tests for three tasks: normal gait, motor dual task and cognitive dual task. The participants were 30 children and adolescents with CP, 18 with spastic hemiplegia (HP) and 12 with diplegia (DP) and 31 matched TD-subjects, which had an IMU placed on the lumbar spine (L3-L5 vertebrae). The parameters evaluated were the spatiotemporal and gait complexity.

This investigation resulted in a lower walking speed in DP in comparison to HP and TD groups. The step duration was similar for all groups but had more variance on the HP group for all the tasks involved. Additionally, the gait complexity was higher in the CP group compared to the TD group. Regarding the comparison per task, the step duration was greater for both dual tasks in comparison to normal gait.

Other investigations that focused on the performance on the lumbar spine were Mutoh et al. (2016; 2018), which will be presented on the Section 3.2.5.

3.2.5. Pre- and post-treatment analysis of CP

One of the main objectives of this work is to conduct comparative analysis of gait data collected before and after therapy sessions. A comprehensive review of the existing literature revealed that only two investigations are focused on this comparison, with a distinct focus on the use of hippotherapy (Mutoh et al., 2016; Mutoh et al., 2018). These investigations were conducted before and after the administration of hippotherapy sessions of 30 minutes on two children and an adult with CP in the first study, and on 20 children and adolescents with CP in the subsequent study.

An IMU sensor was placed at the lumbar region (near L3 vertebrae). Subsequently, a 10-meter walking test was conducted without any assistance in the initial study, and in the second study, participants were permitted to use orthoses if they desired. The parameters that were evaluated are common spatiotemporal parameters, mean accelerations, and the horizontal/vertical displacement ratio.

Mutoh et al. (2016) yielded evidence of a notable increase in step length was observed, with an improvement on the horizontal and vertical displacement, as observed

qualitatively following a single hippotherapy session. After the two years of hippotherapy treatments an improvement in the GMFCS-level was acknowledged. In addition, there was a significant increase in step rate and length, gait velocity and mean accelerations. On the other hand, Mutoh et al. (2018) registered a faster gait speed after a six week period of therapy, related to the stride length or mean acceleration increase. Since the twelve-week mark, the horizontal/vertical ratio has exhibited an observed increase, almost to normal levels. And over the course of the initial twelve-month period, parameters such as cadence, gait speed, stride length and mean acceleration were increasing steadily.

3.2.6. Research overview

A fundamental preliminary step in the software development process is the comprehensive review of previous research that has employed comparable data-gathering methods. In Table 3.2, some studies are presented that used IMU sensors primarily attached to the sternum and lumbar regions of participants and could walk independently.

Table 3.2 Summary of similar investigations (part 1).

Research	Sensor	Location	Participants	Tests	Parameters	Results
Iosa et al., 2012	FreeSense Sensorize	L2 – L3	17 CP-children with spastic hemiplegia 17 matched TD-children	10mWT	RMS Acceleration, a_{min}, PP_v, HR, RI of a_{min}, RI of PP_v	Clear reduction in gait stability and harmony in at least one axis. CP children had higher negative a_{min} along the AP and VT axes and per stride in the ML axis.
Iosa et al., 2013	FreeSense Sensorize	L2 – L3	20 CP-children 20 matched TD-children	10mWT	RMS Acceleration, a_{min}, PP_v, HR, RI of a_{min}, RI of PP_v	Higher step length and lower step time on the running tests. Gait smoothness (HR) for running increased in CP and decreased in TD in comparison to walking. The type of movement only affected the RMS acceleration for all, with no significant difference between groups.
Sæther et al., 2014	Xsens Movelia	L3	41 spastic CP-children 29 TD-children	5mWT	Spatiotemporal, RMS acceleration, interstep and interstride regularity, trunk asymmetry	Age produced a noticeable difference in stride regularity along AP axis. Significant variations on asymmetry were observed between CP with GMFCS I level and TD. Higher VT accelerations in unilateral CP and along the three axes for bilateral CP.
Sæther et al., 2015	Xsens Movelia	L3	26 spastic CP-children	5mWT and trunk control in sitting with TIS and TCMS	Spatiotemporal, RMS acceleration, interstep and interstride regularity	Trunk control during sitting was strongly correlated with AP acceleration and regularity, leading to a higher importance of the trunk during the gait.
Summa et al., 2016	APDM Opal	Head Sternum L5	20 CP-children 20 matched TD-children	10mWT	Spatiotemporal, RMS acceleration, attenuation coefficient (AC)	Higher difference on all the acceleration RMS between CP and TD children. Larger AC_{PS} and smaller AC_{SH} on CP children compared to the TD.
Mutoh et al., 2016	MVP-RF8-GC	L3	2 CP-children and a CP-adult	10mWT before and after hippotherapy	Spatiotemporal, mean accelerations, and horizontal/ vertical displacement ratio	Improvement in the GMFCS level, significant increase in step rate and length, gait velocity and mean accelerations after therapy.
Tramontano et al., 2017	Not specified	L2 – L3	15 CP-children 10 TD-children 20 healthy young adults 20 healthy elderly 20 adults with stroke	7mWT with dual task	Spatiotemporal, RMS acceleration	Noticeably reduction of walking speed and significant increase of RMS acceleration while performing dual task in comparison to a single task for all groups.
Chen et al., 2017	MPU9250 InvenSense	L2 – L3 Thighs	22 CP-children 10 TD-children 14 healthy adults	30-strides gait test	Acceleration, Pearson coefficient, variance ratio, extreme points number, HR and symmetry	Difference on gait pattern, gait parameter values and gait score. The algorithm was able to successfully detect and evaluated abnormal gait in CP-children.

Table 3.2 Summary of similar investigations (part 2).

Research	Sensor	Location	Participants	Tests	Parameters	Results
Mutoh et al., 2018	MVP-RF8-GC	L3	20 CP-children and adolescents	10mWT before and after hippotherapy	Spatiotemporal, mean accelerations, and horizontal/ vertical displacement ratio	Faster gait speed, stride length or mean acceleration after a 6 week period of therapy. After 12 weeks, the horizontal/vertical ratio increased almost to normal levels. In the initial 12 months, cadence, gait speed, stride length and mean acceleration increasing steadily.
Sivarajah et al., 2018	APDM Opal	L3 Ankles	15 SB- or CP-children 15 matched TD-children	10mWT, TUG test, obstacles test and Curb test	Spatiotemporal, trunk ROM	Significant difference on the frontal and horizontal ROM and walking velocity during 10mWT and TUG test. Limb asymmetry differences between both TD and SB/CP groups.
Paraschiv-Ionescu et al., 2019	Physilog4 GaitUp	Sternum L5 Thighs Shanks	15 CP-children 11 matched TD-child	10mWT	Spatiotemporal, RMS acceleration, attenuation coefficient (AC)	More variability on the AC in CP-children with a GMFCS of II and III, in comparison to the GMFCS I group. Step detection algorithm showed a similar performance for both trunk sensors.
Contini et al. 2019	APDM Opal	Sternum L3 Ankles	10 diplegic CP-children	10mWT on three conditions: 1. barefoot 2. after 20 days using a hAFO 3. after 20 days using a sAFO	Spatiotemporal, gait stability and symmetry, attenuation coefficient (AC)	The use of hAFOs generated a small increase on the trunk ROM. After the use of sAFOs the average number of step length and the AC increased significantly.
Carcreff et al., 2020	Physilog4 GaitUp	Chest Thighs Shanks	14 CP-children 14 matched TD-children	10mWT	Spatiotemporal, walk ratio, smoothness, gait variability, symmetry	Pace and variability significantly varied depending on where the test location. The pace was slower when the gait was performed in daily life activities. CP-children had increased rhythm and asymmetry parameters in comparison to TD-children. The double support phase time was increased in CP-children.
Digo et al., 2019	Xsens Movelia	C7 T12 S1	A young healthy adult	6 mWT five times	Tilt, tilt-azimuth and twist angles. Flexion/extension, lateral bending and axial rotation	Compared both IMU sensors and optoelectronic systems, performed the tilt-twist method and obtained acceptable results (ROM differs between 0.2° and 1.1°)
Digo et al., 2020	Xsens Movelia	C7 T6 – T12 L3 S1	18 young healthy adults	three 20mWT on three conditions: 1. Normal speed 2. Slow speed 3. Self-selected comfortable speed	Tilt, twist method: Tilt, tilt-azimuth and twist angles. Flexion/extension, lateral bending and axial rotation. Relative angles from each vertebra.	Effectively identifies a pattern of trunk compensation during gait, providing a valuable indicator of efficiency in both clinical and rehabilitative settings.
Piitulainen et al., 2021	NGIMU	L3 – L5	30 CP children and adolescents (18 HP and 12 DP) 31 matching TD	10mWT on three conditions: 1. Normal gait 2. Motor dual task 3. Cognitive dual task	Spatiotemporal, gait complexity	Lower walking speed in DP in comparison to HP and TD groups. The step duration had more variance on the HP group for all the tasks involved. Gait complexity was higher in CP compared to the TD group. The step duration was greater for both dual tasks in comparison to normal gait.

4. Design of experiments

4.1. Participants

The DRK–Kinderklinik Siegen is conducting a feasibility study with children with CP who can walk independently and typically developing (TD) children. The main objective is to provide proof of concept that improvements in postural control, particularly in the trunk, following an intensive block of physiotherapy treatment (using Vojta therapy), can be quantified by analyzing IMU measurements.

Until now there are six children with CP and six in age and gender matched TD-children involved in this study. In Table 4.1, we can see the relation of the participants.

Table 4.1 Participants involved in the study

Nr.	CP – Children	Age	GMFCS Level	TD – Children
1	CP1	10	I	TD1
2	CP2	10	I	TD2
3	CP3	12	I	TD3
4	CP4	12	I	TD4
5	CP5	6	II	TD5
6	CP6	7	I	TD6

4.2. Experimental setup

4.2.1. Set-up

The test setup consists of a mat track measuring 7-meter in length and 0,7 meters in width. It incorporates two pairs of cones arranged in a configuration that facilitates the creation of a 5-meter pathway, which will serve the evaluation track for the trails. As illustrated in Figure 4.1, two video cameras are part of the setup which will record each child's performance for a posterior qualitative analysis by the physiotherapist.

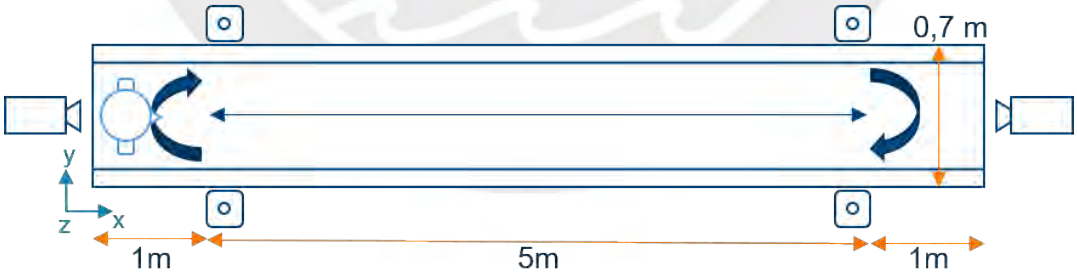


Figure 4.1 Tests set-up (designed by Bruchmüller).

For this study, four APDM Opal triaxial IMU sensors were used, which incorporated an accelerometer, gyroscope and magnetometer. The full specifications of this sensor are presented on Table 4.2.

Table 4.2 Specifications of the APDM Opal IMU (Clario, 2024).

Property	Accelerometer	Gyroscope	Magnetometer	Barometer
Axes	3	3	3	1
Range	± 200 g	± 2000 °/s	± 8 Gauss	300 – 1100 hPa
Noise Density	5 mg/√Hz for > 16 g 120 µg/√Hz for ≤ 16 g	0.025˚ /s/√Hz	2 mGauss/√Hz	1.3 Pa

Each OPAL IMU sensor was covered with a latex fingertip and attached to the child's body using an Opsite Flexifix gentle, with a silicone gel adhesive. The sensors were placed on the sternum, the lumbar region (over L4/L5 vertebrae) and on the dorsum of each foot (see Figure 4.2). The sensors sample the data at a frequency of 128 Hz.

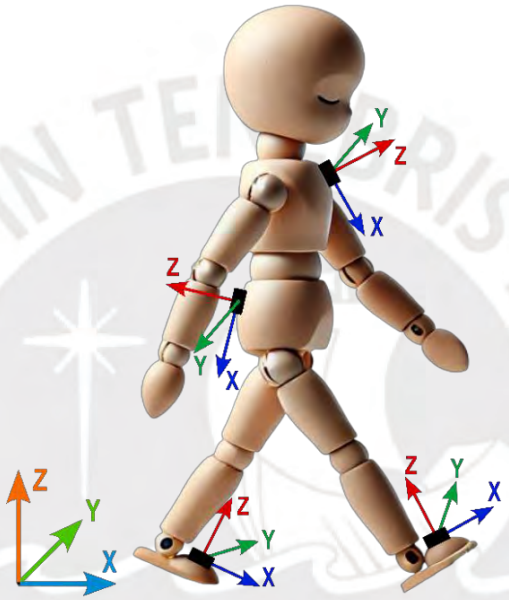


Figure 4.2 Position and coordinate system of the IMU sensors on the participants and the global coordinate system in the left corner (generated by DALL-E, adapted).

4.2.2. Tests procedure

Firstly, all four sensors were synchronized using the Mobility Lab software. Afterwards, the sensors were fixed in positions shown in Figure 4.1. The children were then instructed to remain stationary for a period of three seconds at the start of the walkway. At the end of the three seconds on a beep tone they started walking barefoot five meters along the flat walking track at a comfortable self-selected speed, back and forth three times in a straight line.

The children with CP completed one measurement, previously described, prior to the initial treatment and one after the 18th treatment. In total there were 20 treatments within 12 days. The TD children completed two measurements taken with a 12-day interval and without any therapy.

5. Data analysis

As a solution to the problem described in the first chapter, this thesis proposes the algorithm of Figure 5.1, to do the data pre-processing and processing, the gait parameters calculation and posterior comparison of datasets per patient.

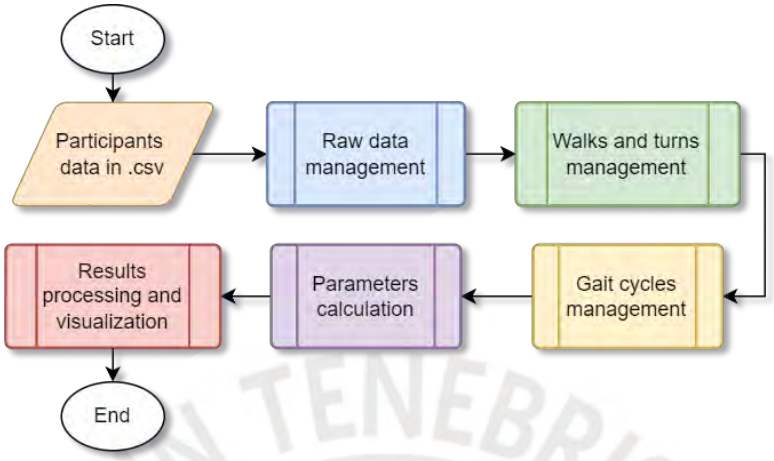


Figure 5.1 Flowchart of the general elements of the data analysis.

This chapter presents the algorithm utilized for the processing of inertial data. Accordingly, the figures presented with the methods will exclusively analyze the CP1 patient dataset, with the same procedure being carried out for each patient. Figure 5.2 provides a detailed breakdown of the data analysis in Figure 5.1, offering a more comprehensive understanding of the methods involved. The methods described in the subchapters are exemplified by the analysis of CP1, until the subchapter of parameter calculation. Nevertheless, the datasets processing and visualization of results is shown in Chapter 6.

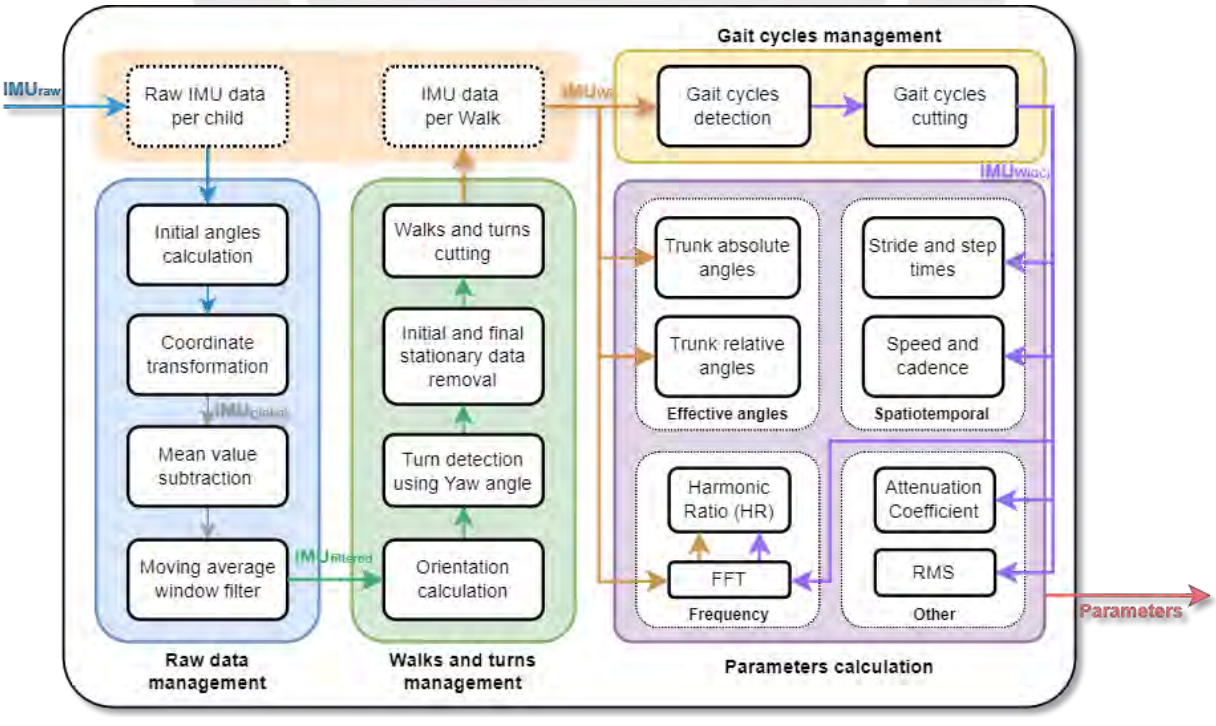


Figure 5.2 Detailed breakdown of proposed algorithm showing the handled data types: raw data IMU_{raw}, global transformed data IMU_{global}, filtered data $IMU_{filtered}$, data per-walk IMU_{W_i}, and data per gait cycle $IMU_{Wi_{GC_j}}$.

5.1. Raw data management

The data were acquired and read through CSV data files, which consisted of accelerometer, gyroscope and magnetometer raw data from the sensors, as illustrated in Figure 5.3. When analyzing one child with CP, 4 data files should be preprocessed: Pre therapy, post therapy and two measurements recorded from the matched TD-child.

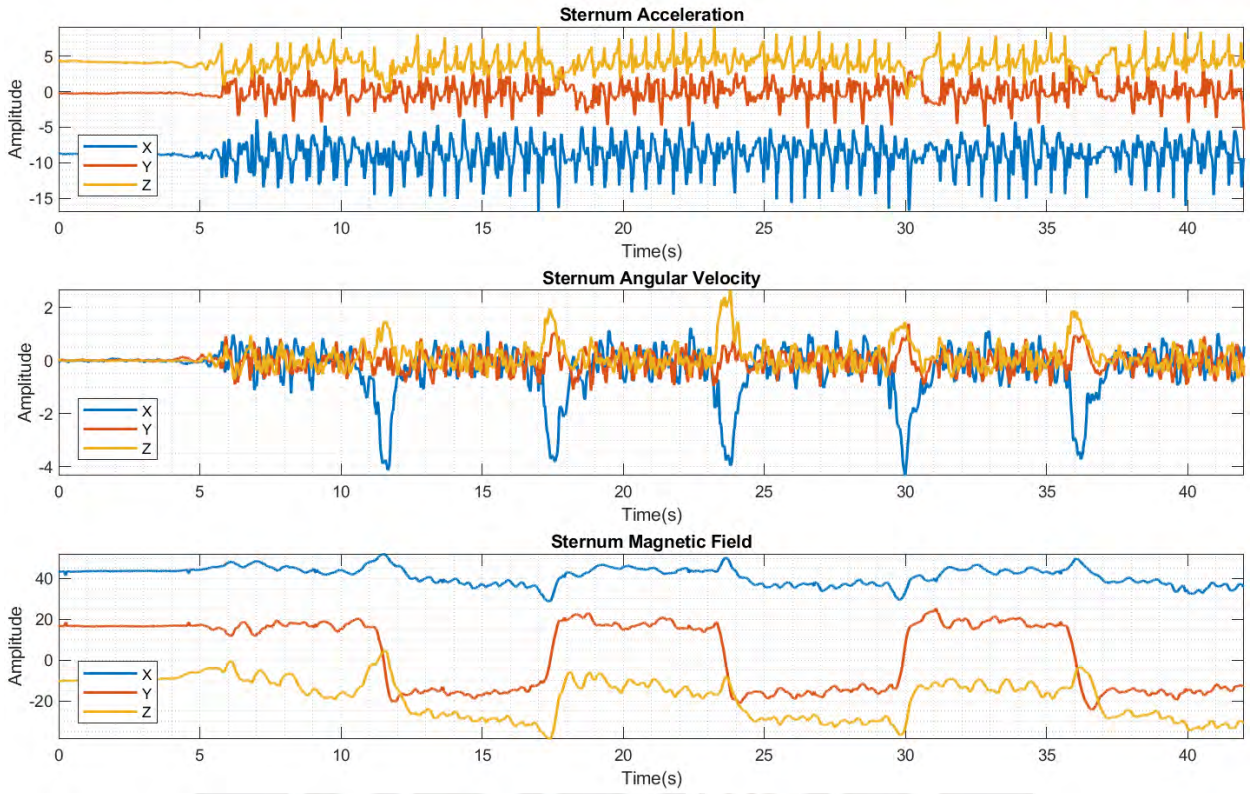


Figure 5.3 Raw data of a sternum sensor in (pre therapy measurement CP1).

5.1.1. Calculation of initial angles

A coordinate transformation was required to align all measurements into the global coordinate reference which is vital for further data processing and analysis. A crucial preliminary step is to calculate the initial angles of the rotation of the sensors due to their positioning on the participants' bodies. The calculation of these angles can be done using the initial segment of each test, when the participants remained stationary.

In this position, the sole actuating force is gravity. Therefore, it can be expected from the positioning of the sensors that the primary component of the gravity will be along the X-axis, for the sternum and lumbar sensors, and the Z-axis for the feet sensors (see Figure 4.2).

The next step is to calculate the Euler angles of each sensor: Roll (ϕ), pitch (θ) and yaw (ψ). For further reference of the representation of the Euler Angles in this thesis, see Figure 5.4.

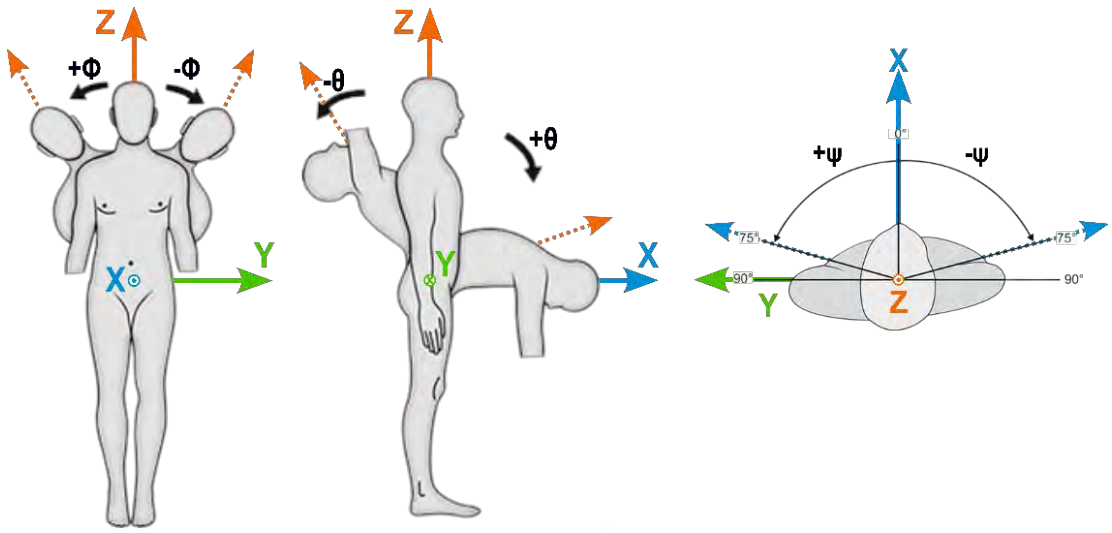


Figure 5.4 Euler Angles: Roll (ϕ), Pitch (θ) and Yaw (ψ), adapted from Jain and Jain (2016).

Roll and Pitch angle calculations

Not all Euler angles can be calculated from accelerometer data, only roll and pitch angles. We can prove this by making the R_{XYZ} rotation matrix equal to the gravity vector (Eq. 5.1 to. Eq.5.5).

$$R_{XYZ} = R_\psi R_\theta R_\phi \qquad \textit{(Eq. 5.1)}$$

$$R_{XYZ} = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} \qquad \textit{(Eq. 5.2)}$$

$$R_{XYZ} = \begin{pmatrix} \cos\psi\cos\theta & \cos\phi\sin\psi + \cos\psi\sin\phi\sin\theta & \cos\phi\cos\psi\sin\theta - \sin\phi\sin\psi \\ -\cos\theta\sin\psi & \cos\phi\cos\psi - \sin\phi\sin\psi\sin\theta & -\cos\phi\sin\psi\sin\theta - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\phi\cos\theta \end{pmatrix} \qquad \textit{(Eq. 5.3)}$$

$$\bar{g} = R_{XYZ}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\phi\cos\psi\sin\theta - \sin\phi\sin\psi \\ -\cos\phi\sin\psi\sin\theta - \cos\psi\sin\phi \\ \cos\phi\cos\theta \end{pmatrix} \qquad \textit{(Eq. 5.4)}$$

$$g = \cos\phi\cos\theta \qquad \textit{(Eq. 5.5)}$$

It can thus be demonstrated that there is no dependency on the yaw angle, since it is the angle in which the initial rotation is carried out (Pedley, 2013).

In order to obtain the roll and pitch angles, it is necessary to derive the rotation matrix that consists only of these angles (Eq. 5.7).

$$R = R_\theta R_\phi = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} \qquad \textit{(Eq. 5.6)}$$

$$R = \begin{pmatrix} \cos\theta & \sin\theta\sin\phi & \cos\phi\cos\theta \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\phi\cos\theta \end{pmatrix} \qquad \textit{(Eq. 5.7)}$$

To obtain the global coordinate system for all the sensors, in the stationary phase, the multiplication between the matrix and the acceleration vector of the IMU should be done

$$\bar{g} = R\,\bar{a}_{IMU} \quad \textit{(Eq. 5.8)}$$

$$\begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} = \begin{pmatrix} a_{x-IMU}\cos\theta + a_{y-IMU}\sin\theta\sin\phi + a_{z-IMU}\sin\theta\cos\phi \\ a_{y-IMU}\cos\phi - a_{z-IMU}\sin\phi \\ -a_{x-IMU}\sin\theta + a_{y-IMU}\cos\theta\sin\phi + a_{z-IMU}\cos\theta\cos\phi \end{pmatrix} \quad \textit{(Eq. 5.9)}$$

Therefore, the roll and pitch angles can be obtained by the Eq. 5.10 – 5.11.

$$\phi = \arctan\left(\frac{a_{y-IMU}}{a_{z-IMU}}\right) \quad \textit{(Eq. 5.10)}$$

$$\theta = \arctan\left(-\frac{a_{x-IMU}}{a_{y-IMU}\sin\phi + a_{z-IMU}\cos\phi}\right) \quad \textit{(Eq. 5.11)}$$

Yaw angle calculation

Since the use of the accelerometer data does not allow to estimate the yaw angle, its estimation can be calculated using the magnetic field data.

$$\bar{M} = R\,\bar{M}_{IMU} \quad \textit{(Eq. 5.12)}$$

$$\begin{pmatrix} M_X \\ M_Y \\ M_Z \end{pmatrix} = \begin{pmatrix} M_{x-IMU}\cos\theta + M_{y-IMU}\sin\theta\sin\phi + M_{z-IMU}\sin\theta\cos\phi \\ M_{y-IMU}\cos\phi - M_{z-IMU}\sin\phi \\ -M_{x-IMU}\sin\theta + M_{y-IMU}\cos\theta\sin\phi + M_{z-IMU}\cos\theta\cos\phi \end{pmatrix} \quad \textit{(Eq. 5.13)}$$

The yaw angle can be estimated as the angle between the magnetic field of the Y-axis and the X-axis, therefore:

$$\psi = \arctan\left(\frac{M_Y}{M_X}\right) \quad \textit{(Eq. 5.14)}$$

$$\psi = \arctan\left(\frac{M_{y-IMU}\cos\phi - M_{z-IMU}\sin\phi}{M_{x-IMU}\cos\theta + M_{y-IMU}\sin\theta\sin\phi + M_{z-IMU}\sin\theta\cos\phi}\right) \quad \textit{(Eq. 5.15)}$$

As can be observed, all the Euler angles are dependent only on the accelerometer or the magnetometer data. Thus, they could be used in situations that involve not only stationary but also dynamic data.

Proper estimation of the Euler angles

The estimation of the Euler angles throughout the stationary phase could be estimated only with the equations shown above. However, the orientation estimation will have measurement errors given by the accelerometer and magnetometer. Therefore, it would have a considerable impact on the quality of the angle's calculation (Kok et al., 2017).

To mitigate the likelihood of measurement errors, it is recommended the use of sensor fusion algorithms, such as Complementary Filtering, the Extended Kalman Filter or the Madgwick Filter. For the sake of simplicity, in this work the Complementary Filtering is used (see Section 2.4.1) with a tuning parameter of $\gamma = 0.95$ to fuse the information of the accelerometer, gyroscope and the magnetometer.

5.1.2. Rotation to global coordinate system by quaternions

On the other hand, to avoid drift problems, the rotation of the initial data (sensor frame) to the global coordinate system (global frame) is carried out using quaternions. The procedure for doing this involves, first, the creation of the quaternions for the initial estimated angles ($q_{roll}, q_{pitch}, q_{yaw}$) and then, the multiplication of these quaternions to get the final rotation quaternion (q_{RPY}).

$$q_{RPY} = q_{yaw} \star q_{pitch} \star q_{roll} \qquad \textit{(Eq. 5.16)}$$

By having this rotation quaternion, the data rotation of the sensors can be done with the following Eq. 5.17 (Shoemake).

$$v_{global} = q_{RPY} v_{IMU} \overline{q}_{RPY} \qquad \textit{(Eq. 5.17)}$$

where $v_{IMU} = (\boldsymbol{v}_{IMU}, 0)$ is the quaternion of the IMU data and $\overline{q}_{RPY}$ is the conjugate quaternion of q_{RPY}.

Figure 5.5 describes the explained coordinate transformation method applied for each IMU in the pre-therapy dataset, focusing only on the initial five seconds of the test, which are designated as the static part of the test for better comprehension.

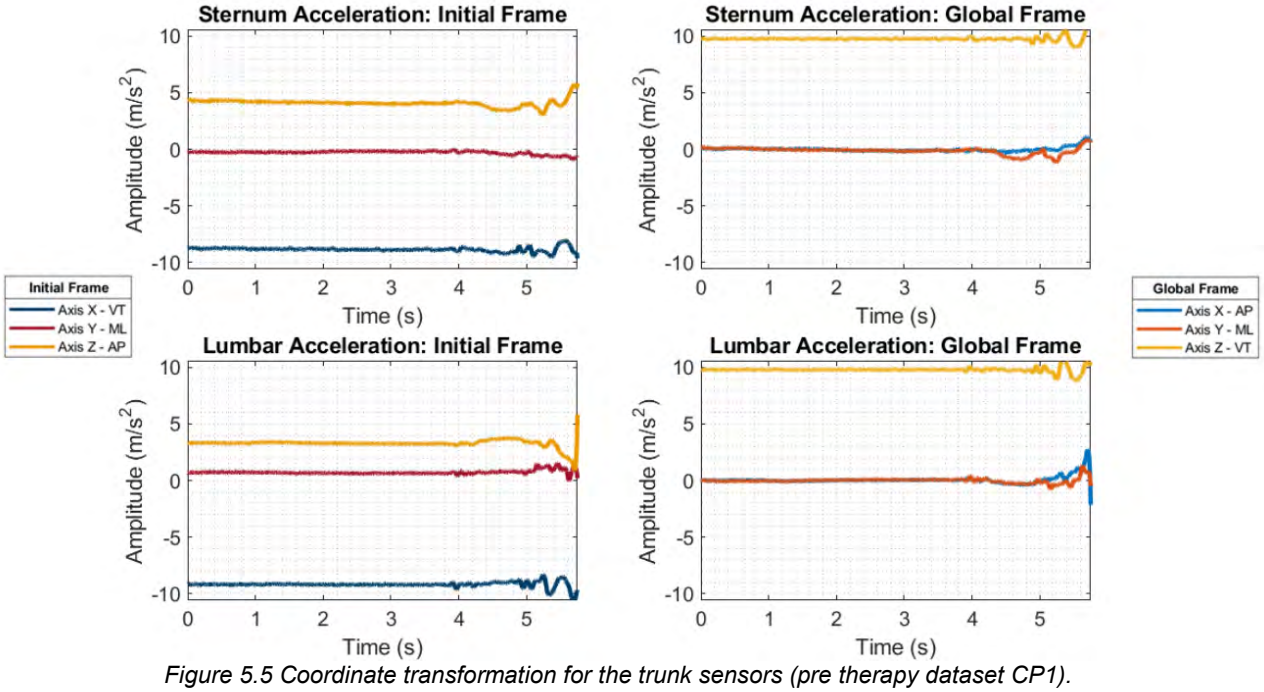


Figure 5.5 Coordinate transformation for the trunk sensors (pre therapy dataset CP1).

In conclusion, the transformation of the sensor data to the global coordinate system was successfully completed. This was achieved by confirming that the X- and Y-axis data were located in or near to zero, and that the Z-axis data reflected the gravitational acceleration component, which was distributed differently across the sensors' coordinate systems.

5.1.3. Data preprocessing

Mean subtraction

Firstly, the inertial data was subtracted by the mean value of the static part, the initial five seconds in which the patient stood still. In other cases, only a gravity subtraction is performed on the Z-axis for accelerometer data. Nevertheless, even when the remaining data is situated near to zero, it is preferable to maintain it in zero value.

Filtering

Subsequently, the data must be filtered to facilitate the succeeding analysis. As commented in Section 2.5, the most used filters in this approach are lowpass Butterworth and Moving average filter. Therefore, in this section further investigation regarding the effect of both filters on the current dataset is provided.

Firstly, the effect of both filters was tested for different parameters, which is illustrated on Figure 5.7. This evaluation was accomplished using the lumbar AP acceleration signal, which is more susceptible to noise than signals from other axes.

As illustrated in Figure 5.7, the lowpass Butterworth filter provides minimal attenuation of peaks and offset of the data by lowering the cutoff frequency (f_c). This response is logical since the attenuation of high frequency components indicated by the f_c, generates the smoothing of the peaks. The magnitude of the shift in the data is directly proportional to the order of the filter selected resulting in the largest shift. In this case, the used order is a fourth order, thus, the shift is more perceived in the smaller f_c.

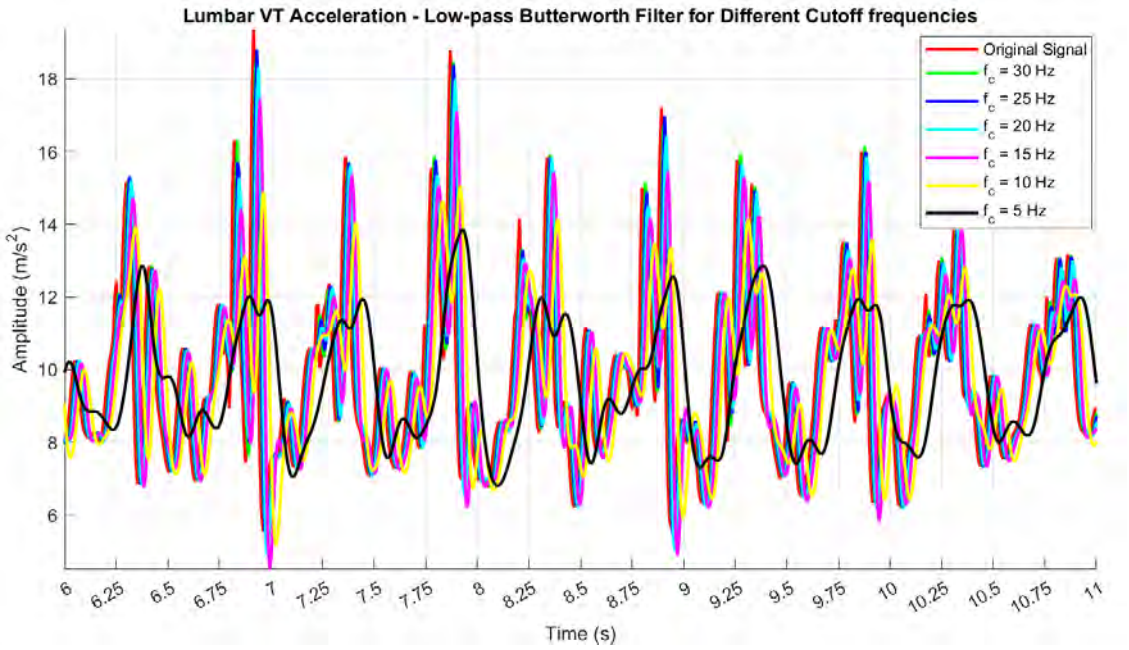


Figure 5.6 Filters effect on many cutoff frequencies for the lowpass Butterworth filter.

In most of the CP related gait analysis, the cutoff frequency ranged from 15 to 20 Hz. In the graph, these mentioned frequencies effectively preserve the peaks, although they exhibit a slight shift in comparison to the original signal.

The effect of the moving average filter in Figure 5.7, in contrast, shows a discernible reduction in the peak's amplitude and the absence of displacement of the data. This would result in a more effective preservation of the signal patterns and the elimination of noise introduced by extraneous peaks.

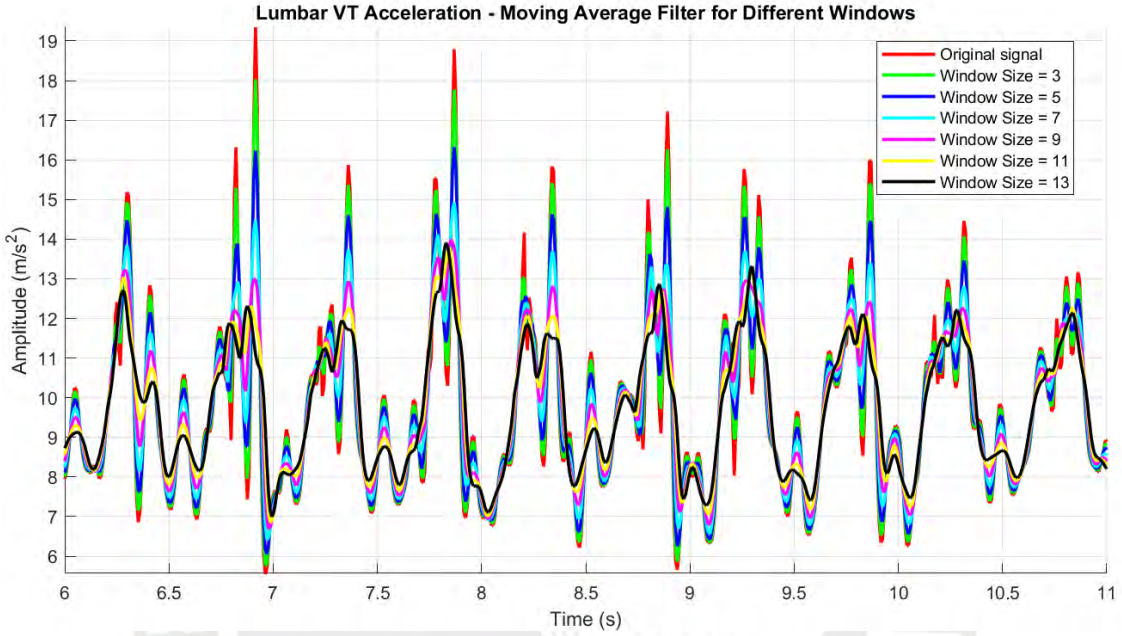


Figure 5.7 Filters effect on many window sizes for the moving average filter.

Nevertheless, as this filter does not operate by means of frequency attenuation, Figure 5.8 and Figure 5.9 illustrate the FFT magnitudes of both mentioned filters, in order to facilitate a comparison of their efficiency in the frequency domain.

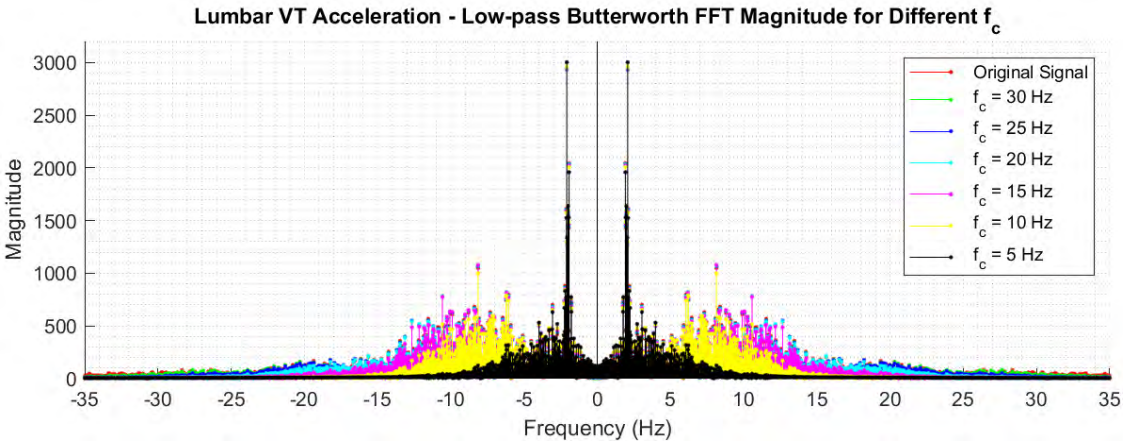


Figure 5.8 FFT magnitude for different cutoff frequencies for the lowpass Butterworth filter.

The lowpass filter from Figure 5.8 shows the attenuation of frequencies given by the f_c, meanwhile, the moving average window filter from Figure 5.9 attenuates the signal frequencies proportionally in both magnitude and range of frequencies. To take one

example, in the case of a window size of 13, a certain degree of attenuation can be observed up to a frequency of 10 Hz and an approximate magnitude of 400 m/s^2. Conversely, in Figure 5.8, a f_c of 10 Hz shows an approximate magnitude of 1000 m/s^2.

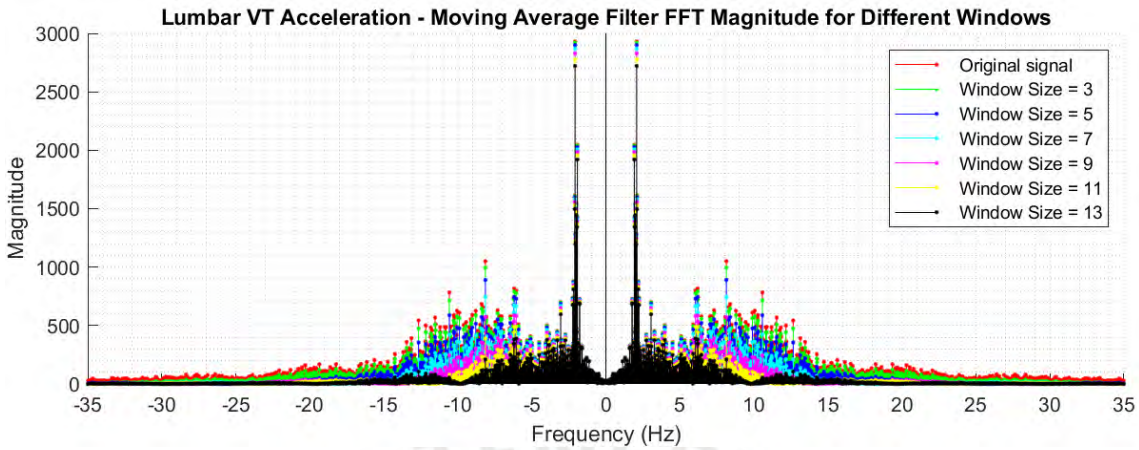


Figure 5.9 FFT magnitude for different window sizes for the moving average filter.

Filter selection

In the case of this investigation, the primary focus is to preserve the pattern of the inertial signals, and another objective is to conduct partial automatic gait cycle detection. Then, a reduction or smoothing of the peak's amplitude is necessary to obtain a more accurate gait event detection, which are based mostly on the acceleration peaks detection.

As conclusion the selected filter is the moving average filter. However, excessive peak smoothing should be avoided. Therefore, a filter with a window of 6 is the choice for the following data analysis, because it permits the reduction of the peak amplitude while maintaining the pattern of the signal and preventing data shifting.

5.2. Walks and turns management

Firstly, to identify the 180° turns made by the participants on each test, a thorough analysis of all raw data was executed, which allowed the collection of angular velocity and magnetic field data that seemed to trigger a response to the realization of turns. Then, two approaches to detect turns were developed.

5.2.1. First approach

This approach was developed using only raw magnetic field data. Figure 5.10 illustrates the analyzed signals, where the turns are exhibited as positive or negative slopes, indicative of significant transitions from the initial to the final turn direction.

Therefore, the first approach consists of detecting the slopes and separating them from the plateaus using the sliding window method. However, this method detected better

when the slopes were larger, so in this case, the recommended signals to use could be raw sternum Y-axis, lumbar Y-axis or Z-axis.

Nevertheless, the sliding window did not function optimally when utilizing a single magnetometer signal, since one threshold value did not detect every expected turn. Furthermore, the combination of two signals did not yield optimal results as well. Consequently, this approach finally employs the combination of the three previously mentioned magnetometer data each with a different threshold.

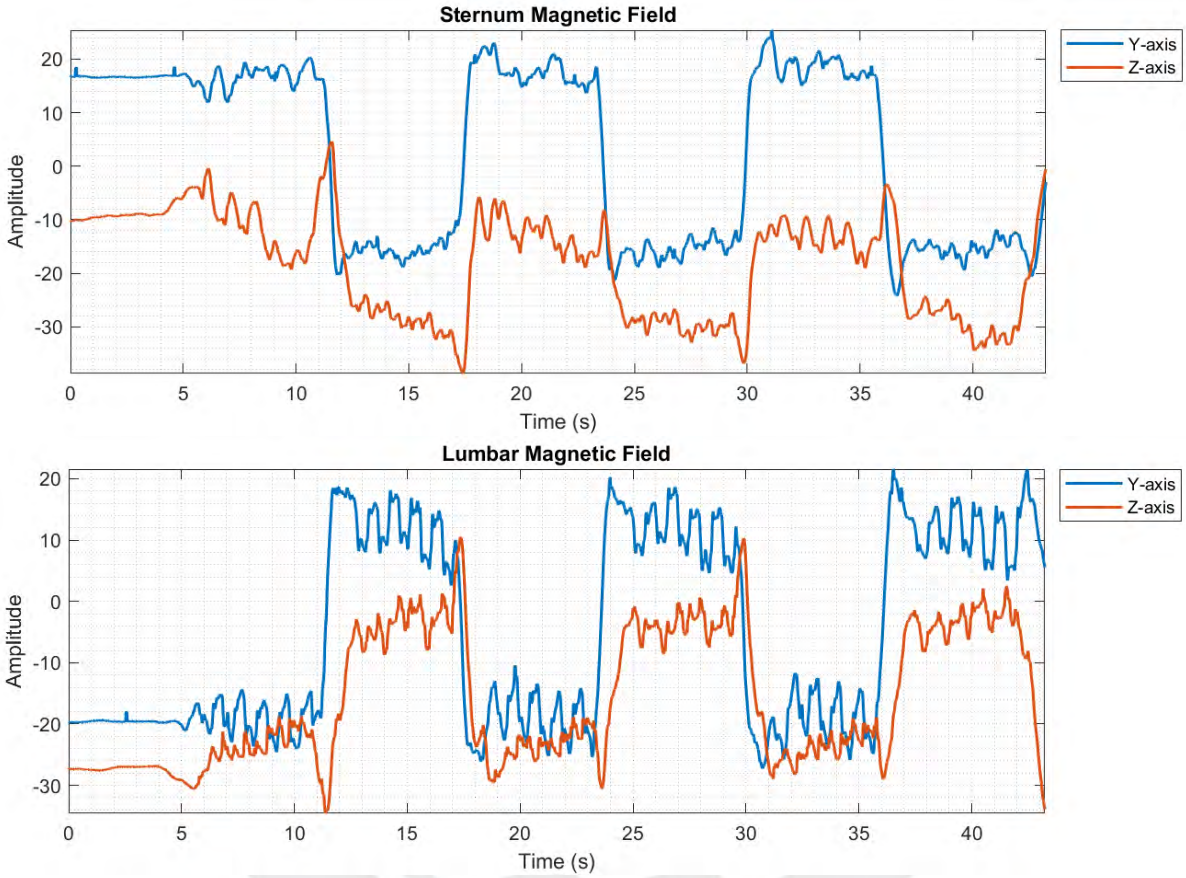


Figure 5.10. Magnetometer signals from sternum (top) and lumbar (bottom).

While this approach proved effective for the majority of children, it was not the final turn detection algorithm used. That was due to the variability of the magnetic field on each test day.

5.2.2. Second approach

The alternative approach does not utilize the raw data correctly; instead, it employs the orientation data provided by the complementary filter (see 5.1.2). In Figure 5.11, the Euler angles for an entire test of the sternum sensor are presented.

From this figure, it is evident that the only pattern that repeats five times and, therefore, represents the turns is clearly discernible in the plot of the yaw angle versus time. Additionally, compared to the sternum gyroscope X-axis data from Figure 5.3, which also gave similar patterns, the yaw angle provides a cleaner and higher peak.

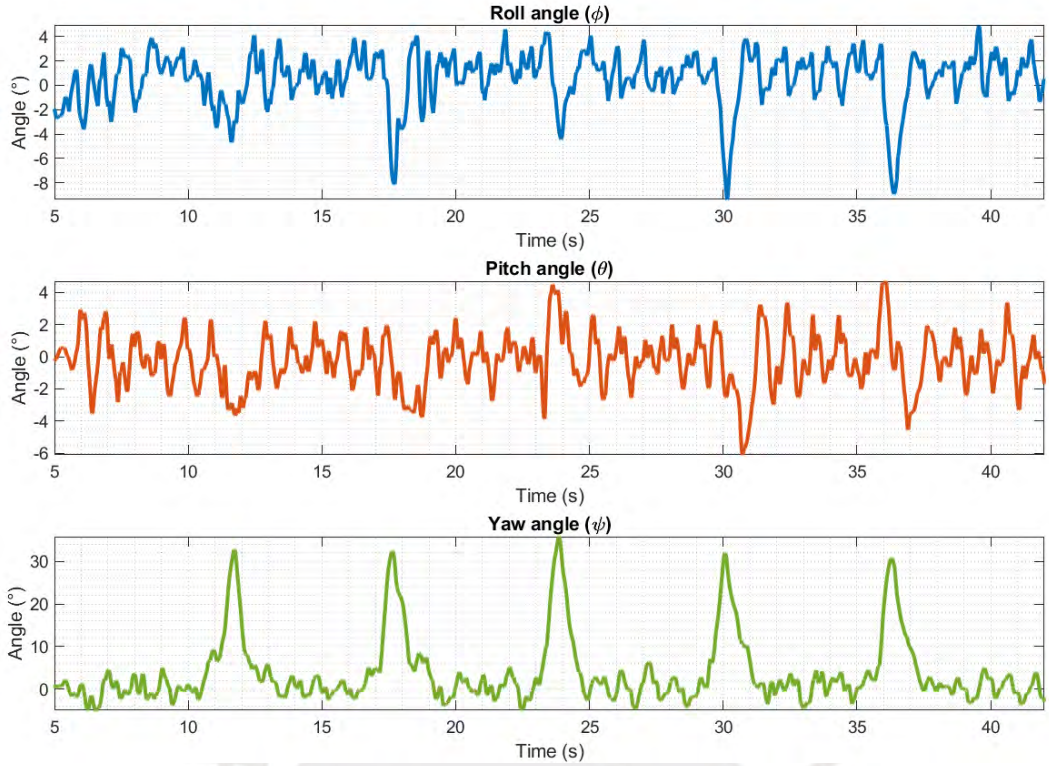


Figure 5.11 Euler angles for sternum IMU Pre therapy test.

Additional to the sternum orientation data, to be certain of the peak detection, the lumbar spine orientation data is also included. Both signals have a similar peak pattern, therefore, this makes the peaks easier to detect using the *findpeaks* function (see Appendix G.3) and only using one threshold, in comparison to the three thresholds used on the first method.

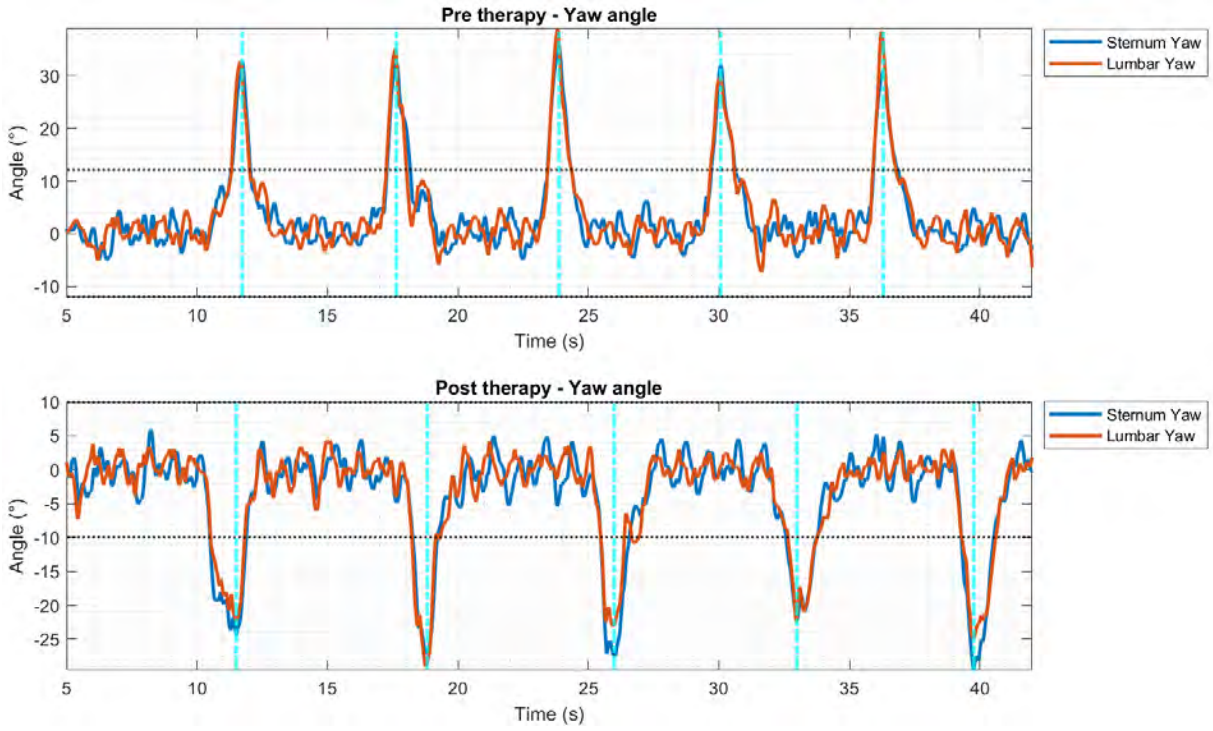


Figure 5.12 Peak detection and thresholds for CP1 pre and post therapy data.

However, this pattern is variable depending on each test, as the yaw angle pattern provides only turns to one side, which in this case would be to the left side. Subsequently, in the case of other tests, turns to the left and the right side are possible, which would result in the occurrence of positive and negative peaks. Because of the fact that these peaks are only mirrored with respect to the time axis, the previously established threshold is also mirrored. The double thresholding process is illustrated in both the pre- and post-therapy datasets in Figure 5.12.

Subsequently, in order to identify the initial and concluding points of each turn, it is necessary to locate the two nearest zero-crossing points in the interval preceding and succeeding the identified peak. With that final measure to take, each one of the five turns is then detected (see Figure 5.13).

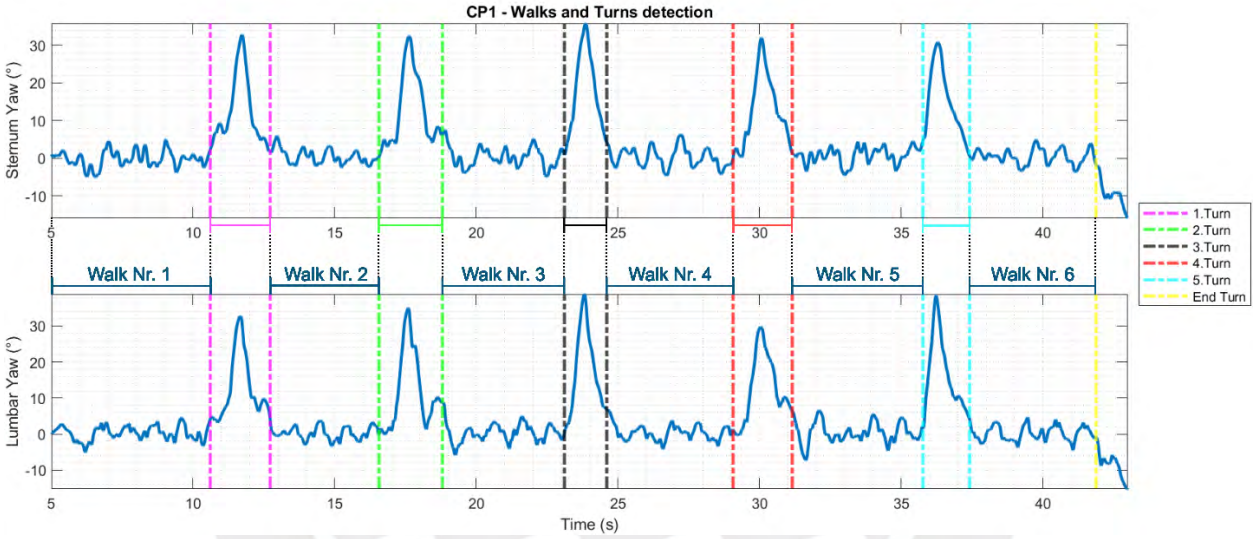


Figure 5.13 Turn segments detection for CP1 pre therapy data

Due to the effectiveness of the second approach in comparison to the first one, it was used to determine the turn segments for all CP- and TD-children. Refer to Appendix F.1 to see the walks and turns detection algorithm in detail.

5.2.3. Management of errors

Both methods present the inclusion of magnetometer data. The first approach uses the raw magnetic field data which, as presented in Section 2.3.3, is susceptible to hard- and soft-iron errors. Those errors deviates or distorts the magnetometer data, respectively, depending on the surrendering environment.

Meanwhile, the second method includes the orientation calculation, the yaw angle specifically. This angle has been estimated with both magnetometer and gyroscope data through the complementary filter. Consequently, it is possible that the approach may

exhibit some of the same errors that were previously discussed in relation to the first method.

5.3. Gait cycles management

In this thesis, two methods of gait event detection are presented using trunk information but different sensor data and axes. Then, a similar procedure of gait cycle definition is presented.

5.3.1. First approach

The initial approach entails the estimation of a cyclic event that is independent from gait events, such as heel-strike, toe-strike, and so forth. For this approach, the cyclic event was defined based on an approximation of the center of mass (COM) displacement throughout the gait. According to Kuo and Donelan (2010), the COM displacement behaves in a manner similar to that illustrated in Figure 5.14 for both normal, longer and faster steps, with initial contact or HS marking the onset of the gait.

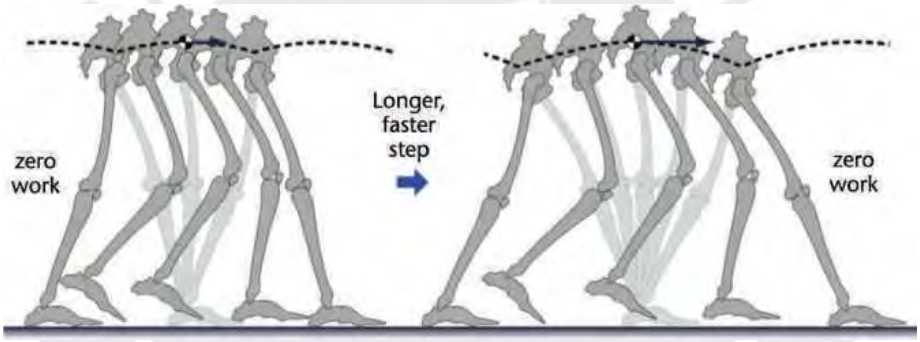


Figure 5.14 COM displacement according to the Pendulum Swing Leg analogy (Kuo and Donelan, 2010).

In both instances, the initial contact or HS represents the local minimum of the pattern. Conversely, the local maximum is indicative of the midswing phase.

In the context of this investigation, the COM can be approximated to the sensor located on the lumbar spine. Consequently, the lumbar IMU displacement on the VT-axis should exhibit a similar pattern. It is evident that the resulting waveform should not display a sinusoidal shape, as observed previously. However, the local maxima and minima behavior should be analogous.

Definition of cyclic events

As previously explained, the maxima would represent a certain and cyclic moment on the midswing (midSw) and midstance (midSt) phases, when both feet are considered. The minima would represent a moment in the double support (DS) phase, which is not directly associated with the foot initial contact. Subsequently, the double differentiation of the signal with sine or cosine shape would give an inverted signal.

Therefore, the minimum of the sternum acceleration VT-axis would denote an event on the double support phase, meanwhile, the maximum would stand for an event in the midSt or midSw phases. Figure 5.15 illustrates the original and filtered linear VT acceleration of the lumbar with both maxima and minima as double support and midSt/midSw events.

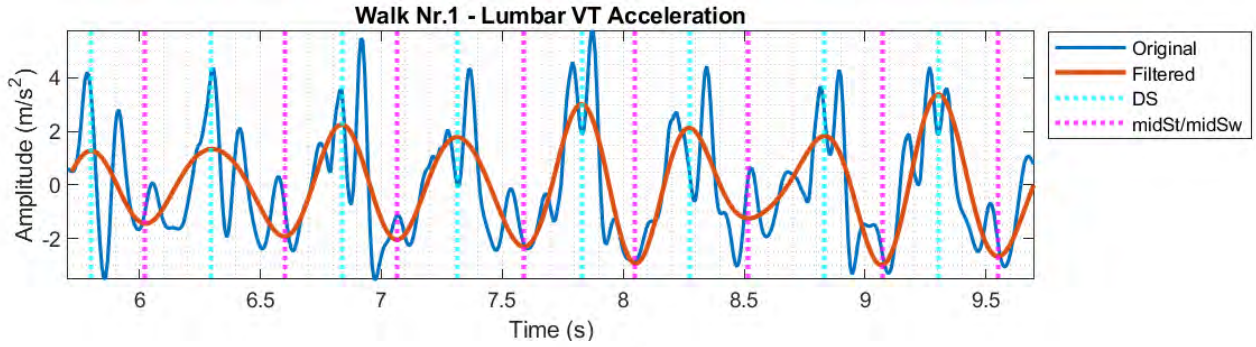


Figure 5.15 Gait events detection using lumbar VT acceleration (Pre therapy CP1).

Displacement calculation

The original acceleration signal was initially integrated twice to find the linear displacement of the sternum along the VT-axis. Subsequently, the signal was filtered using a bandpass filter (function *bandpass*) with cutoff frequencies ranging from 1 to 3 Hz (see Figure 5.16). Then, this filtered displacement was derived twice to get the corresponding sinusoidal wave from which locate the previously defined phases using the function *findpeaks*.

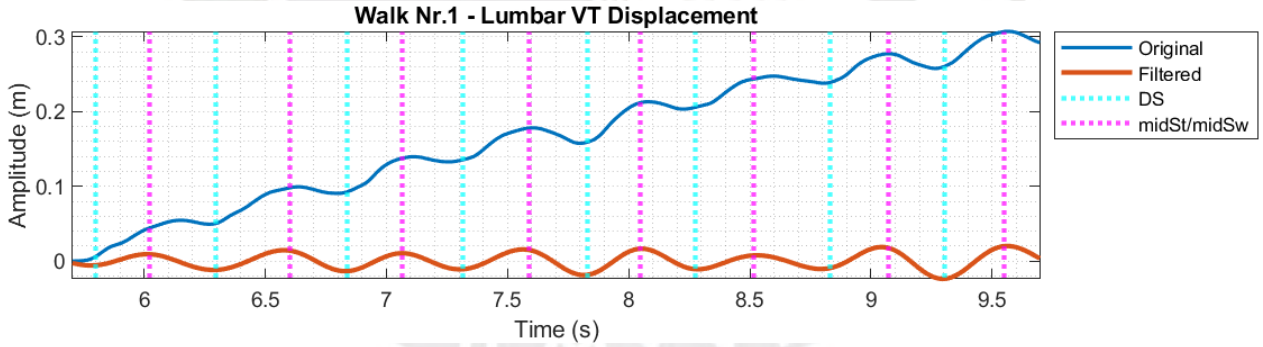


Figure 5.16 Drift correction of lumbar VT displacement (Pre therapy CP1).

Gait cycle definition

Once the gait events have been obtained from both methods, the gait cycles must be defined. From the sensors placed on the sternum spine, the DS phase of both feet is identified. Therefore, the distance between two consecutive DS is defined as a step, meanwhile, the distance between a DS and the subsequent DS is distinct as a stride and represent a complete gait cycle.

5.3.2. Second approach

Another approach is based on the algorithm proposed by Digo et al. (2020), which consisted of evaluating the lumbar spine acceleration of the AP axis. The maximum values were considered as a heel-strike (HS) and the minimum values as toe-strike (TS). As described in Section 3.2.3, this method was validated in that investigation with 18 young healthy adults; therefore, it would be more difficult to prove its effectiveness on the acceleration of children with CP. Thus, a simple peak detection algorithm wouldn't provide these two events because of the noisier acceleration behavior of CP signals.

On the other hand, some investigations that also focused only on gait event detection using trunk sensors placed in the lumbar spine, have used a Continuous Wavelet Transform (CWT) to perform this detection of events on healthy adults (McCamley et al., 2012; O'Brien et al., 2019). This method allows for the estimation of peaks by decomposing signal frequencies, thereby preserving the most important frequency information, such as sloped minima or maxima, while removing most superfluous data that could potentially introduce noise. Then, this information is displayed in the signal's time domain.

As a solution to the mentioned issue, an efficient estimation of the HS and TS was conducted using the lumbar AP-axis acceleration. This was achieved by employing the *cwt* function and then, the *findpeaks* function, to identify both gait events. The wavelet used for both events is the Morlet wavelet instead of the Gaussian. In the current version of MATLAB, there is no Gaussian wavelet, thus, the most similar wavelet is Morlet wavelet.

Event detection

Firstly, the detection of the HS event is accomplished through the estimation of the maxima and the estimation of the minima of the integral of the coefficients obtained from the signal's CWT. Each peak detection was obtained with the *findpeaks* function. Subsequently, a subtraction of the two-preceding time-series estimates was used to identify the peaks exhibiting the greatest frequency fluctuations.

In second instance, the TS event detection was achieved with the use of the same algorithm employed to detect the HS event. However, the time-series estimates were computed using the minimum values of the lumbar AP acceleration and the maximum values of the integral of the coefficients of the CWT from the signal.

Subsequently, the time-series estimation of the gait events was employed to facilitate the selection of the events, with the consideration that a HS, or maximum peak, must be

accompanied by a following TS, or minimum peak. Then, the subtraction of the timestamps of each HS and TS was computed. Thus, only the events that occurred within a time range were saved and considered as the appropriate heel- or toe-strike events.

The duration of the interval was approximately from 20 to 250 milliseconds and was determined through experimental analysis of the lumbar AP acceleration of each child involved. For instance, the CP1 range is 20 to 130 ms, while the TD1 range is 20 to 170 ms. Figure 5.17 illustrates the results of this procedure. Refer to Appendix F.2 for a detailed explanation of the gait events detection, and to Appendix F.3 for more insight into the estimation of the HS and TS events.

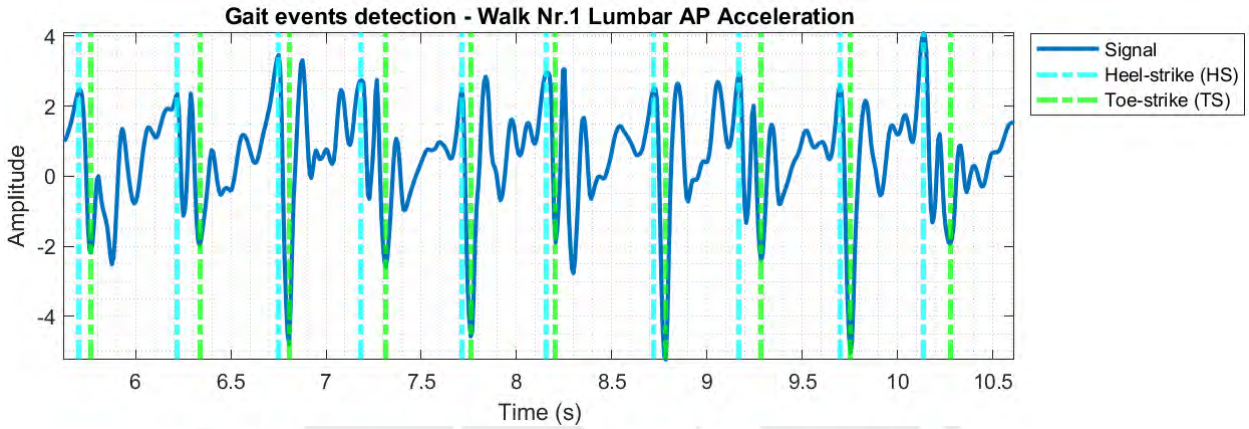


Figure 5.17 Gait event detection using lumbar spine acceleration along AP-axis from Pre therapy for CP1.

Gait cycle definition

The gait cycle definition is similar as the previous approach, however, in this case the identification of events is obtained instead of a phase event; therefore, this approach would permit a more direct estimation of the foot's initial contact. From the sensors placed on the lumbar spine, the HS phase of both feet is identified. Subsequently, two consecutive HS are defined as a step, meanwhile, and an HS and its subsequent HS are defined as a stride.

5.4. Parameters calculation

5.4.1. Spatiotemporal

The spatiotemporal parameters considered were the speed, cadence per step and stride and average step and stride length. Some of these calculations (Eq. 5.18 to Eq. 5.23) were adapted from Contini et al. (2019) and others were estimated from the gait cycles information.

$$speed(m/s) = \frac{d}{\Delta W_{i_{time}}} \qquad \textit{(Eq. 5.18)}$$

$$cadence\ (step/s) = \frac{Nr.steps}{\Delta W_{i_{time}}} \qquad \textit{(Eq. 5.19)}$$

$$cadence\ (stride/s) = \frac{Nr.stride}{\Delta W_{i_{time}}} \quad \textit{(Eq. 5.20)}$$

$$step\ length\ (m) = \frac{d}{Nr.steps} \quad \textit{(Eq. 5.21)}$$

$$stride\ length\ (m) = \frac{d}{Nr.stride} \quad \textit{(Eq. 5.22)}$$

$$stride\ time\ (m) = \Delta t_{GC_i}, step\ time\ (m) = \frac{stride\ time}{2} \quad \textit{(Eq. 5.23)}$$

The distance would be the pathway distance $d = 5$ m. The calculations were initially estimated per walk. Subsequently, the mean speed value for the dataset was calculated as the average of all the walks.

5.4.2. RMS acceleration

The RMS of the trunk sensors acceleration was calculated according to Eq. 2.1 explained in Section 2.1.3. This parameter is calculated from the acceleration obtained from IMU sensors to calculate the magnitude of the evaluated signal in a certain direction.

Sekine et al. (2013) calculated this parameter with consideration of the three axes (AP, ML and VT) to, additionally, estimate a ratio of the RMS ($RMSR$) per direction with regard to the total RMS of the three-dimensional signal (RMS_{total}).

$$RMSR_{AP} = \frac{RMS_{AP}}{RMS_{total}} \quad \textit{(Eq. 5.24)}$$

where:

$$RMS_{total} = \sqrt{RMS_{AP}^2 + RMS_{ML}^2 + RMS_{VT}^2} \quad \textit{(Eq. 5.25)}$$

The RMS and RMSR values for every axis were calculated per gait cycle and per walk. Then, the average RMS and RMSR values were estimated per dataset by averaging the RMS values of all the six walks.

5.4.3. Attenuation coefficient

The formula used to estimate the attenuation coefficient of the trunk sensors acceleration was adapted from Eq. 2.2, as detailed in Section 2.1.3. In this case, the equation to calculate the attenuation coefficient or percentage from the sternum spine sensor with respect to the lumbar spine sensor is presented in Eq. 5.25.

$$AC = \left(1 - \frac{RMS_{sternum-acc}}{RMS_{lumbar-acc}}\right) \times 100\% \quad \textit{(Eq. 5.26)}$$

The AC was calculated per gait cycle and per walk, due to the continuity of the signal. Therefore, the dataset's attenuation coefficient was averaged and saved.

5.4.4. Harmonic ratio

The HR of the trunk sensors is calculated with Eq. 2.3 per every direction axis. Subsequently, this parameter is calculated first per gait cycle, then per walk. Finally, the mean HR of the dataset is calculated by taking the mean of each walk's HR.

5.4.5. Absolute angles

The absolute angles of the trunk correspond to the roll, pitch and yaw angles calculation made with the complementary filter. According to the coordinate system, Table 5.1 presents the convention of the Euler angles to a clearer analysis of the angle values in the following chapters.

Table 5.1 Convention of signs in Euler angles.

Angle	Axis	Sign	Anatomical movement
Roll (ϕ)	AP	+	Lateral bending to the right
		-	Lateral bending to the left
Pitch (θ)	ML	+	Sagittal bending forward (Flexion)
		-	Sagittal bending backward (Extension)
Yaw (ψ)	VT	+	Longitudinal torsion to the left
		-	Longitudinal torsion to the right

5.4.6. Relative angles

The relative angles of the trunk were calculated by using the quaternions of the absolute angles of the sternum and lumbar spine, in addition to the application of Eq. 5.26.

$$q_{rel} = q^{*}_{lumbar}\, q_{sternum} \qquad \textit{(Eq. 5.27)}$$

The formula provides the trunk relative angle with the lumbar spine as a fixed point of reference. Accordingly, the convention of signs is consistent with the presented in Table 5.1.

6. Results

This chapter presents the results for each child with CP included in the study. Initially, the following radar plots illustrate a summary and description of the most effective parameters for each child.

These figures were generated using the *spider_plot* toolbox in MATLAB (Moses, 2024). This chapter offers a brief overview, while more detailed information is provided in each subsection corresponding to the parameters.

As illustrated in Figure 6.1, CP1 exhibits a decrease in sternum ML RMS, the sternum roll and yaw angle, the sternum and lumbar AP HR, ML and VT AC, and the trunk relative yaw angle. Furthermore, there is an increase in AP attenuation coefficient.

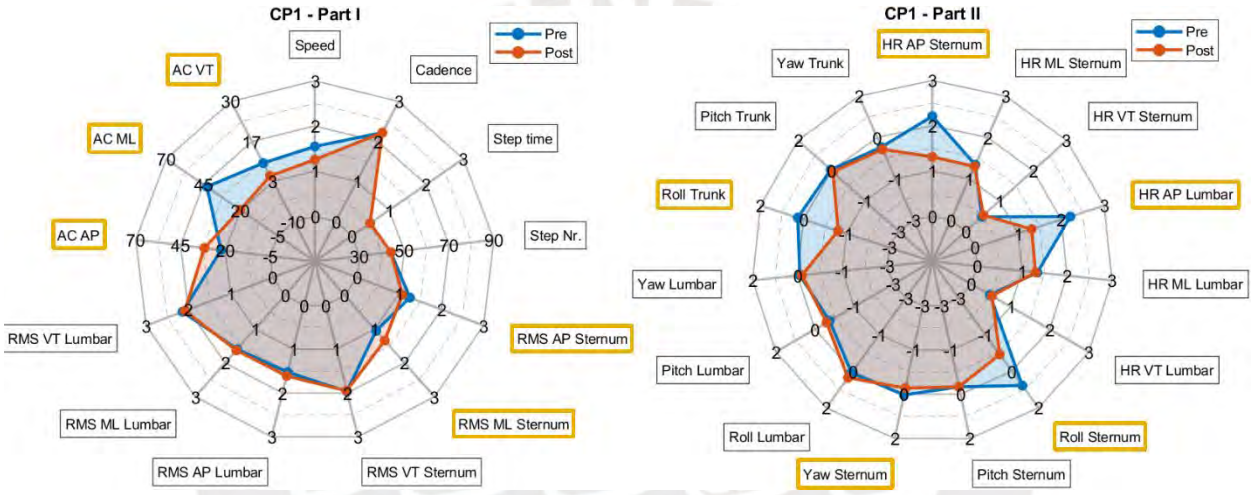


Figure 6.1 Radar plots for multiple parameters for CP1.

Figure 6.2 illustrates a slight variation of the sternum roll, all the lumbar absolute angles and each trunk relative angle of CP2. Additionally, a minor decrease in AC in the ML and VT axes, the sternum VT RMS and lumbar AP RMS is appreciated.

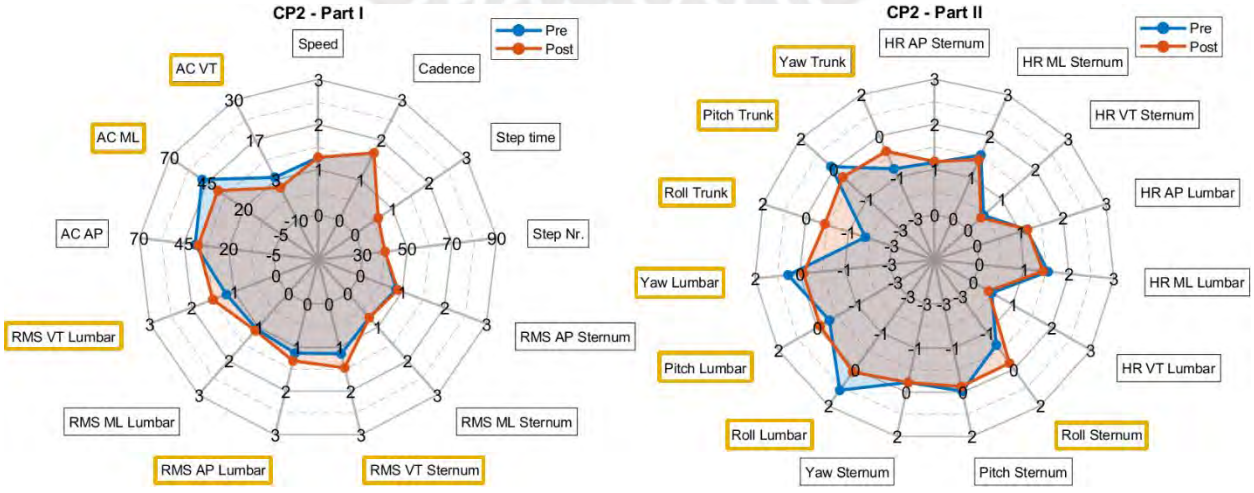


Figure 6.2 Radar plots for multiple parameters for CP2.

Regarding child CP3, Figure 6.3 illustrates a slight variation of sternum pitch and yaw, as well as lumbar roll angles and trunk relative angles. A slight reduction was also observed on parameters like sternum roll and lumbar AP RMS. However, there are no great noticeable changes after treatment.

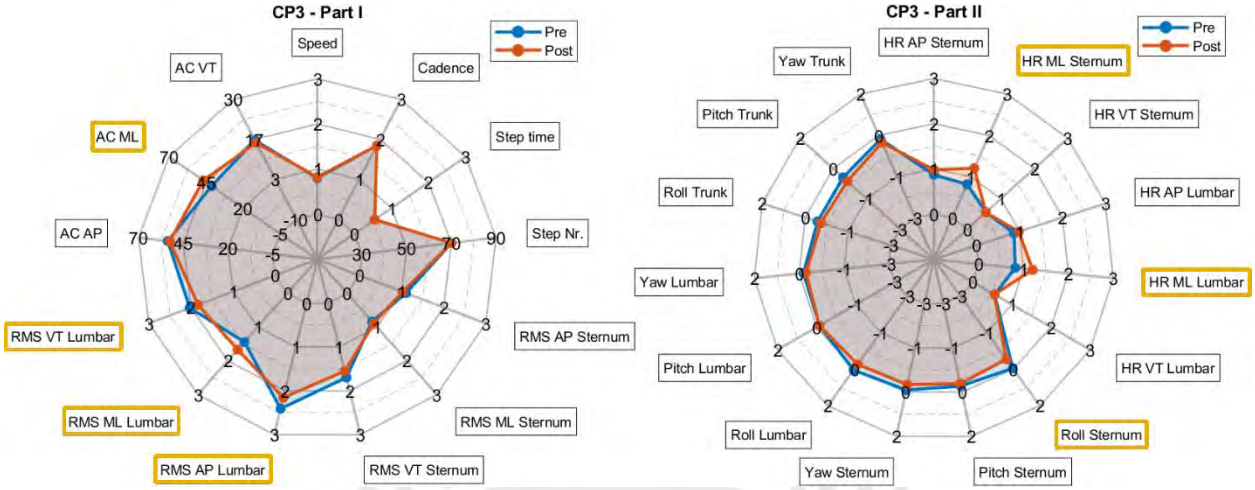


Figure 6.3 Radar plots for multiple parameters for CP3.

Figure 6.4 details a minor increase on sternum ML and VT RMS, sternum yaw, lumbar pitch and yaw. Furthermore, there is a considerable reduction in ML AC and a slight decrease of lumbar VT RMS.

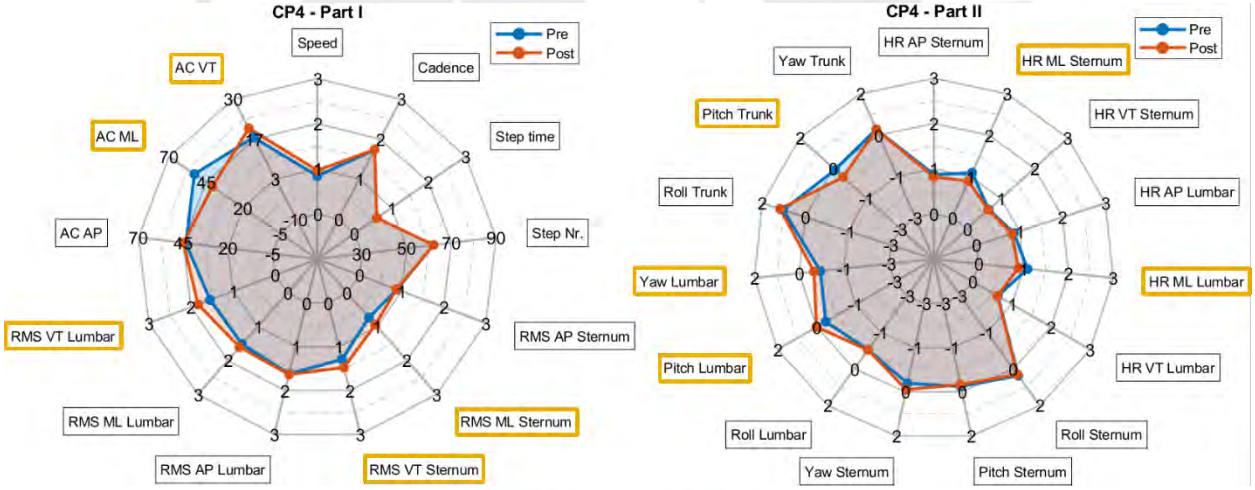


Figure 6.4 Radar plots for multiple parameters for CP4.

Figure 6.5 shows slight variations in almost every sternum and lumbar RMS, AC, lumbar HR parameters and trunk relative angles. The most remarkable variations are the decrease in sternum ML RMS and pitch angle, as well as the minor increase of sternum yaw and significant increase of ML AC.

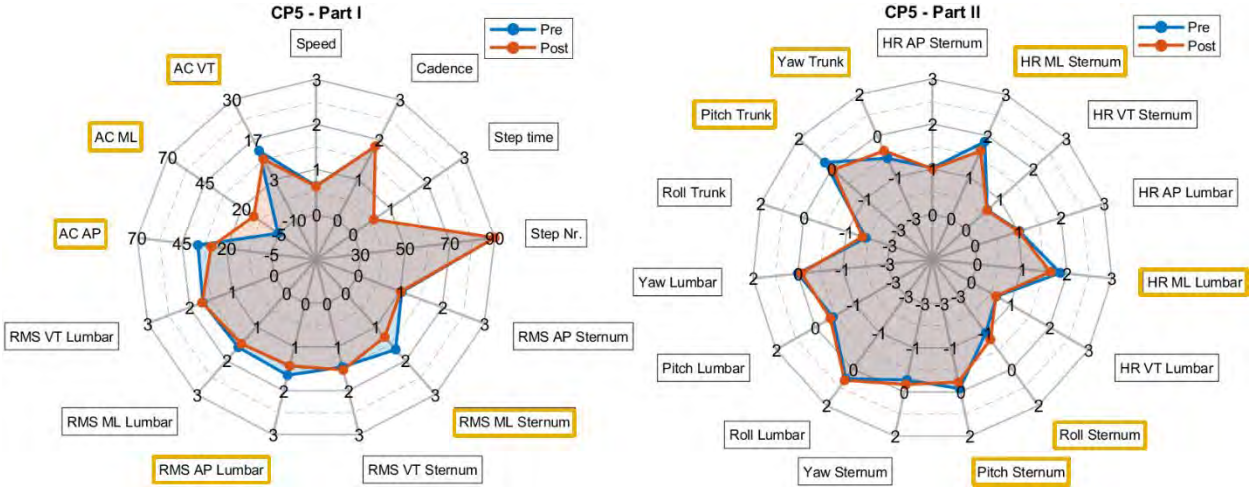


Figure 6.5 Radar plots for multiple parameters for CP5.

Figure 6.6 provides a detailed variation in parameters in both CP6 datasets, pre- and post-therapy. It is observed an increase of sternum roll, ML AC and all trunk relative angles, accompanied by a minor decrease on sternum AP and ML RMS, as well as lumbar AP RMS and absolute angles.

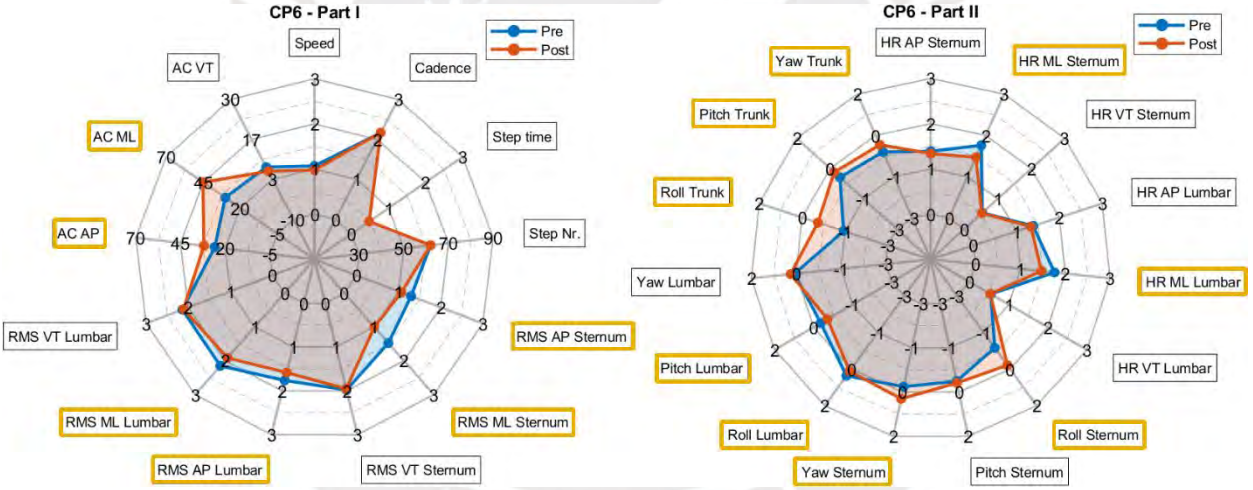


Figure 6.6 Radar plots for multiple parameters for CP6.

6.1. Spatiotemporal parameters

Appendix A provides details regarding the variation of specific spatiotemporal parameters for every child with CP. The children identified as CP1, CP4 and CP6 describe the highest variations in spatiotemporal parameters after the implementation of Vojta therapy. Therefore, in this section, the observed variations will be described and illustrated using box plots.

Figure 6.7 illustrates that CP1 exhibited a reduction in median walking velocity, a slight increment in cadence, and a minor increment in step time. From Appendix A.1, it is perceived a diminution in step length and an augmentation in the number of steps after

performing therapy. This suggests that CP1 completed a greater number of steps in a shorter time compared to before therapy.

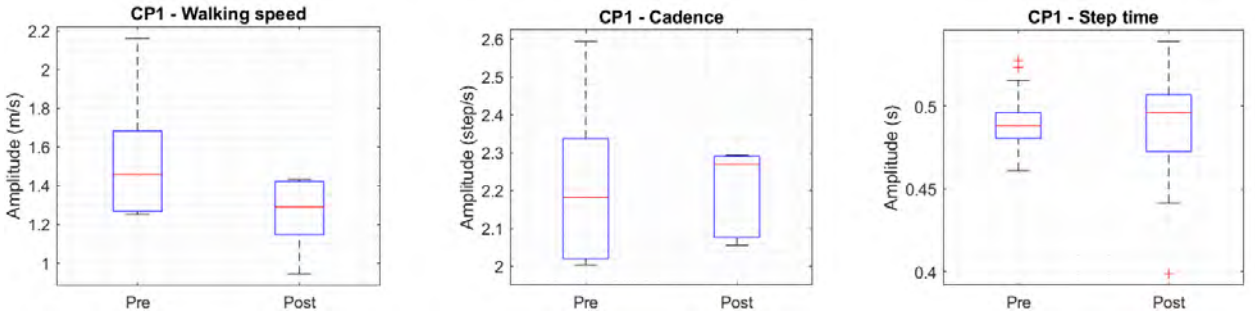


Figure 6.7 Spatiotemporal parameters for CP1.

In contrast, Appendix A.4 illustrates that CP4 demonstrated a great increase in gait speed and slight increase on median cadence, as well as a reduction in step time in post-therapy, when compared to the prior therapy data. This suggests that step length was slightly increased and that the number of steps performed was diminished in post-therapy.

Appendix A.6 illustrates that CP6 exhibited a markedly decreased gait speed with less data variability in post-therapy. Cadence, step time and number of steps are increased slightly, as well as the variability of the data distribution in cadence and step time box plots.

6.2. RMS acceleration

CP1 presents lower sternum AP RMS values and higher variability, significantly greater ML RMS median acceleration and more dispersion in post-therapy compared to pre-therapy data, as illustrated in Figure 6.8.

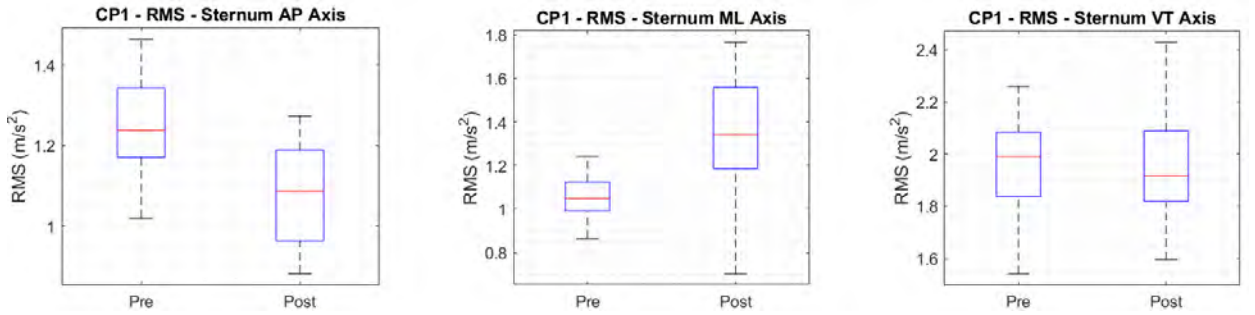


Figure 6.8 Sternum RMS Acceleration for CP1.

Figure 6.9 describes that, following therapy, CP1 shows a slightly higher lumbar AP RMS median and more dispersion, as well as a slightly decreased ML and VT RMS median in comparison to pre-therapy dataset.

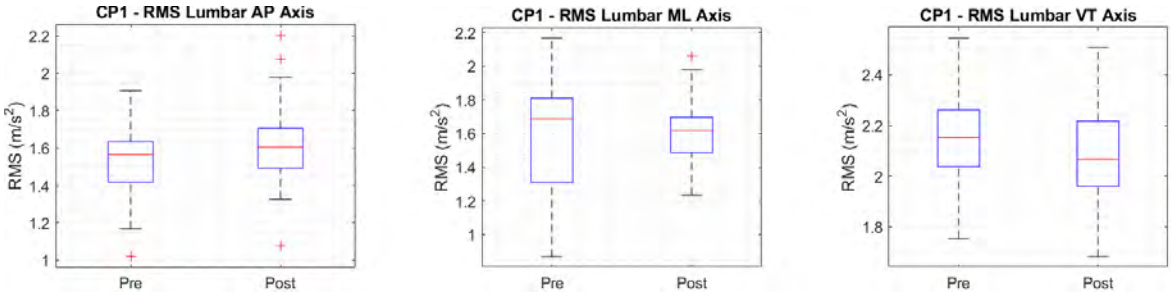


Figure 6.9 Lumbar RMS Acceleration for CP1.

Appendix B.2 presents the sternum RMS acceleration per axis and dataset. CP2 exhibits a slight increase in AP RMS median and reduced variability, a smaller ML RMS dispersion with a higher number of outliers, and a notable increase in VT RMS acceleration.

After treatment, CP2 demonstrates higher lumbar AP RMS median and less variability, slightly increase in ML RMS values with diminished dispersion, and a significant increase in VT RMS magnitude and higher variability.

Appendix B.5 illustrates the sternum sensor RMS acceleration data across all CP5 datasets. Following Vojta therapy, CP5 performed a notable reduction in the variability observed in the AP and ML axes, accompanied by a considerable decrease of ML RMS acceleration.

There is a notable decrease in variability across all axes for the lumbar RMS magnitudes. Furthermore, there has been a significant reduction in AP and ML RMS magnitude.

The CP6 post-therapy data exhibits a reduction in sternum median RMS across all axes, with a more pronounced decrease in variability. CP6 groups have also a markedly lower RMS data distribution and median. Refer to Appendix B.6 for further details.

Regarding the lumbar spine behavior, CP6 post-therapy data exhibited a decrease in AP and ML RMS median and variability of data distribution, accompanied by a slight reduction in VT axis variability.

6.3. Attenuation coefficient

Following therapy, the AC for CP1 demonstrates a notable decrease in both ML and VT axes, accompanied by increased dispersion of data. Furthermore, there is a slight increase in AP coefficient of attenuation, as illustrated in Figure 6.10.

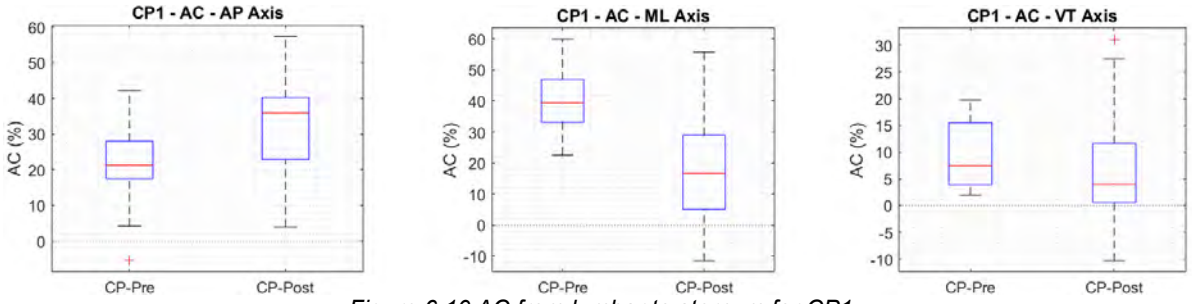


Figure 6.10 AC from lumbar to sternum for CP1.

The CP4 post-therapy dataset exhibited lower AP coefficient variability, a great decrease in ML AC and a slightly higher VT coefficient magnitude in comparison to pre-therapy, as shown in the boxplots of Appendix C.4.

Following therapy, Appendix C.5 details that CP5 demonstrated a minor reduction in AP and VT axes, accompanied by a slight increase in ML coefficient. The ML axis value after therapy displays that there is a great increase in the coefficient of attenuation.

Additionally, Appendix C.6 illustrates the CP6 performance after therapy, which resulted in a modest elevation in the coefficient in the AP axis, a significant augmentation of ML AC and an increase in the dispersion of the VT axis.

6.4. Harmonic ratio

Figure 6.11 display the CP1 dataset's harmonic ratio in sternum. CP1 post-therapy data harmonic ratio significantly decreased in AP axis and slightly increased in the other axes.

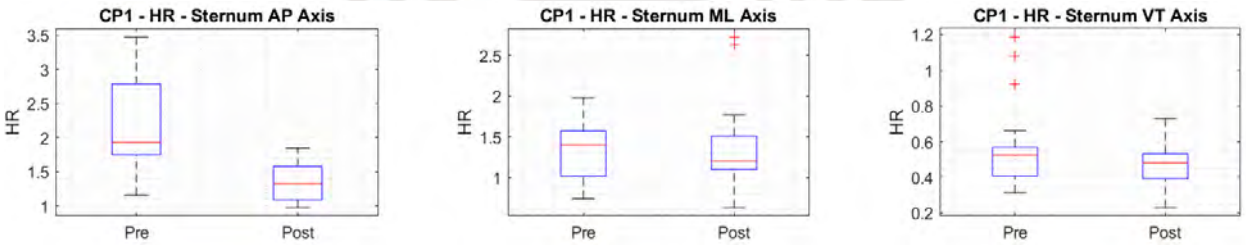


Figure 6.11 Sternum HR for CP1.

Figure 6.12 illustrates a greater lumbar ML ratio that is either comparable or lower. There has been a slight reduction in the AP and VT axes, and a slight increase in ML HR between the CP1 pre- and post-therapy datasets. Furthermore, there has been a greater variability in AP HR data distribution following Vojta therapy.

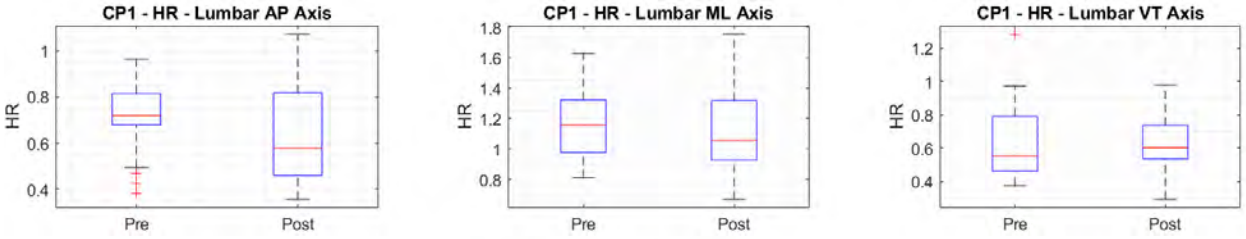


Figure 6.12 Lumbar HR for CP1.

Appendix D.3 details that, after Vojta therapy, the AP and ML HR median values and variability increased, while the VT HR exhibited a minor decrease compared to the CP3 pre-therapy. There is also a slight increase in the lumbar AP and ML HR in CP3 post-therapy, with a minor decrease in VT axis in the CP3 post-therapy dataset.

Appendix D.5 illustrates the comparable sternum HR median values between both CP5 datasets. It is only noticeable that CP5 post-therapy has slightly decreased median HR values along every axis. There is an increase in lumbar HR median and data distribution

in the CP5 post-therapy. Following therapy, this child demonstrates a significant increase in ML HR, as well as a slightly higher median HR on the VT axis.

As illustrated in the box plots from Appendix D.6, prior therapy, the CP6 child exhibited lower sternum HR values along the AP and VT axes. Conversely, on the ML axis, the median and HR distribution show slightly higher values than those observed post-therapy. The lumbar HR median values for both CP6 groups are similar across all axes, except for ML axis. Following therapy, CP6 demonstrated greater variability in the AP and VT axes, with less dispersion in the ML axis.

6.5. Absolute angles

In this section, the box plots for the first mentioned child are presented, and the plots for the other children are presented in Appendix E. To interpretate better the results, please refer to Table 5.1 to see the convention of absolute and relative angles.

Figure 6.13 represents the variation of the sternum angles pre- and post-therapy for CP1 child. Following therapy, a change of tendency in the roll angle is perceived, from a lateral flexion to the right to the left. The pitch and yaw angles present a slight variation near to 0°. Additionally, post-therapy dataset presents less outliers than pre-therapy data.

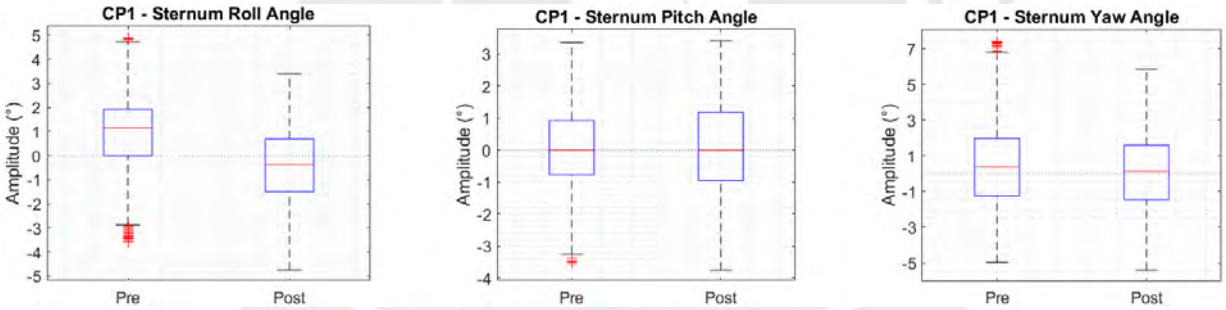


Figure 6.13 Sternum absolute angles for CP1.

Figure 6.14 illustrates a comparable or slightly minor difference between the median lumbar absolute angles from CP1. There is less data dispersion in roll and yaw angles, as well as more symmetry on the pitch angles distribution and outliers in the post-therapy dataset.

Appendix E.2 details the variation in sternum roll and pitch angles from CP2. After Vojta therapy, CP2 presents more symmetry in both angles, but especially in roll angles. The yaw angle presents a greater degree of positive asymmetry, indicating a tendency for a longitudinal trunk torsion to the left.

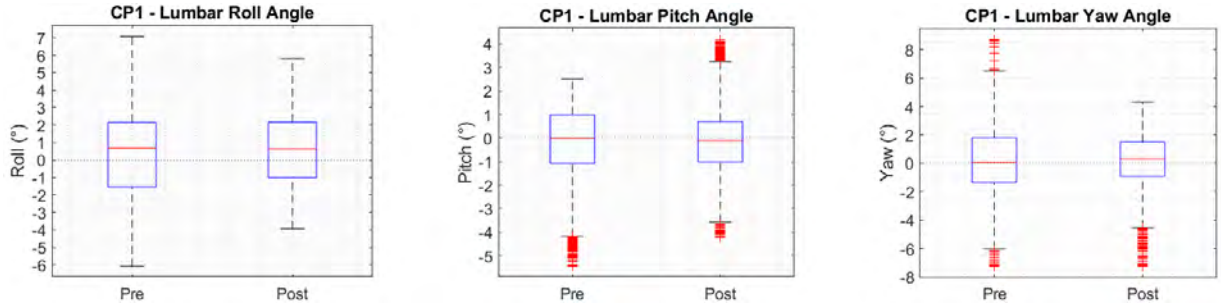


Figure 6.14 Lumbar absolute angles for CP1.

Following the treatment, a notable decrease of the roll median lumbar angle is appreciated through the box plots, as well as more dispersion and symmetry. However, the pitch and yaw angles exhibit a slight reduction of the median values.

With regard to CP4, Appendix E.4 illustrates a similar behavior of the angles data for pre- and post-therapy datasets. It is noticeable that the pitch angle following therapy exhibits greater data distribution and a greater degree of proximity to zero. There is also a slight increase in lumbar pitch and yaw median values after therapy. There is additionally more data distribution on roll and pitch angles in CP4.

CP6's absolute angles in Appendix E.6 exhibited a noticeable sternum roll and yaw angles increase and a clear pitch angles data distribution reduction. A great number of outliers are observed as well in pitch and yaw angles.

There is a slight reduction in roll and pitch angles after Vojta therapy, as well as a decrease in outliers. However, CP6 exhibits a positive asymmetry in roll and yaw angles, indicating a significant tendency towards lateral bending to the right and longitudinal torsion to the left.

6.6. Relative angles

CP1 presented a noticeable reduction of the trunk relative angles from the roll and yaw angles, as well as a minor increase in pitch median angle. There is also an increase in pitch angles, as well as higher dispersion and outliers.

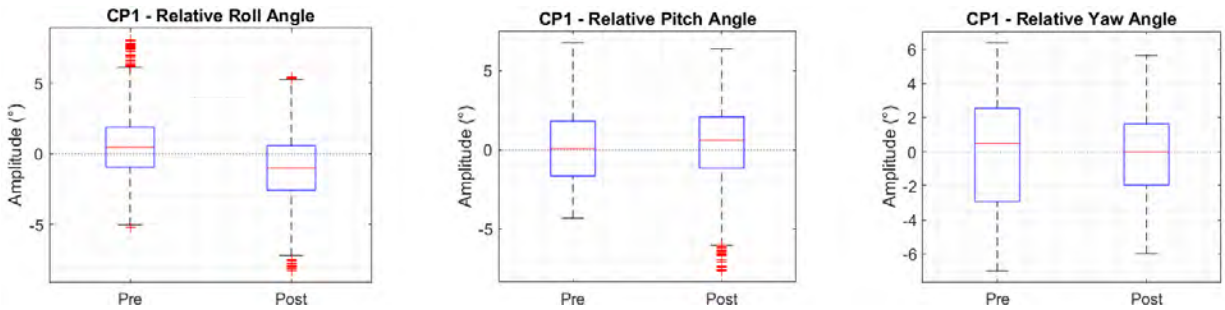


Figure 6.15 Trunk relative angles for CP1.

In Appendix F.2, CP2 displays a more pronounced increase in roll angle, accompanied by a slight variation in pitch and yaw. In general, CP2 exhibits a more symmetrical

distribution of pitch and yaw angles, as well as a significant negative tendency in roll angles.

Appendix F.6 illustrates that CP6 has a similar median pitch and yaw angles following therapy, as well as a noticeable increase in roll angle median. There is also a diminution of outliers in post-therapy angles and more symmetrical data behavior in roll and pitch relative angles.

7. Discussion

This chapter will carry out a comprehensive analysis of the findings by selected parameters. In order to achieve this objective, the uniqueness of each child and their condition will be examined. This is due to the heterogeneity of clinical conditions and age groups within the children with CP. Thus, the unique features of each participant will be portrayed using careful reasoning and interpretation of findings.

The following subchapters present the separated by categories. RMS acceleration and AC (each axis) characterize the smoothness of gait, whilst the HR represents the harmony or rhythmicity of gait. The angular motion given by lateral bending, sagittal bending and longitudinal torsion are represented by the absolute and relative angles in the AP, ML and VT axes, respectively.

7.1. Gait smoothness

The CP1 RMS box plots from Appendix B.1 illustrate that there is a decrease in the sternum AP RMS from a median of 1.24 m/s^2 to 1.09 m/s^2 in post-therapy. The lumbar AP RMS, on the other hand, slightly increased from 1.07 m/s^2 to 1.34 m/s^2, as well as the elevation of the whiskers. CP1 presents lower sternum AP acceleration magnitudes are lower, which resembles smoother motion, and the lumbar AP magnitudes are slightly higher, indicating slightly low shock-absorption ability.

Appendix C.1 indicates that the median AP AC value increased (pre: 21.35%, post: 35.94%) and its maximum value increased 15.32% in post-therapy dataset. This indicates that there is a higher tendency for greater acceleration reduction from lumbar spine to the sternum. Possibly it may be indicative of a more controlled sternum movement for the post-therapy dataset in this direction.

The sternum ML RMS median value notably increased from 1.05 m/s^2 to 1.34 m/s^2 and the lumbar ML RMS slightly decreased from 1.69 m/s^2 to 1.62 m/s^2. The sternum ML RMS distribution increased, showing asymmetrical behavior towards higher RMS values. In the contrary, lumbar distribution decreased. The high increase on the sternum RMS may indicate noticeable higher ML acceleration magnitudes and, therefore, the motion in this direction exhibits a markedly reduced degree of smoothness and a greater impact.

The ML AC drastically reduced, which may be correlated with the sternum ML RMS. When the AC is positive, it shows a reduction from acceleration between the lower to upper reference, which in this case is the attenuation from lumbar spine to the sternum. When the AC is negative, it shows an increment of the sternum acceleration with respect to the lumbar spine.

The ML AC median decreased (pre: 39.35%, post: 16.6%), accompanied also by a diminution from the minimum AC value from 22.53% to -11.53%. This can be interpreted as less ML acceleration being attenuated and allowing CP1 to do more erratic movements. The ML RMS asymmetry towards higher values may explain why there is less attenuation of acceleration in this direction.

The sternum VT RMS demonstrates a diminution from the median value from 1.99 m/s^2 to 1.92 m/s^2, accompanied by a higher extension of the maximum value (2.43 m/s^2). The lumbar VT RMS exhibits a similar behavior, with a median decrease from 2.15 m/s^2 to 2.07 m/s^2 and a slight diminution in the data distribution.

There is a decrease in CP1 VT AC median from 7.44% to 4%, as well as an increase in data distribution from a minimum of -10.33% to a maximum of 27.43%. The reduction of this parameter may be related to the high extension of the sternum VT accelerations and the minimal diminution of the lumbar VT magnitudes.

Regarding CP2, Appendix B.2 illustrates a comparable median sternum (pre: 0.84 m/s^2, post: 0.89 m/s^2) and a slightly increased lumbar median AP RMS acceleration is appreciated (pre: 1.18 m/s^2, post: 1.35 m/s^2), respectively. Additionally, there is also a decrease in AP RMS distribution in both locations, with the major concentration of data in higher RMS values. Appendix C.2 shows a slight diminution of the median AP AC (pre: 38.87%, post: 37.82%), accompanied by higher data variability.

Following therapy, CP2 exhibits comparable ML RMS accelerations for the sternum and lumbar spine (pre: 0.71 m/s^2, post: 0.72 m/s^2). The only difference is the decrease in variability for both data distribution, with a higher concentration of data in lower RMS values. There is a diminution of ML AC (pre: 48.2%, post: 37.52%), accompanied by a similar reduction in ML AC magnitude and a higher data variability.

There is a substantial increment in the VT RMS median values in the sternum (pre: 1.14 m/s^2, post: 1.44 m/s^2) and in the lumbar (pre: 1.17 m/s^2, post: 1.45 m/s^2). CP2 has also decreased the VT AC median (pre: 3.53%, post: -0.02%) and its data distribution with a minimum value of -15.05% and a maximum of 13.56%. However, the majority of the data is concentrated within the positive value range.

An increase in VT RMS can only be achieved when the child performs the gait with more movement intensity or vibrations in the VT axis. It may suggest greater impact on the feet contact to the ground and greater VT acceleration variation in the trunk sensors, even though there is an attenuation of acceleration in the trunk.

Regarding CP3, Appendix B.3 illustrates the decrease in AP RMS median value in both sternum (pre: 1.1 m/s^2, post: 1 m/s^2) and lumbar areas (pre: 2.43 m/s^2, post: 2.2 m/s^2),

with a decrease in RMS variability. CP3 then presents a greater concentration of AP accelerations in lower magnitudes, which may suggest a reduced possibility of abrupt movements and a tendency towards smoother movement.

Appendix C.3 illustrates a minimal reduction in AP AC median (pre: 54.43%, post: 54.06%), accompanied by an augmentation of the AC distribution. Following therapy, CP3 presents an AC minimum value of 27.64%, a maximum of 61.71% and an asymmetry to higher values.

Therefore, it can be concluded that there is a greater possibility that some gait cycles exhibited less attenuation of the acceleration. However, a significant quantity of cycles tends to exhibit a reduction of acceleration similar to that observed in pre-therapy dataset.

There is an increase in the ML RMS median for both sternum (pre: 0.84 m/s^2, post: 0.9 m/s^2) and lumbar (pre: 1.46 m/s^2, post: 1.69 m/s^2). CP3 presents also a higher RMS variability in both regions. Following therapy, Appendix C.3 illustrates that CP3 presents also a slightly higher ML AC (pre: 42.44%, post: 47.39%). In addition, the AC data variability seems also bigger.

Although CP3 may exhibit more intense movements, particularly in the lumbar spine, there is an indicative of a major ML acceleration attenuation in the trunk. This may allow CP3 to perform a higher, but controlled, lateral balancing motion.

Regarding VT RMS, there is a reduction in sternum (pre: 1.73 m/s^2, post: 1.56 m/s^2) and lumbar (pre: 2.06 m/s^2, post: 1.86 m/s^2) acceleration, accompanied by a slight augmentation of the data distribution. The RMS reduction suggests that the gait is performed with less turbulent motions and greater shock absorption capacity.

Appendix C.3 illustrates a minor decrease in VT AC from a median value of 17.2% to 15.63% after therapy. There is slightly reduced data distribution and improved symmetric behavior of the AC.

Regarding CP4, Appendix B.4 shows a noticeably decrease in AP RMS magnitude variability after Vojta therapy, accompanied by a comparable median AP RMS value in both sternum (pre: 0.86 m/s^2, post: 0.88 m/s^2) and lumbar spine (pre: 1.63 m/s^2, post: 1.62 m/s^2). Furthermore, Appendix C.4 indicates a minor increase in AP AC median (pre: 45.9%, post: 46.34%), data variability and greater asymmetry to higher values, with an increase of the maximum value (pre: 59.53%, post: 65.12%).

In summary, there are slightly lower AP accelerations with a tendency towards higher attenuation. Thus, CP4 may exhibit slightly smoother movements in the AP direction.

CP4 demonstrates a slight ML RMS increase in both sternum (pre: 0.71 m/s^2, post: 0.95 m/s^2) and lumbar magnitudes (pre: 1.54 m/s^2, post: 1.61 m/s^2). Lumbar ML RMS data distribution is also higher, with a maximum value that reaches 2.1 m/s^2.

There is also a great decrease in the ML AC median (pre: 53.64%, post: 39.87%), with a significant increase in variability. The degree of attenuation of ML accelerations is markedly reduced since the AC values ranges from a minimum of 22.65% to a maximum of 59.62% following therapy. CP4 exhibits a slightly greater acceleration amplitude in both regions, which may be interpreted as minimal jerky movements. These may also be part of the low AC values.

There is a comparable performance in both the median VT RMS values from the sternum (pre: 1.28 m/s^2, post: 1.45 m/s^2) and lumbar spine (pre: 1.53 m/s^2, post: 1.82 m/s^2). Post-therapy RMS values are significantly higher than those from pre-therapy. It is perceived a minor increase in the VT AC median (pre: 15.64%, post: 18.32%) and data variability, with a slight tendency towards higher VT acceleration attenuation.

CP4 demonstrates similar behavior to CP2 only in the VT RMS magnitudes, since CP4 exhibits higher values of attenuation (minimum: 15.64%, maximum: 18.32%). Thus, CP4 may exhibit greater impact in the VT accelerations and lower shock-absorption between the lower limbs and the trunk.

In Appendix B.5, CP5 demonstrates a decrease in the lumbar AP RMS median (pre: 1.67 m/s^2, post: 1.43 m/s^2) and data distribution, as well as a comparable behavior in sternum AP RMS median (pre: 1.01 m/s^2, post: 1 m/s^2) with a slight reduction in variability. Appendix C.5 displays a slight decrease in AP AC in both median (pre: 36.78%, post: 32.47%) and values distribution. Post-therapy AC is distributed from a minimum of -5.91% to a maximum of 53.27%.

CP5 presents a noticeable diminution in the sternum (pre: 1.59 m/s^2, post: 1.32 m/s^2) and a slight decrease in the lumbar ML RMS median (pre: 1.59 m/s^2, post: 1.49 m/s^2), as well as a decreased data distribution towards lower ML values.

There is a great increase in CP5 ML AC (pre: -2.21%, post: 15.65%), as well as a significant decrease in data variability, from a minimum of -15.79% to a maximum of 36.14%.

There is a comparable behavior between sternum (pre: 1.49 m/s^2, post: 1.48 m/s^2) and lumbar (pre: 1.67 m/s^2, post: 1.65 m/s^2) VT RMS median values. Lumbar VT RMS distribution also decreased, whilst sternum values are slightly increased towards high RMS values after therapy.

Regarding VT AC, there is a decrease in the median value (pre: 16.1%, post: 11.69%) and data distribution. The post-therapy VT AC exhibits a minimum of -4.67% and a maximum of 21.26%.

Regarding CP6, Appendix B.6 displays a slight decrease in AP RMS in both sternum (pre: 1.23 m/s^2, post: 1.03 m/s^2) and lumbar spine (pre: 1.72 m/s^2, post: 1.55 m/s^2), with a decrease in the RMS variability. Appendix C.6 exhibits a slight increase in the AP AC median (pre: 27.75%, post: 32.92%) and data distribution.

There is a slight decrease in the ML RMS median values from the sternum (pre: 1.47 m/s^2, post: 1.01 m/s^2) and lumbar (pre: 2.09 m/s^2, post: 1.87 m/s^2), as well as a reduction in data distribution. Regarding the AC, there is an increase in both median (pre: 27.36%, post: 47.42%) and data dispersion.

CP6 exhibits a decrease in the VT RMS data variability in both regions, accompanied by a comparable median in the sternum (pre: 1.98 m/s^2, post: 1.95 m/s^2) and lumbar spine (pre: 2.13 m/s^2, post: 2.08 m/s^2). On the other hand, the VT AC has increased slightly in dispersion and negative coefficients, whilst the median decreased (pre: 8.4%, post: 7.73%).

7.2. Gait harmonicity

Appendix D.1 illustrates the marked decrease in the sternum AP HR in the median (pre: 1.93, post: 1.32) and the mean (pre: 2.21, post: 1.33), accompanied by a great reduction in data distribution. There is also a decrease in AP HR lumbar median (pre: 0.72, post: 0.58), as well as a noticeable increase in the lumbar data distribution.

It is important to note that a large HR implies more harmonic movement. Then, CP1 exhibits fewer harmonic movements in the sternum and lumbar after therapy, which would translate into more irregular movements in the AP axis.

Furthermore, CP1 has an increase in the ML HR median from the sternum (pre: 1.22, post: 1.4) and the lumbar (pre: 1.02, post: 1.16), with an increase in the HR distribution towards high HR values. This may indicate that there is presence of more harmonic movements in both regions following therapy.

The VT HR median values had slightly increased in the sternum (pre: 0.48, post: 0.51) and decreased in the lumbar spine (pre: 0.6, post: 0.56). Both data distributions have higher data concentration in lower VT HR values. These variations do not provide sufficient information for a meaningful interpretation of the comparison before and after therapy.

Appendix D.2 indicates an increase in the AP HR median values in both the sternum (pre: 1, post: 1.05) and the lumbar spine (pre: 0.47, post: 0.56). However, the sternum HR variability decreased, whilst lumbar HR distribution increased (minimum: 0.38, maximum: 0.83). Thus, there is a slight increase in the degree of harmony exhibited in the lumbar spine in the AP direction after therapy.

Regarding ML HR, CP2 exhibits a slight decrease in both sternum (pre: 1.44, post: 1.38) and lumbar spine (pre: 1.29, post: 1.28). There is also a decrease in both data distributions with a bigger tendency towards lower HR values.

There is a slight diminution in the sternum VT HR (pre: 0.45, post: 0.4) and great decrease in the lumbar VT HR (pre: 0.63, post: 0.44). Both HR distributions have a tendency towards lower HR values than pre-therapy.

These observations suggest that CP2 exhibits a slight increase of smoothness in the AP sternum motion, and a noticeable reduction in the damping of vertical acceleration from the lumbar spine to the sternum.

Appendix D.3 illustrates a minor augmentation in the median AP HR values of sternum (pre: 0.86, post: 0.96) and lumbar (pre: 0.5, post: 0.57). Lumbar AP HR exhibits a reduction in the data variability with a major tendency towards greater HR values. Then, CP3 demonstrates slightly harmonic movements compared to the pre-therapy state.

CP3 exhibits an elevation in both the median and data variability from sternum ML HR (pre: 0.75, post: 1.16), with the presence of some outliers on higher HR values. The lumbar ML HR exhibits an increase in the median (pre: 1.24, post: 1.34), and a higher asymmetry at higher HR values. Consequently, both regions may demonstrate more harmonic movements than before therapy, which are more perceptible in the sternum.

CP3 exhibits comparable behavior in sternum VT HR in the median (pre: 0.54, post: 0.53) and data variability. In the contrary, there is a noticeable decrease in lumbar VT HR in both median (pre: 0.82, post: 0.66) and data distribution values.

Appendix D.4 illustrates that the sternum and lumbar spine AP HR variability has slightly increased. Sternum HR exhibits a minor decrease in the median (pre: 0.87, post: 0.78) as well as an increase in the asymmetry to higher HR values. Lumbar HR displays a minor increase in the median value (pre: 0.57, post: 0.61), accompanied by an incremental asymmetry in higher HR AP values.

There is a minimal decrease in both ML HR median sternum (pre: 1.02, post: 0.88) and lumbar (pre: 1.17, post: 1.05) and data distribution. CP4 exhibits a minor decrease in gait movement harmony and stability in the ML direction.

CP4 exhibits a higher VT HR data variability with more asymmetry towards high HR values for both regions, with a decrease in the sternum median (pre: 0.63, post: 0.61) as well as an increase in the lumbar median (pre: 0.85, post: 0.87).

Appendix D.5 displays a decrease in the sternum AP HR median (pre: 0.95, post: 0.91) and an increase of data distribution with tendency to lower HR values. There is also an augmentation of the lumbar AP HR median (pre: 0.87, post: 0.78) and a higher dispersion with a maximum of 1.14. Following therapy, CP5 demonstrates a rise in lumbar motion harmony.

There is a great reduction in the data distribution from both sternum and lumbar ML HR, accompanied by an increase in the sternum median (pre: 1.84, post: 1.64) and a decrease in the lumbar spine median (pre: 1.34, post: 1.58).

CP5 exhibits an increase in the VT HR data variability in both regions. However, there is a slight decrease in the sternum median (pre: 0.59, post: 0.54) and a slight increase in the lumbar VT HR median (pre: 0.68, post: 0.70). As a result, CP5 denotes slightly better lumbar harmonic gait patterns.

Appendix D.6 details a notable increase in sternum and lumbar AP HR values distribution with greater asymmetry towards high HR values. However, the sternum median decreased (pre: 1.35, post: 1.19), while the lumbar median exhibited a comparable value (pre: 0.77, post: 0.78). Therefore, this may suggest that CP6 exhibits less jerky and smoother movements, which would result in slightly better harmony in the gait exhibited.

CP6 exhibits a decrease in the sternum ML HR median (pre: 1.66, post: 1.41) and an increase in its data variability. However, the lumbar displays a slight increase (pre: 1.08, post: 1.14) and a decrease in data variability. In addition, CP6 exhibits an increase in VT HR data variability and a decrease in the median values for the sternum (pre: 0.53, post: 0.49) and lumbar spine (pre: 0.66, post: 0.59).

7.3. Lateral bending

The roll angle corresponds to an angular displacement on the AP axis. Appendix E.1 displays that CP1 presents a decay from a sternum mean value of 1.14° to -0.36° after completing therapy. It is also perceived a major tendency from the roll angle distribution towards a negative value, including the median. The lumbar median exhibited a slight decrease from 0.68° to 0.64°, accompanied by a noticeable decrease in the extension of the whiskers.

The trunk relative angles also demonstrate the same tendency as the sternum. The median value of this parameter is 0.47° prior therapy and is -1.02° following Vojta therapy.

The relative roll angles exhibit negative asymmetry, which reinforces the sternum roll absolute angle performance following physiotherapy.

According to the proposed angle convention, a negative roll angle indicates a lateral bending to the left. Therefore, CP1 exhibits a marked lateral bending of the sternum to the left with respect to the global frame and to the lumbar spine. The presence of outliers in the relative roll plot indicates the occurrence of abrupt lateral bending motions to both sides.

As illustrated in Figure 6.2, the sternum mean roll angle increased, the lumbar mean decreased, and the relative mean roll angle also increased following therapy. This behavior is also appreciated in the box plots from Appendix E.2.

A more symmetric pattern of the angle's distribution is also appreciated, accompanied by the increase of the sternum roll median (pre: -0.84°, post: 0.13°) and the decrease of the lumbar roll median (pre: 1.64°, post: 0.34°). There is also an increase in the trunk relative roll median (pre: -2.32°, post: -0.91°).

Prior to the beginning of therapy, the sternum slightly tended to bend laterally to the left, whilst the lumbar spine had a propensity to bend to the right. Vojta therapy enabled CP2 to perform more symmetrical lateral bending movements.

There is also the complete removal of outliers after therapy in lumbar absolute and trunk relative angles, whilst the sternum only presents one negative outlier. This may suggest that CP2 exhibited more controlled lateral bending motions.

Regarding CP3, Appendix E.3 displays a reduction in the sternum (pre: 0.36°, post: 0.05°) and lumbar (pre: 0.45°, post: 0.06°) roll median angle. In both cases there is also greater angle variability, accompanied by asymmetries to negative and positive values, respectively. CP3 exhibits a slight increase in the relative roll median (pre: -1.29°, post: -1.07°) and higher data distribution with negative asymmetry.

Post-therapy absolute roll angles exhibit a lateral bending behavior concentrated close to 0°, accompanied by a slight affinity for lateral bending to the left for the sternum and to the right for the lumbar spine. It can also be inferred that the trunk exhibits a slight increase in lateral balancing, with a tendency towards greater lateral bending to the left.

Regarding CP4, Appendix E.4 illustrates a minimal decrease and increase in the median angle value of the sternum (pre: 0.82°, post: 0.71°) and lumbar spine roll (pre: -0.88°, post: -0.71°), respectively. Sternum and lumbar roll exhibits comparable positive asymmetrical behavior and less outliers in the post-therapy dataset.

There is also an increase in the relative roll angles distribution in comparison to the pre-therapy dataset. It also exhibits a positive asymmetry following therapy, as well as similar median roll angle (pre: 1.53°, post: 1.52°).

To summarize, the sternum and lumbar exhibit a slight proclivity towards a right lateral bending, however, the sternum presents more abnormal sternum movements in the opposite direction. The relative angles indicate a lateral bending with an asymmetry towards the right side.

Appendix E.5 illustrates that there is a minor increase in the sternum (pre: -1.65°, post: -1.37°) and the lumbar (pre: 0.4°, post: 0.82°) roll angle medians. Sternum angles possess less quantity (pre: 210, post: 116) and extension of outliers. Additionally, the lumbar spine exhibits slightly better symmetry, with still a tendency towards positive value.

There is a decrease in the relative roll angle distribution with a comparable median (pre: -2.42°, post: -2.3°), accompanied by a minor rise in the number of positive outliers (pre: 204, post: 213).

Therefore, since CP5 exhibits a high quantity of data from a negative relative angle, CP5 demonstrates a tendency towards a sternum lateral bending to the left, even though this proclivity is smaller than pre-therapy data. Additionally, the lumbar roll angles indicate slightly better symmetry, with a minor proclivity towards a lateral bending to the right.

The trunk roll angles demonstrate a tendency towards a left lateral bending, with a greater tendency to abnormal lateral bending to the right.

Regarding CP6, Appendix E.6 indicates an increase in sternum (pre: -0.62°, post: 0.31°) and lumbar (pre: 0.74°, post: 0.41°) roll median angle, approaching closer to the zero-degree angle. There is also a reduction in the range of sternum angle distributions and an increase in those of the lumbar region, accompanied by the complete elimination of outliers.

CP6 exhibits a noticeable increase in the relative roll median (pre: -1.61°, post: -0.47°) and a more symmetrical behavior (minimum: -6.45°, maximum: 6.77°). However, there is the presence of more outliers in positive angle values. Following therapy, CP6 exhibits a relative trunk movement with a greater proclivity to a lateral bending to the left with abnormal bending jerky movements to the right.

7.4. Sagittal bending

Appendix E.1 suggests a minor increase of variability on all pitch angles from CP1, which indicates a better symmetry in the sagittal bending. In addition, there is a minor rise in the pitch median and higher symmetry on the whiskers' extension of the relative angles box

plot, suggesting a more symmetric sagittal bending as well with presence of outliers at the extension of the trunk.

The presence of outliers on a single side may be interpreted as the appearance of erratic or abnormal movements on the sternum with respect to the lumbar spine when the trunk is extended. The elevated dispersion of the absolute and relative pitch angles may indicate enhanced trunk flexibility during sagittal bending. Consequently, CP1 exhibits also a minor reduction in trunk flexion due to the more symmetrical movements.

As illustrated in Appendix E.2, sternum pitch angle distribution presents more symmetry compared to pre-therapy data, as well as a decrease of the median value (pre: 0.5°, post: 0.17°). There is presence of outliers to both flexion and extension movements.

There is an increase in median lumbar pitch angle (pre: -0.03°, post: 0.29°) and data distribution, accompanied by less negative outliers and more symmetric behavior in post-therapy.

There is a diminution of the median relative pitch angle (pre: 0.43°, post: -0.06°), accompanied by an increase in symmetry in the data distribution and, unfortunately, a greater presence of outliers on both sides.

Therefore, these findings may suggest that there is a slight tendency to flexion in both anatomical locations, with a minor decrease in flexion in the sternum and a slight increase in lumbar. CP2 exhibits a balanced and symmetric trunk sagittal bending movement, with the presence of some abnormal movements in both trunk flexion and extension.

Appendix E.3 demonstrate that the median pitch angles from the sternum (pre: 0.06°, post: -0.01°) and lumbar (pre: 0.94°, post: 0.52°) are closer to a zero-value. Sternum presents more symmetrical behavior (minimum: -3.39°, maximum: 3.22°), whilst lumbar spine exhibits a slightly negative asymmetrical performance with lower data dispersion (minimum: -5.06°, maximum: 4.45°).

The median relative trunk pitch angles in both datasets are comparable (pre: -0.66°, post: -0.64°). Following therapy, the relative pitch parameters exhibit a slight reduction of data variability and minor increase in symmetry, accompanied by a slight tendency towards trunk flexion.

The relative angles exhibited that CP3 slightly tends toward a trunk flexion in pre- and post-therapy datasets, however, this behavior is better controlled after therapy. Absolute angles in pre-therapy suggest a greater sternum and lumbar extension movement, whilst post-therapy data provides a more symmetrical sagittal bending performance.

Appendix E.4 illustrates an increment in the sternum and lumbar spine pitch angle variability, accompanied by a slight decrease in the sternum median (pre: 0.15°, post: 0°)

and a minor increase in the lumbar spine median (pre: -0.06°, post: 0.33°). Both regions display less symmetrical behavior, accompanied by a negative asymmetry in the sternum (minimum: -3.38°, maximum: 2.61°) and negative outliers. In addition, the lumbar spine exhibits a positive asymmetry (minimum: -3.6°, maximum: 5.36°).

In addition, CP4 exhibits a slight decrease in the relative pitch median (pre: 0.3°, post: -0.28°) and data distribution, however, it still has a positive asymmetry (minimum: -4.7°, maximum: 5.34°). This may be indicative of a slight trunk flexion movement, accompanied by a minor tendency for the lumbar flexion and abrupt or abnormal sternum extension movements.

Appendix E.5 provides information regarding CP5 performance. There is a more symmetric pattern in sternum pitch whiskers (minimum: -3.6°, maximum: 3.47°), as well as slightly higher data distribution. Additionally, there is the presence of an increased number of outliers (pre: 13, post: 38). The lumbar pitch median has also increased (pre: -0.11°, post: 0.18°), as well as the data variability with a greater negative asymmetry (minimum: -5.97°, maximum: 3.12°) and less outliers (pre: 41, post: 5).

The relative pitch angles exhibit a decrease in the median trunk pitch (pre: 0.59°, post: -0.02°). Also, there is no presence of outliers following therapy but there is a minor positive asymmetry (minimum: -6.18°, maximum: 7.43°).

CP5 may exhibit a minor tendency towards a slight increase in lumbar extension, with the presence of some erratic extension movements. Furthermore, CP5 displays greater relative trunk sagittal bending magnitude, accompanied by an asymmetry that suggests slightly greater trunk flexion movements.

Appendix E.6 illustrates a comparable lumbar pitch angle distribution and a great reduction in the sternum pitch distribution. There is a minimal increase in sternum median (pre: -0.03°, post: 0.07°) and a decrease in the lumbar median (pre: -0.1°, post: -0.51°). Also, there is an increase in the number of outliers on both sides of the sternum angles (pre: 116, post: 181).

CP6 presents a reduction in trunk relative angles distribution and the outliers' number (pre: 121, post: 39), which are now only located in a negative value. The median value has also shifted away from the 0° reference value (pre: -0.13°, post: 0.37°); however, the whiskers denote symmetric behavior (minimum: -6.25°, maximum: 6.29°).

In general, CP6 may demonstrate a discernible reduction in sternum sagittal bending with erratic movements on both sides following the treatment. CP6 exhibits a more symmetric sagittal bending performance with abrupt bending movements towards a trunk extension.

7.5. Longitudinal torsion

CP1 demonstrated more symmetry than pre-therapy on the sternum yaw angle with the median near to the zero-value. It is also notable that there is a great decrease in lumbar yaw angle distribution, as well as more negative asymmetry and a greater quantity of negative outliers. The relative trunk angle exhibited a more symmetrical behavior compared to pre-therapy (minimum: -5.99°, maximum: 5.66°).

It can then be concluded that CP1 exhibited a minor lumbar longitudinal torsion to the left, given that the relative angles are calculated using the lumbar spine as a reference point.

CP2 exhibits a more positive asymmetrical distribution of sternum yaw angles, with the decrease of the median value (pre: -0.1°, post: -0.22°). There is also a slightly decreased lumbar median value (pre: 0.51°, post: -0.14°). In addition, both sternum and lumbar yaw distributions have a bigger extension towards positive yaw angles. This may indicate that both anatomical positions exhibit similar behaviors, with a higher positive asymmetry, which means a slightly higher left side longitudinal torsion.

CP2 demonstrates a more symmetrical relative yaw angle behavior, accompanied by the increase of the median value (pre: -0.69, post: -0.06°). Nevertheless, the dispersion of data ranges from -7.25° to 6.09°, indicating that CP2 exhibited a more symmetrical movement of the sternum and lumbar areas, but with a minor tendency to rotate axially to the right.

In Appendix E.3, CP3 exhibits a slight decrease in sternum median value (pre: 0.22°, post: -0.2°) and increase in lumbar median (pre: 0.3°, post: 0.4°). Moreover, both regions also display an increment in the yaw angle distribution, accompanied by a negative asymmetry that is more prominent in the sternum. Also, there is notable presence of outliers with a greater prevalence on the negative side in the sternum and on both sides in the lumbar spine.

CP3 exhibits also a great increase in the asymmetry of the relative trunk yaw angle (minimum: -7.88°, maximum: 10.86°), accompanied by a slight decrease in the median yaw (pre: 0.14°, post: -0.4°).

Therefore, CP3 demonstrates a tendency towards a longitudinal torsion to the left, which is higher following therapy. The yaw absolute angles may indicate that CP3 displays more abnormal torsional motion to the left in both the sternum and lumbar, accompanied by sudden and abrupt torsional movements to the right.

Appendix E.4 provides information of great contrast to the pre- and post-therapy sternum yaw angles from CP4. The sternum yaw angles exhibit a more symmetrical distribution

as well as a similar median (pre: 0.26°, post: 0.24°). It is also remarkable that there is no presence of outliers, whilst the number of outliers in pre-therapy was 18.

Additionally, the lumbar yaw angles demonstrate a smaller data distribution, accompanied by a negative asymmetry (minimum: -9.28°, maximum: 5.82°). There is also a slight increase in the lumbar yaw median (pre: -0.05°, post: 0.21°).

Regarding the relative yaw angles, the median has decreased slightly (pre: -0.32°, post: -0.49°) and greater data distribution with positive asymmetrical behavior (minimum: -9.07°, maximum: 13.93°).

It can be concluded that, after treatment, CP4 sternum exhibits a more symmetric yaw angles oscillation, and the lumbar spine has a slight tendency to perform a torsion to the right. Because of this behavior, the relative angle between the sternum and the lumbar region manifests a higher proclivity towards a torsional deviation to the left.

Appendix E.5 displays that CP5 exhibits a decrease in the sternum (pre: -0.24°, post: 0.74°) and lumbar (pre: 0.73°, post: 0.34°) yaw median angle. There is also a change in data dispersion: a slight increase in sternum variability and a decrease in lumbar variability. There is also a small increase in relative yaw median (pre: -0.61°, post: -0.27°), and a slightly positive asymmetry (minimum: -7.48°, maximum: 8.32°).

Thus, CP5 may exhibit a minor increase in the flexibility in longitudinal torsion, with a slight tendency towards a torsion to the right side. In contrast, the relative angles suggest a slight proclivity for the sternum to rotate axially to the left with respect to the lumbar.

Appendix E.6 displays that CP6 has an increase in the sternum median (pre: 0.28°, post: 0.65°) and the appearance of outliers (post: 66) on both sides, accompanied by a diminished angle distribution. The lumbar yaw exhibits a more asymmetrical pattern towards positive yaw angles (minimum: -6.95°, maximum: 10.23°), as well as a median value close to the zero-degree value.

Furthermore, there is an increase in the median relative yaw (pre: -0.32°, post: 0.49°) and more asymmetrical behavior with a higher proclivity to negative yaw angle values (minimum: -14.1°, maximum: 10.62°). There is also the appearance of outliers on the negative side.

CP6 exhibits a proclivity to a longitudinal torsion to the left, which is higher in the lumbar spine, accompanied by a presence of abnormal torsion movements in both sides. The trunk relative angle gives insight into a greater degree of asymmetry, with a tendency towards torsion to the right. Also, some unusual or abnormal right-side torsion movements are observed.

7.6. Performance after treatment

CP1 exhibits two significant differences: smoother sternum AP and jerkier motions in both lumbar AP and sternum ML. However, CP1 demonstrates slightly more harmonic movements in the ML direction and a great decrease of harmony in the AP axis, for both regions. Although CP1 sternum ML accelerations represent more erratic or high-impact movements, the ML motion appears to be slightly smoother. Meanwhile, the smooth sternum AP movement also have a great decrease in harmony. Thus, CP1 may exhibit more controlled AP motion but still it may not demonstrate a rhythmic pattern of motion.

Regarding angular motion, CP1 presents a noticeable lateral bending to the left, a minor reduction of trunk flexion and a minimal trunk longitudinal torsion to the right. From the first two types of trunk movement, it was concluded that there is an increase of trunk flexibility on the lateral oscillation and the trunk extension, respectively. This behavior can be appreciated in Figure 7.1 minimum angles radar plot.

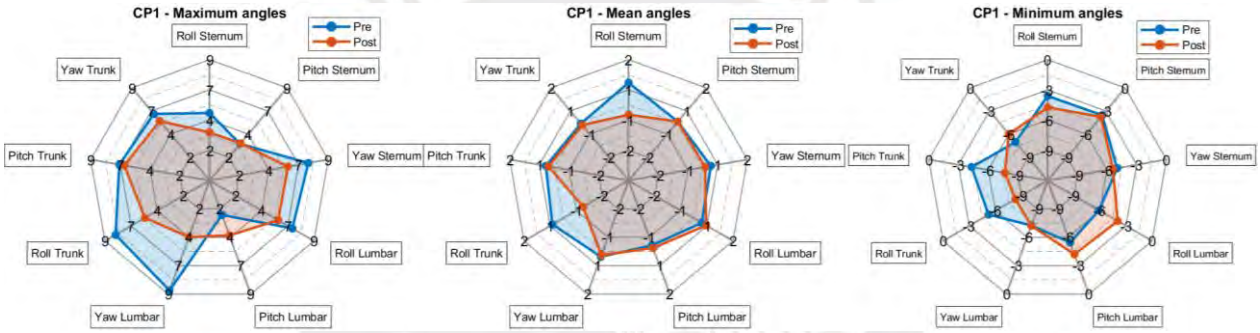


Figure 7.1 CP1 maximum, mean and minimum absolute and relative angles per axis.

The inclination or left lateral bending of the trunk interpretated from the parameters studied may indicate a compensatory trunk movement to the left side of CP1 body and, therefore, compensation for a probable right leg movement deficiency.

According to the parameters studied, CP2 demonstrates greater lumbar AP accelerations with higher degrees of harmony. CP2 exhibits also an increase in VT RMS in both regions, with a reduction of harmony more prominently on the lumbar spine. This may indicate high impact but controlled and harmonious accelerations in the AP direction, whilst there may be a reduction of the shock absorption ability in the VT axis after treatment.

By examining the variation of absolute and relative angles in Figure 7.2, it becomes evident that there is a steadier variation of angles or consistent angle patterns in post-therapy data. In other words, the angle parameters are more perceptible since they form a circle within a radius that is near to a zero-degree value in the mean radar plot.

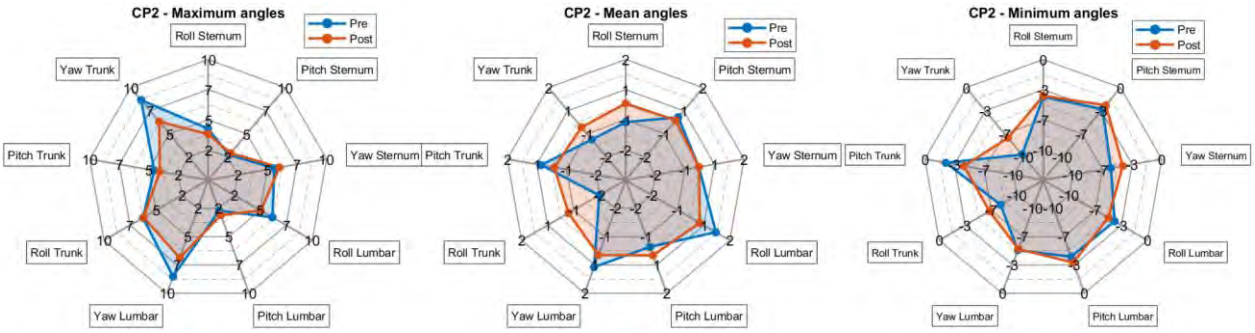


Figure 7.2 CP2 maximum, mean and minimum absolute and relative angles per axis.

Before the treatment, CP2 exhibited a propensity to laterally bend the trunk to the left, sagittal bend forward and rotate axially to the left. The sternum and lumbar spine also behave differently, while the sternum laterally bent to the left, the lumbar spine bent to the right and, thus, the relative roll angle between them was remarkable. A similar outcome was noticed with regard to sagittal bending and longitudinal torsion.

Nevertheless, it is concluded that following therapy CP2 demonstrates more symmetrical bending and torsion movements, by the distribution of absolute angles centered near the zero-value and the reduction of most of the maximum values.

CP3 exhibits a slight smoother sternum VT, lumbar AP and VT movements, as well as jerkier ML lumbar motion. Additionally, there is a higher degree of harmony in the lumbar ML, sternum AP and ML, and a decrease in harmonicity in the lumbar VT direction. Therefore, CP3 tends towards movements with lower impact in the AP and VT axes, with a certain degree of rhythmicity only in the VT axis.

By observing the mean angle values of CP3 radar plot in Figure 7.3, more consistent angle behavior in post-therapy dataset is observed. At plain sight, post-therapy angles are observed to be located near the zero-degree value. Nevertheless, CP3 demonstrates slight trunk compensation movements.

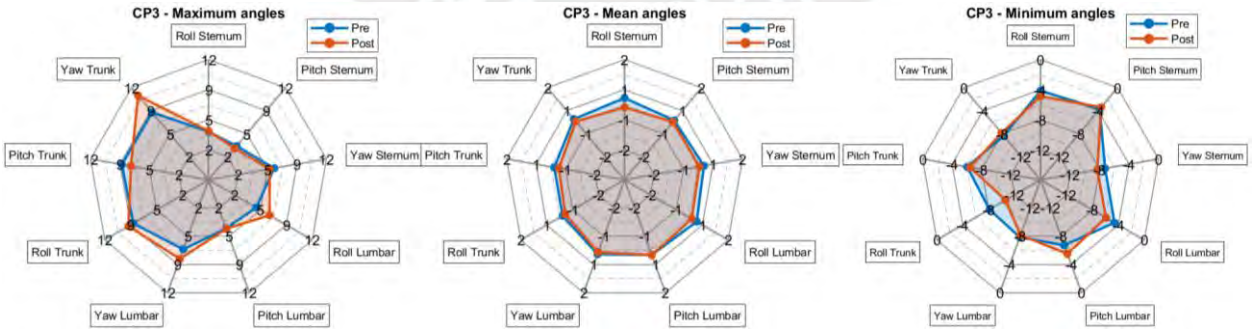


Figure 7.3 CP3 maximum, mean and minimum absolute and relative angles per axis.

CP3 exhibits, after treatment, a higher flexibility at bending laterally with a minor tendency towards the left, more symmetrical sagittal bending, and a greater asymmetry in longitudinal torsion with a tendency to the left.

The acceleration-derived parameters indicate that CP3 exhibits slightly enhanced stability and smoothness, though the observed changes are relatively minimal. The observations indicated a propensity to higher VT acceleration and a slight possibility of more harmonic lumbar spine movements, even though these two parameters often have a contrasting correlation.

Therefore, this observation may be attributed to a low shock absorption capacity with the repetition of a stable gait pattern. CP4 may experience a high-magnitude VT acceleration that is not completely attenuated by the lower limbs, despite the existence of stable and rhythmical gait cycles.

Regarding angular motion, the roll and yaw angles exhibit elevated median and data distribution values, as illustrated in Figure 7.4.

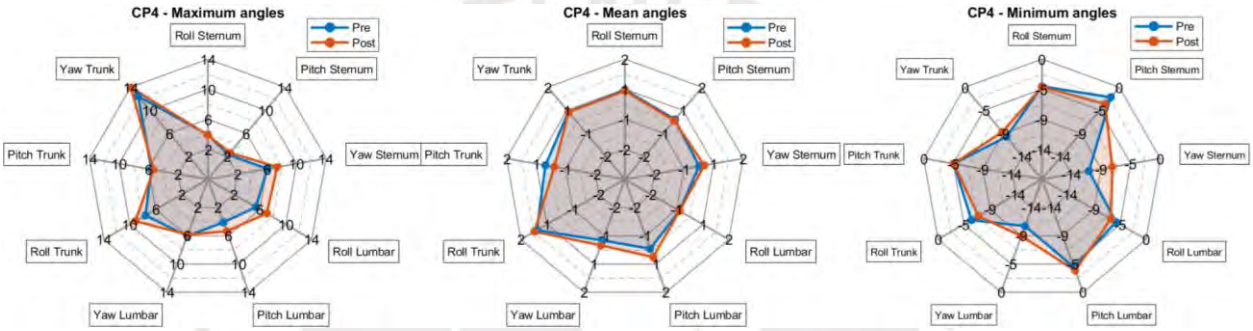


Figure 7.4 CP4 maximum, mean and minimum absolute and relative angles per axis.

CP4 exhibited a considerable tendency towards a lateral bending of the trunk to the right and a longitudinal torsion to the left. CP4 also demonstrated a minor proclivity towards trunk flexion. However, the low variation in this parameter after therapy does not allow a direct conclusion.

CP5 exhibits ML acceleration with lower magnitudes and less harmonicity especially in the sternum, as well as lower harmonic lumbar AP accelerations. However, there is a proclivity towards a greater ML acceleration attenuation from the lumbar spine to the sternum.

Figure 7.5 illustrates that the mean and minimum angles of CP5 before and after treatment exhibit a similar or close performance. However, most of the maximum values have changed. The relative angles of the trunk illustrate a lower tendency to lateral bending to the left, with a greater tendency to abnormal lateral bending to the right, slightly greater trunk flexion, and a lower proclivity to a longitudinal torsion to the left.

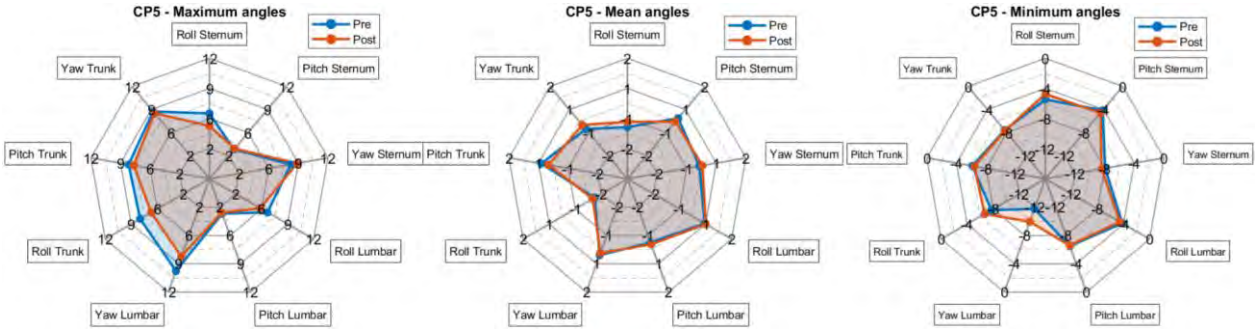


Figure 7.5 CP5 maximum, mean and minimum absolute and relative angles per axis.

CP6 presents greater smooth motion in both sternum and lumbar spine in the AP direction, in other words, this may suggest the existence of less abrupt and slightly better gait cycles. Although CP6 indicates a slight reduction in the gait cycles harmonicity in the ML direction, the trunk ML accelerations are significantly lower following therapy and the AC is higher.

This may indicate a greater absorption of the trunk acceleration in the ML axis, which may also exhibit movements without significant impact and may contain gait cycles with no discernible pattern, in that way resulting in a gait that is less harmonious.

Therefore, CP6 may exhibit similar or a minimal increase in harmonicity in the lumbar spine and similar or less attenuated gait cycles in the VT axis. This may be indicative of fewer jerky movements and more shock absorption capacity at least in the lumbar spine. Figure 7.6 displays a more symmetrical performance of the mean angles of CP6, as well as variation in the maximum and minimum radar plots.

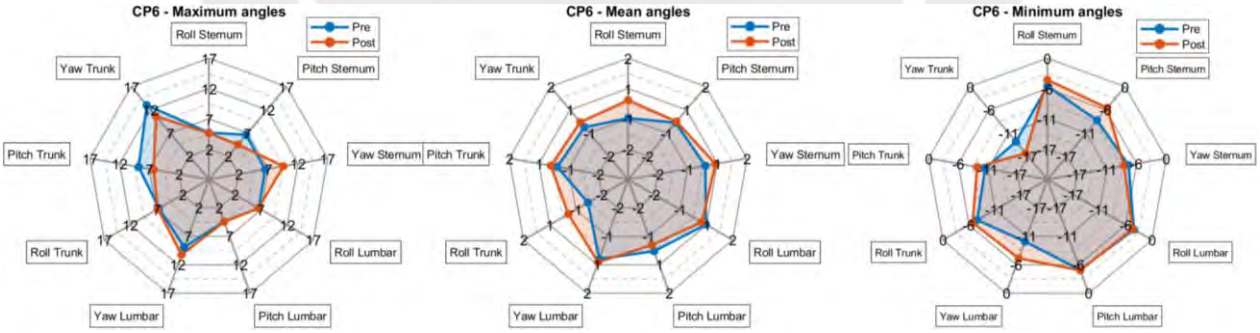


Figure 7.6 CP6 maximum, mean and minimum absolute and relative angles per axis.

After treatment, CP6 demonstrates more tendency to lateral bending to the left with abnormal bending motion to the right, a more symmetrical sagittal bending performance with erratic trunk extension movements, and a longitudinal torsion with a greater tendency towards torsion to the right.

7.7. Parameter variation

From the selected parameters only one varies for all the evaluated children with CP: the ML AC. In addition, five of the six participants exhibited a variation in the sternum roll

absolute angle. Four of the six children had a variation in the sternum ML RMS, lumbar AP RMS, VT AC, sternum and lumbar ML HR, and relative pitch angle.

A remarkable thing to notice is that the ML AC and sternum roll angle are both correlated with compensatory movements of the trunk in the lateral bending direction. Table 7.1 displays the mean values per dataset and its variation for each child with CP. It is notable that three of the participants decreased the ML AC, whilst the other three participants exhibited an increase in this parameter. Also, the sternum roll angle decreased for two children and increased for three children.

Table 7.1 ML AC and sternum roll mean values and variation per child with CP.

Child	ML AC (%)			Sternum Roll (°)		
	Pre	Post	Variation	Pre	Post	Variation
CP1	39.98	16.87	-23.11	0.98	-0.43	-1.41
CP2	47.89	38.14	-9.75	-0.77	0.05	0.82
CP3	41.4	45.45	4.05	0.3	-0.11	-0.41
CP4	51.79	40.24	-11.55	*0.64*	*0.59*	*-0.05*
CP5	-5.34	12.93	18.27	-1.31	-1.01	0.3
CP6	30.76	46.69	15.93	-0.64	0.16	0.8

Each of these participants was analyzed and discussed in Section 7.6, however, it is important to remark that there is an association between the ML AC and sternum roll angle with the sternum ML HR and the sternum ML RMS.

RMS and AC give information about the smoothness and HR provides information about the gait harmonicity based on the acceleration. Thus, it is important to analyze not only one parameter (ML AC) but also the associated parameters in that direction. The absolute roll angle from sternum provides angular motion information about that region in the ML direction, therefore, an examination of this parameter is also required to do a proper analysis of the performance of the children after treatment.

There is another relationship between parameters, but this time in the AP direction. The AP RMS, AP AC and the pitch angles provide information about the compensatory trunk motions in the context of the walking axis. It would be more feasible to find an association between these parameters in the sternum region, however, the results indicate that the variations were higher on the lumbar AP RMS and the relative pitch angle.

7.8. Comparison with other studies

Although this work focuses on identifying the variation in parameters between pre- and post-therapy dataset per child with CP, it would be beneficial to compare the obtained findings with the actual data in the literature for further analysis. Thus, the following figures

display the obtained RMS and AC for all the participants in this study to be able to make a comparison with other studies.

Firstly, a comparison between the findings of Summa et al. (2016) and this thesis is carried out. Figure 7.7 displays the results of the sternum RMS between both investigations. The distribution of the data per axis in pre- and post-therapy data is illustrated only in children with CP (Figure 7.7a), whilst Summa et al. (2016) presents this information for both children with CP and TD (Figure 7.7b).

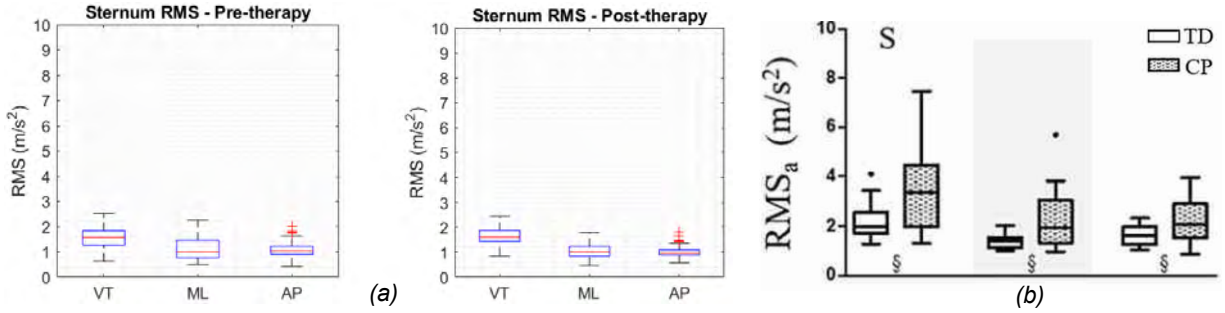


Figure 7.7 Comparison of sternum VT, ML and AP RMS between (a) this study and (b) Summa et al. (2016).

It is noticeable that Figure 7.7b exhibits greater RMS medians and data dispersion than this study per axis. Every RMS distribution from children with CP may be comparable or even smaller than the data from the TD-children, even though the RMS should be higher in children with CP. In addition, there is a notable reduction in the data dispersion in this study after treatment, which is remarkable in the ML direction.

Figure 7.8 displays the comparison between both studies regarding the lumbar RMS. This displays a similar behavior as the sternum RMS, with a pronounced increase of RMS in all the axis on Figure 7.8b.

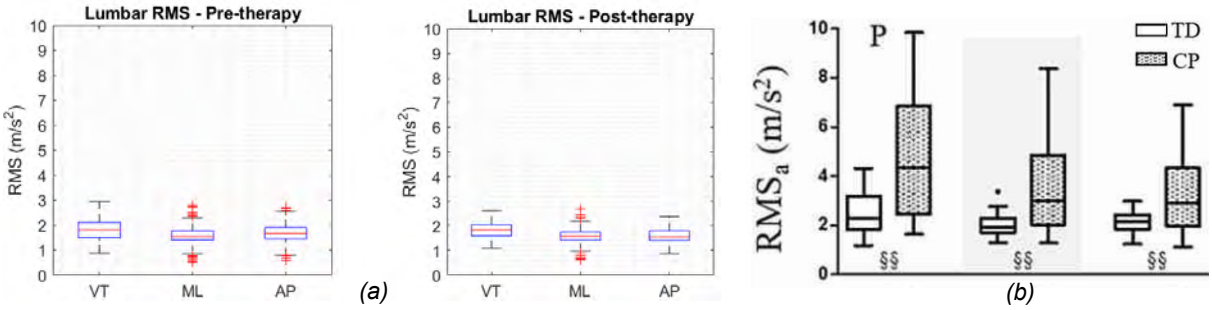


Figure 7.8 Comparison of lumbar VT, ML and AP RMS between (a) this study and (b) Summa et al. (2016).

The CP data from this study appear to have smaller RMS magnitudes compared to the TD data from Summa et al. (2016), even in the AP and ML axes. Between the pre- and post-therapy data of this thesis, the RMS distribution exhibits a slight decrease, which may indicate smoother signals after therapy.

On the other hand, Figure 7.9 displays a comparison between AC data. At first glance, the distributions appear to be centered near to the same values in each plot, however, Figure 7.9a possess greater mean AC values, data distribution and number of outliers compared to Figure 7.9b. Additionally, following Vojta therapy, there is a reduction of the AC data distribution, median value and number of outliers.

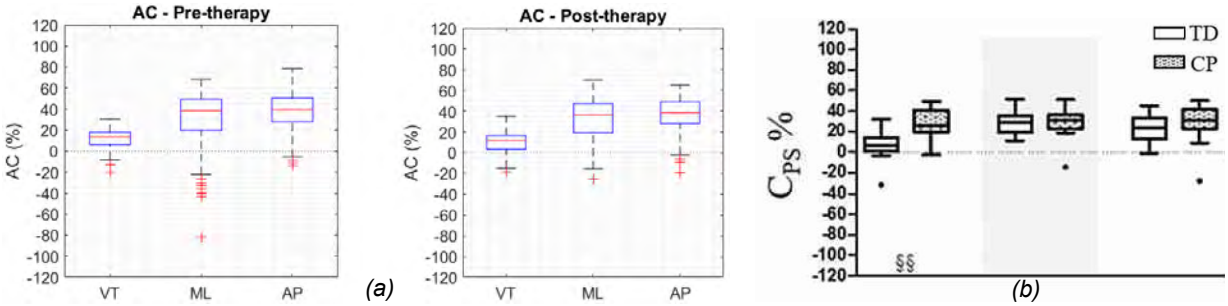


Figure 7.9 Comparison of VT, ML and AP AC between (a) this study and (b) Summa et al. (2016).

This comparison may be affected by the ages and gender of the participants evaluated, since this study evaluates six children with ages 9.83 ± 2.71, whilst Summa et al. (2016) evaluates 20 children with ages 5.85 ± 2.18. Thus, the higher RMS and lower AC values may be due to the reduced ages in the investigation evaluated.

This reasoning may be accompanied by the fact that lower RMS denotes smoother and more controlled movement, while a higher AC implies a better attenuation of the accelerations and may suggest strengthened muscles.

Secondly, a comparison between this study and Paraschiv-Ionescu et al. (2019) is carried out. Figure 7.10 illustrates the comparison between both studies regarding the total AC in several groups in children with CP with GMFCS level I and II and TD-children.

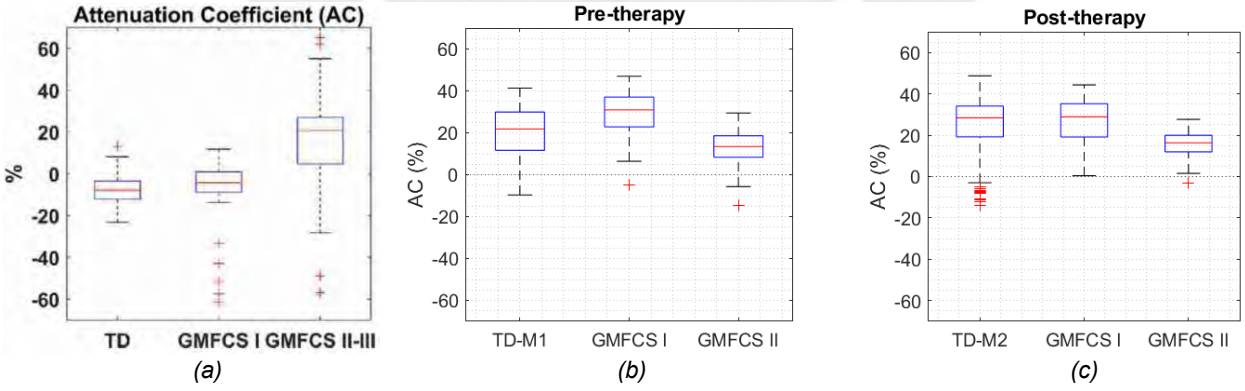


Figure 7.10 AC comparison between (a) Paraschiv-Ionescu et al. (2019) and (b-c) this study.

Comparing the TD-children only, it is unclear why the children from this investigation exhibit in both measurements (Figure 7.10b-c) a greater total AC than Figure 7.10a. There is also a contradiction between the children with CP GMFCS level, because this study shows that GMFCS-I exhibits greater total AC values in both pre- and post-therapy

datasets, compared to Paraschiv-Ionescu et al. (2019), and the GMFCS-II group of this study demonstrated the opposite behavior.

Nevertheless, it may be inaccurate to compare both GMFCS levels in this study since the GMFCS-II group has only one participant. Additionally, Paraschiv-Ionescu et al. (2019) evaluated children and adolescents with CP with ages 12.8 ± 3.1.

7.9. Performance of TD-children

Since the main objective of this thesis is to be able to quantify the improvement on the Vojta therapy in children with CP, the analysis of TD-children is not required. However, each matched TD-child should provide insight into some normality values for the related child with CP.

Due to the age of the evaluated TD-children, it is normal to expect slight differences between the two measurements (M1 and M2). Figure 7.11 displays the performance of TD4 in both trials in a radar plot. This figure shows that there are no significant differences between those tests, only in some parameters like AC, HR and the lateral bending.

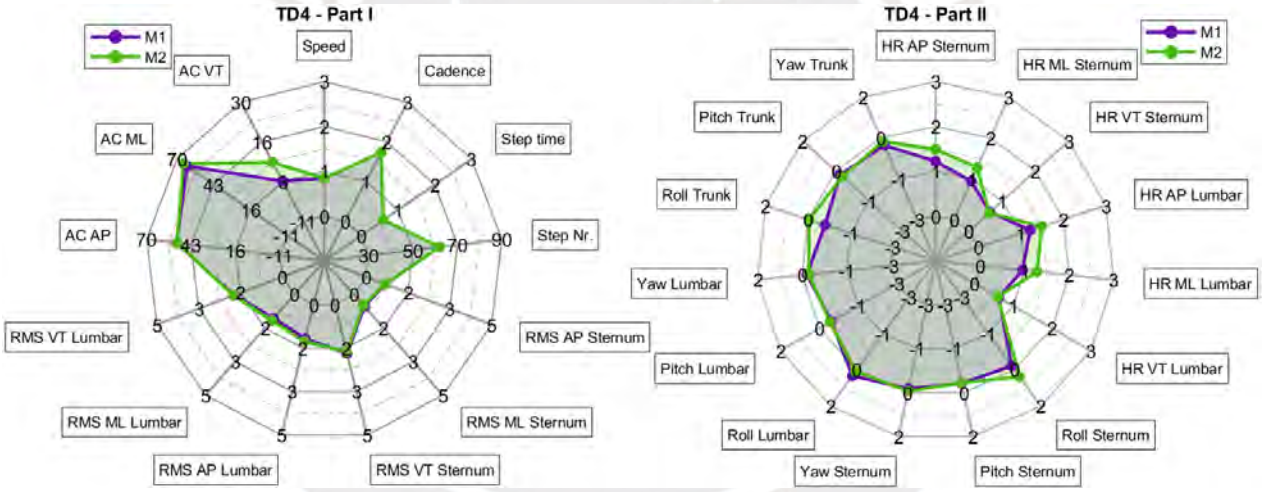


Figure 7.11 Radar plots for the selected parameters for TD4.

The variations of these parameters can be easily explained. The increase of the HR may give insight into a more harmonic gait performance, as well as a higher attenuation in the VT direction. Circumstantial factors could have also had an effect, TD4 may have been unsure or nervous while walking on the first test and, therefore, could have lost some rhythmicity in the gait. The AC mean values of TD4 remain between the range obtained from other studies.

On the other hand, Figure 7.12 exhibits the parameters for TD2 in both measurements (M1 and M2). TD2 and TD4 exhibit different performances even though both are the same age. It concerns the different behavior between the AP and ML AC between M1 and M2. Furthermore, the evaluated parameters do not remain constant in TD2 as they do in TD4.

TD4 also demonstrates better symmetry in most of the angles, whilst TD2 tends to an extension of the lumbar and a left lateral bending of the trunk.

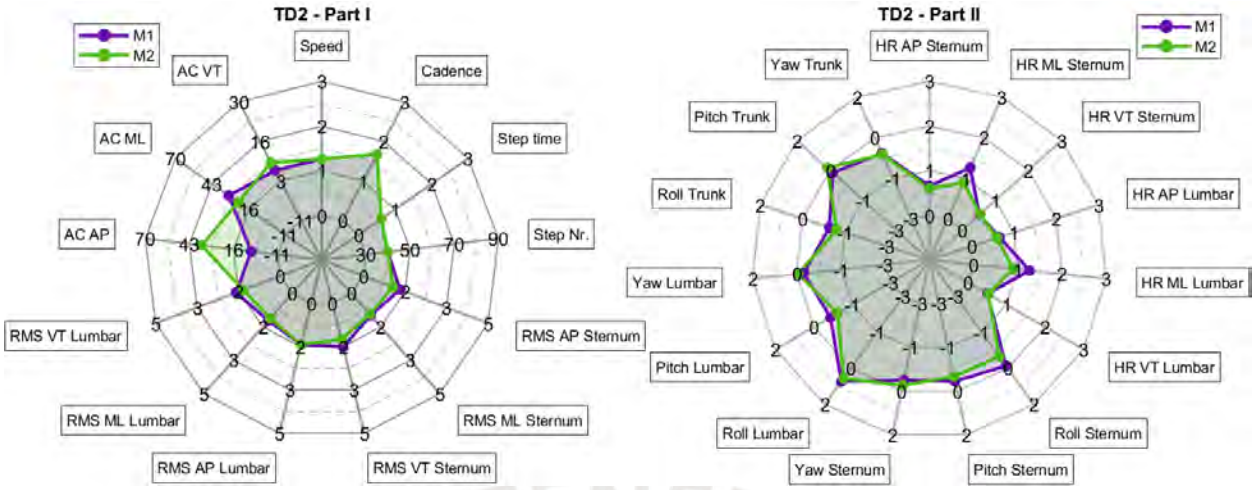


Figure 7.12 Radar plots for the selected parameters for TD2.

Another example of an out of abnormal behavior in TD-children is displayed in Figure 7.13. TD6 exhibits superior RMS values especially in the VT direction in comparison to all the children with CP. The expected performance to get from the TD-matching children was lower RMS values in comparison to children with CP. However, TD6 displays the higher RMS mean values among all the populations.

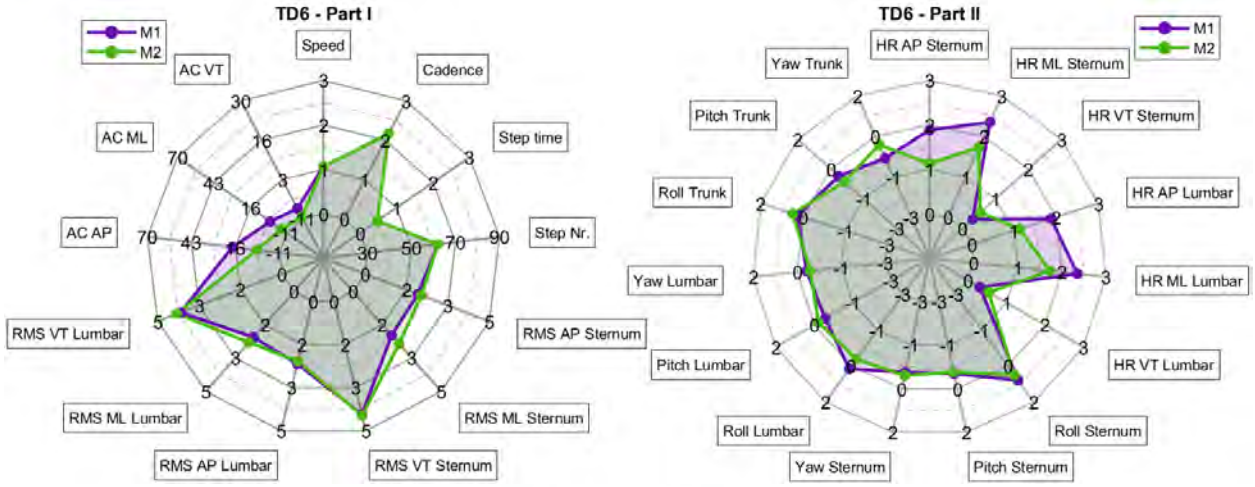


Figure 7.13 Radar plots for the selected parameters for TD6.

Comparing Figure 7.13 and Figure 6.6, there is a great difference between RMS, AC and some HR values. It is expected that TD-children have lower RMS values as well as high AP and ML AC values. However, TD6 exhibited more harmonic gait performance in some of the axis.

7.10. Error discussion

This section will provide information about some of the sources of error discovered in this work.

7.10.1. Magnetometer

The magnetometer presents a substantial source of error due to the changing magnetic field in each test. This was perceived while observing the stationary part of each measurement per child. Some of the starting magnetometer values were similar for each axis, while others were out of the observed range.

There are many reasons for this behavior. Firstly, the test setup varied four times per child dataset, given that the time between each CP- and TD-child assessment was 12 days of interval. Additionally, the initial measurement of the first test of both CP- and TD-children was not performed on the same date.

Secondly, the tests included the presence of the parents of the children because it was required for children of the smallest ages. However, if the tests were conducted in the presence of more people, the observed effect may have been pronounced. The magnetic field may be affected by this when electronic devices such as smartphones and smartwatches are used.

On the other hand, there might have been some shifting of the real magnetic field related to the presence of electronic devices in the test room. This anomaly is also known as a hard iron magnetism, which do not distort the magnetic field signal but instead generates an offset. To overcome this, hard and soft iron calibration is essential, however, it is unclear whether this procedure was conducted or if the OPAL IMU system incorporated it into its calibration protocol.

Due to the different initial magnetic field values per axis in each measurement, a procedure to normalize the magnetometer data for all participants was carried out. It consisted only of a subtraction of the mean magnetic field value of the stationary phase of each test. This approach eliminates any offset at the beginning of the trials, focusing only on the variations of the magnetic field across each movement.

7.10.2. Sensor fusion

One of the biggest sources of error may be the use of the complementary filter for the calculation of the Euler angles. Table 2.2 describes the RMSE of the complementary filter and the EKF, in which the complementary filter with a $\alpha = 0.7$ had the greatest error value in the heading (12.98°).

However, this work used a complementary filter with a tuning parameter $\gamma = 0.95$, which is closer to the complementary filter estimation with $\alpha = 0.07$ from Table 2.2. Therefore, the approximate error values in the axis would be 1.44°, 1.43° and 4.39° for the roll, pitch and heading or yaw, respectively. These values are referential; however, they may be

evidence of a great source of error in the estimation of the absolute and relative angles of the trunk. A variation of 1° in roll and pitch angles as well as the variation of 4° in yaw angles may significantly impact the plotted data.

As a solution, the EKF is an optimal sensor fusion algorithm in comparison to the complementary filter and, therefore, it has lower RMSE values. This algorithm was excluded from this work due to its higher level of complexity and, thus, increased time to test the usage of this filter compared to complementary filter.

Furthermore, the calculation of the yaw angle might have been affected by the magnetic field errors explained in Section Magnetometer7.10.1, since its calculation is based on the magnetic field equations Eq. 5.12 to 5.15.

The complementary filter combines the gyroscope and magnetometer to estimate the yaw angle, giving a greater weight to the gyroscope ($\gamma = 0.95$). Although the greater preference to the angular velocity data, given the explained errors from both the magnetometer and the 4° yaw angle variation in complementary filter, there might be a tendency towards a greater error in the yaw angle estimation in this study.

7.10.3. Children performance

The performance of the children could have been influenced by the test environment. As is known, the tests have the presence of the parents to aid the younger children, and this may influence the gait performance of the children. To illustrate this, if the parent was in a fixed position on the walkway to provide confidence to the child, there is a possibility of different gait performance on the walks.

The child may have been excited to see his parents in three out of the six walks performed where the child could have seen the parent and walked faster to get closer. Therefore, in three of the six walks there is a probability of higher variability of parameters, such as gait speed, RMS acceleration and harmonic ratio.

Another possibility is that the child might have walked laterally towards the parent and deviated from the straight pathway of the test, occasioning an increase of steps and a variation of the parameters derived of the acceleration.

On the other hand, it is unclear whether the test room had empty walls or was decorated with paintings or adornment that might have captured the attention of the children and, therefore, distracted them from the initial focus: conducting a normal gait.

7.10.4. Miscalculation of parameters

Another source of error can be obtained through the calculation of spatiotemporal parameters. Since the main focus of this work is to use only the trunk data, based on

sternum and lumbar spine data, the spatiotemporal parameters were not calculated using feet but trunk information.

The spatiotemporal parameters can be easily derived from lumbar VT accelerations; however, it is necessary to measure the altitude of the sensor with respect to the ground. Then, with this information, the parameters in question can be calculated using the inverted pendulum model.

These parameters were calculated using the formulas from Section 5.4.1, in which is used the distance covered as five meters. This assumption may lead to inaccurate and erroneous estimation of the parameters.

This is due to the possibility that the children may begin each walk prior or after to the start of the walkway, as indicated by the presence of cones (see Figure 4.1). Additionally, the children could begin each walk while performing a turn.

In both instances, the distance covered would differ from five meters and lead to miscalculations. Table 7.2 describes the error using the five meters as the covered distance, considering an additional error distance of ± 0.5 meters. As expected, this distance error of 10% is proportional to the obtained speed and step length errors.

Table 7.2 Error calculation in gait speed and step length estimation for CP1 pre-therapy.

Walk Nr.	Walk time (s)	Step Nr.	Speed (m/s)	Speed error (m/s)	Step length (m)	Error step length (m)
W1	3.993	8	1.252	0.125	0.625	0.063
W2	2.313	6	2.162	0.216	0.833	0.083
W3	3.422	8	1.461	0.146	0.625	0.063
W4	3.422	8	1.461	0.146	0.625	0.063
W5	3.946	8	1.267	0.127	0.625	0.063
W6	2.969	6	1.684	0.168	0.833	0.083
Average	3.344	7.3	1.548	0.155	0.694	0.07

Thus, the real distance covered by the children affects proportionally to the calculation of the speed, step and stride lengths. The estimated parameters are not used for further calculations or analysis in this thesis; however, it is important to acknowledge this discrepancy.

On the other hand, the spatiotemporal parameters that could have been taken into consideration in parameter evaluation are the cadence and stride time. This is due to its estimation, because it only requires the time in which the walk started and finalized, as well as the number of steps taken. Since they are calculated after the gait cycles detection, they do not require the assumption of distance.

8. Conclusions

This master thesis has successfully developed an algorithm that identifies variations in trunk control parameters in a group of six children with CP after performing blocks of Vojta therapy. In conclusion, the quantification of the children's gait performance led to the identification of one parameter that changes for every child with CP – the ML AC. In addition, five children presented a variation of the sternum roll angle.

Both parameters are correlated as a compensatory trunk movement in lateral bending. Nevertheless, the analysis of one or two parameters does not provide the full picture of the performance of the children. It is required to analyze all the associated parameters per direction.

Therefore, four of the six children exhibited a variation in the sternum ML RMS, sternum and lumbar ML HR, lumbar AP RMS, VT AC, and the relative pitch angle between the lumbar and the sternum. These first two parameters are also associated with the ML AC and sternum roll.

The RMS, AC and HR give information about gait smoothness and harmonicity in the ML direction, whilst the sternum roll provides information about the lateral bending motion. It is understandable and remarkable that ML RMS and HR are associated with the parameters that vary the most in this group of children. This may be related to the compensatory trunk movements during walking in the ML direction and the lateral bending of the trunk.

In spastic CP, there is a visible reduced movement during walking. Therefore, Vojta therapy focuses on improving postural control, which results consequently in an increase in the trunk variability and an approach of the trunk to the midline. Among the children with CP analyzed, CP1 and CP2 demonstrated an improved performance in postural control after therapy. CP1 exhibited greater trunk variability and CP2 exhibited a closer approach to the midline. Thus, the effect of the treatment may have been proved in both children with CP in this work.

Most of the improvement noticed in the findings in all the children with CP correspond with the physical therapist perception of the evolution of children. This validates the measurements and the proposed algorithms, setting the path for further analysis.

9. Outlook

In the first place, a very important improvement would be to use EKF instead of the complementary filter due to orientation estimation errors. It would be interesting to quantify the errors between both methods in the used dataset. EKF might provide more accurate measurements of the yaw angle. Since the calculation of this angle in this thesis was subjected to the magnetic field and angular velocity data, it could provide errors.

Subsequently, a great area to improve after this work would be the better detection of the gait events of the children with CP. One potential approach would be to use a Finite State Machine (FSM) to estimate the gait events.

Another approach would be to train a deep learning algorithm to perform automatic gait event detection and therefore, a more automatic manner to obtain the gait cycles. However, for this last approach, a large amount of data would be required to perform appropriate model training and, thus, significant performance in the gait cycles detection for children with CP.

Furthermore, any of the gait event detection methods taken would allow a detection of gait phases and subphases. Therefore, it would be more feasible to not only estimate more spatiotemporal parameters but also parameters such as trunk asymmetry, principal component analysis (PCA), postural sway, gait regularity and so forth. The greater the number of parameters obtained, the greater analysis that should be executed to find associations between parameters that allow to quantify the performance of each child with CP after treatment.

In addition, it would be very interesting to have the implementation of the Tilt-twist method described in Section 3.2.3 to evaluate the spine rotations effectively and make comparisons with the method used in this thesis. To achieve this, the integration of at least one sensor in mid thoracic vertebrae should provide more insight into the trunk movements.

Moreover, the implementation of the AGG and CGG from Section 3.2.4 in all the angular motion estimated parameters would be very beneficial. The angular motion in every direction displays a discernible pattern of behavior, which was not further analyzed and compared between children of the same age. However, both proposed graphs would help to provide insight into the angular patterns in children with CP and compare between children with the same age.

As a final point, to provide a more precise quantification of the findings, it would be feasible to develop a gait score between all the parameters evaluated, considering the expected performance in children with spastic CP – an improved postural control.

References

Armand, S., Decoulon, G. and Bonnefoy-Mazure, A. (2016) 'Gait analysis in children with cerebral palsy', *EFORT open reviews*, vol. 1, no. 12, pp. 448–460.

Baghdadi, A., Cavuoto, L. A. and Crassidis, J. L. (2018) 'Hip and Trunk Kinematics Estimation in Gait Through Kalman Filter Using IMU Data at the Ankle', *IEEE Sensors Journal*, vol. 18, no. 10, pp. 4253–4260.

Carcreff, L., Gerber, C. N., Paraschiv-Ionescu, A., Coulon, G. de, Newman, C. J., Aminian, K. and Armand, S. (2020) 'Comparison of gait characteristics between clinical and daily life settings in children with cerebral palsy', *Scientific reports*, vol. 10, no. 1, p. 2091.

Castiglia, S. F., Tatarelli, A., Trabassi, D., Icco, R. de, Grillo, V., Ranavolo, A., Varrecchia, T., Magnifica, F., Di Lenola, D., Coppola, G., Ferrari, D., Denaro, A., Tassorelli, C. and Serrao, M. (2021) 'Ability of a Set of Trunk Inertial Indexes of Gait to Identify Gait Instability and Recurrent Fallers in Parkinson's Disease', *Sensors (Basel, Switzerland)*, vol. 21, no. 10.

CDC (2024) *About Cerebral Palsy* [Online], US Center for Disease Control and Prevention. Available at https://www.cdc.gov/cerebral-palsy/about/index.html (Accessed 29 July 2024).

CDC Archive (2022) *Data and Statistics for Cerebral Palsy* [Online], Ceneters for Disease Control and Prevention. Available at https://archive.cdc.gov/#/details?url=https://www.cdc.gov/ncbddd/cp/data.html (Accessed 17 July 2024).

Chen, X., Liao, S., Cao, S., de Wu and Zhang, X. (2017) 'An Acceleration-Based Gait Assessment Method for Children with Cerebral Palsy', *Sensors*, vol. 17, no. 5.

Christine, C., Dolk, H., Platt, M. J., Colver, A., Prasauskiene, A. and Krägeloh-Mann, I. (2007) 'Recommendations from the SCPE collaborative group for defining and classifying cerebral palsy', *Developmental medicine and child neurology. Supplement*, vol. 109, pp. 35–38 [Online]. DOI: 10.1111/j.1469-8749.2007.tb12626.x.

Clario (2024) *Opal* [Online], APDM Wearable Technologies, A Clario Company. Available at https://clario.com/wp-content/uploads/2024/05/24_01_PxM_OpalTechSpecsV2R_v2.pdf (Accessed 26 September 2024).

Contini, B. G., Bergamini, E., Alvini, M., Di Stanislao, E., Di Rosa, G., Castelli, E., Vannozzi, G. and Camomilla, V. (2019) 'A wearable gait analysis protocol to support the choice of the appropriate ankle-foot orthosis: A comparative assessment in children with Cerebral Palsy', *Clinical biomechanics (Bristol, Avon)*, vol. 70, pp. 177–185.

Digo, E., Pierro, G., Pastorelli, S. and Gastaldi, L. (2019) 'Tilt-Twist Method Using Inertial Sensors to Assess Spinal Posture During Gait', *International Conference on Robotics in Alpe-Adria Danube Region*, pp. 384–392 [Online]. Available at https://www.researchgate.net/publication/332948660.

Digo, E., Pierro, G., Pastorelli, S. and Gastaldi, L. (2020) 'Evaluation of spinal posture during gait with inertial measurement units', *Proceedings of the Institution of Mechanical Engineers. Part H, Journal of engineering in medicine*, vol. 234, no. 10, pp. 1094–1105.

Fraden, J. (2016) *Handbook of Modern Sensors*, Cham, Springer International Publishing.

Groves, P. D. (2015) 'Navigation using inertial sensors', *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 2, pp. 42–69.

Internationale Vojta Gesellschaft e.V. (2024) *Vojta Therapy* [Online], Internationale Vojta Gesellschaft e.V. Available at https://www.vojta.com/en/the-vojta-principle/vojta-therapy (Accessed 23 July 2024).

Iosa, M., Marro, T., Paolucci, S. and Morelli, D. (2012) 'Stability and harmony of gait in children with cerebral palsy', *Research in developmental disabilities*, vol. 33, no. 1, pp. 129–135.

Iosa, M., Morelli, D., Marro, T., Paolucci, S. and Fusco, A. (2013) 'Ability and stability of running and walking in children with cerebral palsy', *Neuropediatrics*, vol. 44, no. 3, pp. 147–154.

Jain, S. K. and Jain, G. (2016) *Spine and spinal orthoses* [Online], New Delhi, Jaypee, The Health Sciences Publisher. Available at https://www.jaypeedigital.com/book/9789351526407.

Kok, M., Hol, J. D. and Schön, T. B. (2017) 'Using Inertial Sensors for Position and Orientation Estimation', *Foundations and Trends® in Signal Processing*, vol. 11, 1-2, pp. 1–153.

Kuo, A. D. and Donelan, M. (2010) 'Dynamic Principles of Gait and their Clinical Implications', *Physical Therapy*, vol. 90, no. 2, pp. 157–174.

Madgwick, S. O. H., Harrison, A. J. L. and Vaidyanathan, A. (2011) 'Estimation of IMU and MARG orientation using a gradient descent algorithm', *IEEE … International Conference on Rehabilitation Robotics : [proceedings]*, vol. 2011, pp. 1–7.

MathWorks (2024) *Root mean square value* [Online], MathWorks. Available at https://de.mathworks.com/help/matlab/ref/rms.html#d126e1563297 (Accessed 18 September 2024).

McCamley, J., Donati, M., Grimpampi, E. and Mazzà, C. (2012) 'An enhanced estimate of initial contact and final contact instants of time using lower trunk inertial sensor data', *Gait & posture*, vol. 36, no. 2, pp. 316–318.

Moses, Y. (2024) *spider_plot* [Online], GitHub. Available at https://github.com/NewGuy012/spider_plot (Accessed 16 October 2024).

Mutoh, T., Mutoh, T., Takada, M., Doumura, M., Ihara, M., Taki, Y., Tsubone, H. and Ihara, M. (2016) 'Application of a tri-axial accelerometry-based portable motion recorder for the quantitative assessment of hippotherapy in children and adolescents with cerebral palsy', *Journal of physical therapy science*, vol. 28, no. 10, pp. 2970–2974.

Mutoh, T., Mutoh, T., Tsubone, H., Takada, M., Doumura, M., Ihara, M., Shimomura, H., Taki, Y. and Ihara, M. (2018) 'Impact of serial gait analyses on long-term outcome of hippotherapy in children and adolescents with cerebral palsy', *Complementary Therapies in Clinical Practice*, vol. 30, pp. 19–23 [Online]. DOI: 10.1016/j.ctcp.2017.11.003.

O'Brien, M. K., Hidalgo-Araya, M. D., Mummidisetty, C. K., Vallery, H., Ghaffari, R., Rogers, J. A., Lieber, R. and Jayaraman, A. (2019) 'Augmenting Clinical Outcome Measures of Gait and Balance with a Single Inertial Sensor in Age-Ranged Healthy Adults', *Sensors (Basel, Switzerland)*, vol. 19, no. 20.

Oppenheim, A. V. (2011) *Butterworth Filters* [Online], MIT OpenCourseWare. Available at https://ocw.mit.edu/courses/res-6-007-signals-and-systems-spring-2011/6ffe3f6c387555a8db26f1f3bbaddfb5_MITRES_6_007S11_lec24.pdf (Accessed 12 September 2024).

Paraschiv-Ionescu, A., Newman, C. J., Carcreff, L., Gerber, C. N., Armand, S. and Aminian, K. (2019) 'Locomotion and cadence detection using a single trunk-fixed accelerometer: validity for children with cerebral palsy in daily life-like conditions', *Journal of neuroengineering and rehabilitation*, vol. 16, no. 1, p. 24.

Physiopedia (2024) *Physiotherapy Treatment Approaches for Individuals with Cerebral Palsy* [Online]. Available at https://www.physio-pedia.com/Physiotherapy_Treatment_Approaches_for_Individuals_with_Cerebral_Palsy (Accessed 23 July 2024).

Piitulainen, H., Kulmala, J.-P., Mäenpää, H. and Rantalainen, T. (2021) 'The gait is less stable in children with cerebral palsy in normal and dual-task gait compared to typically

developed peers', *Journal of biomechanics*, vol. 117, p. 110244 [Online]. DOI: 10.1016/j.jbiomech.2021.110244.

Proakis, J. G. and Manolakis, D. G. (1996) *Digital signal processing: Principles, algorithms, and applications*, 3rd edn, Upper Saddle River N.J., Prentice Hall.

Sadowska, M., Sarecka-Hujar, B. and Kopyta, I. (2020) 'Cerebral Palsy: Current Opinions on Definition, Epidemiology, Risk Factors, Classification and Treatment Options', *Neuropsychiatric disease and treatment*, vol. 16, pp. 1505–1518.

Sæther, R., Helbostad, J. L., Adde, L., Brændvik, S., Lydersen, S. and Vik, T. (2014) 'Gait characteristics in children and adolescents with cerebral palsy assessed with a trunk-worn accelerometer', *Research in developmental disabilities*, vol. 35, no. 7, pp. 1773–1781.

Sæther, R., Helbostad, J. L., Adde, L., Braendvik, S., Lydersen, S. and Vik, T. (2015) 'The relationship between trunk control in sitting and during gait in children and adolescents with cerebral palsy', *Developmental medicine and child neurology*, vol. 57, no. 4, pp. 344–350.

Schaumann, A. (2024) *Spastik* [Online], Charité Universitätsmedizin Berlin. Available at https://kinderneurochirurgie.charite.de/fuer_patienten/haeufigste_krankheitsbilder/spastik/ (Accessed 17 July 2024).

Sekine, M., Tamura, T., Yoshida, M., Suda, Y., Kimura, Y., Miyoshi, H., Kijima, Y., Higashi, Y. and Fujimoto, T. (2013) 'A gait abnormality measure based on root mean square of trunk acceleration', *Journal of neuroengineering and rehabilitation*, vol. 10.

Shoemake, K. *Quaternions,* University of Pennsylvania [Online]. Available at https://campar.in.tum.de/twiki/pub/Chair/DwarfTutorial/quatut.pdf (Accessed 14 May 2024).

Sivarajah, L., Kane, K. J., Lanovaz, J., Bisaro, D., Oates, A., Ye, M. and Musselman, K. E. (2018) 'The Feasibility and Validity of Body-Worn Sensors to Supplement Timed Walking Tests for Children with Neurological Conditions', *Physical & occupational therapy in pediatrics*, vol. 38, no. 3, pp. 280–290.

Smith, S. W. (1999) *The scientist and engineer's guide to digital signal processing*, 2nd edn, San Diego, Calif., California Technical Pub.

Summa, A., Vannozzi, G., Bergamini, E., Iosa, M., Morelli, D. and Cappozzo, A. (2016) 'Multilevel Upper Body Movement Control during Gait in Children with Cerebral Palsy', *PloS one*, vol. 11, no. 3, e0151792.

Tramontano, M., Morone, G., Curcio, A., Temperoni, G., Medici, A., Morelli, D., Caltagirone, C., Paolucci, S. and Iosa, M. (2017) 'Maintaining gait stability during dual walking task: effects of age and neurological disorders', *European journal of physical and rehabilitation medicine*, vol. 53, no. 1, pp. 7–13.

Vaughan, C. L., Davis, B. L. and O'Connor, J. C. (1999) *Dynamics of Human Gait*, 2nd edn, Cape Town, South Africa, Kiboho Publishers.

VectorNav (2024a) *Magnetometer Hard & Soft Iron Calibration* [Online]. Available at https://www.vectornav.com/resources/inertial-navigation-primer/specifications--and--error-budgets/specs-hsicalibration.

VectorNav (2024b) *MEMS operation* [Online], VectorNav. Available at https://www.vectornav.com/resources/inertial-navigation-primer/theory-of-operation/theory-mems (Accessed 10 September 2024).

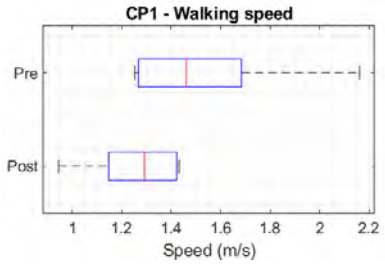
Winter, D. A. (2009) *Biomechanics and motor control of human movement*, 4th edn, Hoboken, N.J., John Wiley & Sons, Inc.

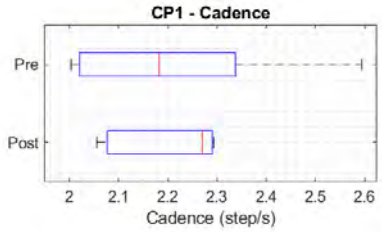
Woodman, O. J. (2007) *An introduction to inertial navigation* [Online], Cambridge, University of Cambridge (696).

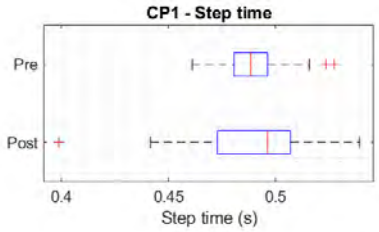
Appendix

A. Spatiotemporal Parameters

A.1 Spatiotemporal Parameters CP1

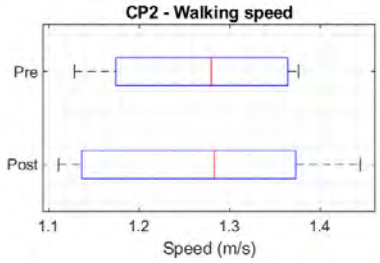


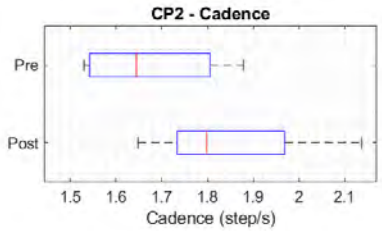


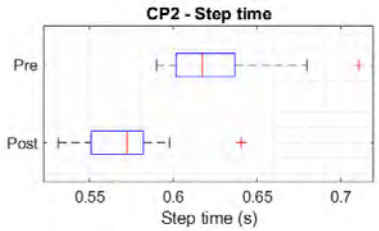


Dataset	Walk Nr.	Speed (m/s)	Cadence (s^{-1})		Length (m)		Time (s)		Number	
			Step	Stride	Step	Stride	Step	Stride	Step	Stride
CP1-Pre	W1	1.252	2.004	1.002	0.625	1.250	0.499	0.998	8	4
	W2	2.162	2.594	1.297	0.833	1.667	0.474	0.948	6	3
	W3	1.461	2.337	1.169	0.625	1.250	0.483	0.965	8	4
	W4	1.461	2.337	1.169	0.625	1.250	0.495	0.990	8	4
	W5	1.267	2.027	1.014	0.625	1.250	0.493	0.987	8	4
	W6	1.684	2.021	1.010	0.833	1.667	0.495	0.990	6	3
	Average	**1.548**	**2.220**	**1.110**	**0.694**	**1.389**	**0.490**	**0.981**	**7.3**	**3.7**
CP1-Post	W1	1.147	2.294	1.147	0.500	1.000	0.483	0.966	10	5
	W2	0.944	2.265	1.133	0.417	0.833	0.478	0.956	12	6
	W3	1.285	2.056	1.028	0.625	1.250	0.486	0.973	8	4
	W4	1.298	2.077	1.038	0.625	1.250	0.482	0.963	8	4
	W5	1.432	2.290	1.145	0.625	1.250	0.499	0.998	8	4
	W6	1.422	2.275	1.138	0.625	1.250	0.511	1.022	8	4
	Average	**1.254**	**2.209**	**1.105**	**0.569**	**1.139**	**0.489**	**0.977**	**9**	**4.5**
TD1-M1	W1	1.379	2.206	1.103	0.625	1.250	0.512	1.024	8	4
	W2	1.666	2.000	1.000	0.833	1.667	0.500	1.000	6	3
	W3	1.620	1.944	0.972	0.833	1.667	0.514	1.029	6	3
	W4	1.981	2.377	1.189	0.833	1.667	0.522	1.044	6	3
	W5	1.347	2.155	1.078	0.625	1.250	0.515	1.029	8	4
	W6	1.350	2.160	1.080	0.625	1.250	0.535	1.071	8	4
	Average	**1.557**	**2.140**	**1.070**	**0.729**	**1.458**	**0.517**	**1.034**	**7**	**3.5**
TD1-M2	W1	1.272	2.035	1.018	0.625	1.250	0.491	0.983	8	4
	W2	1.729	2.075	1.038	0.833	1.667	0.482	0.964	6	3
	W3	1.758	2.109	1.055	0.833	1.667	0.474	0.948	6	3
	W4	1.287	2.060	1.030	0.625	1.250	0.485	0.971	8	4
	W5	1.506	2.409	1.204	0.625	1.250	0.484	0.969	8	4
	W6	1.425	2.280	1.140	0.625	1.250	0.490	0.981	8	4
	Average	**1.496**	**2.162**	**1.081**	**0.694**	**1.389**	**0.485**	**0.970**	**7.3**	**3.7**

A.2 Spatiotemporal Parameters CP2

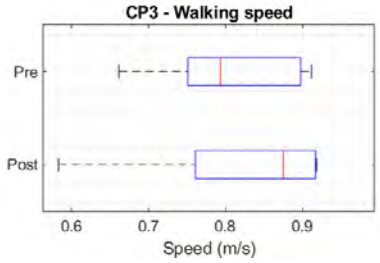


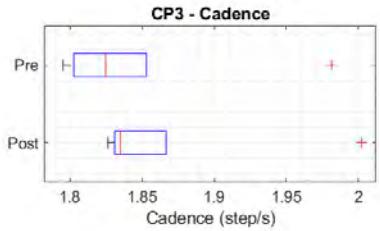


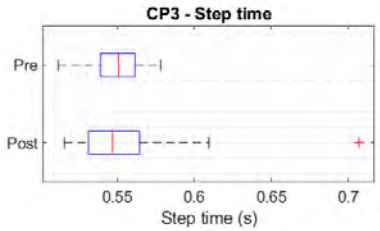


Dataset	Walk Nr.	Speed (m/s)	Cadence (s^{-1})		Length (m)		Time (s)		Number	
			Step	Stride	Step	Stride	Step	Stride	Step	Stride
CP2-Pre	W1	1.275	1.530	0.765	0.833	1.667	0.654	1.307	6	3
	W2	1.285	1.542	0.771	0.833	1.667	0.649	1.297	6	3
	W3	1.376	1.651	0.826	0.833	1.667	0.606	1.211	6	3
	W4	1.129	1.806	0.903	0.625	1.250	0.621	1.242	8	4
	W5	1.364	1.637	0.819	0.833	1.667	0.611	1.222	6	3
	W6	1.174	1.879	0.939	0.625	1.250	0.615	1.231	8	4
	Average	**1.267**	**1.674**	**0.837**	**0.764**	**1.528**	**0.625**	**1.250**	**6.7**	**3.3**
CP2-Post	W1	1.336	2.137	1.069	0.625	1.250	0.557	1.113	8	4
	W2	1.137	1.819	0.909	0.625	1.250	0.550	1.100	8	4
	W3	1.111	1.777	0.889	0.625	1.250	0.563	1.125	8	4
	W4	1.444	1.733	0.867	0.833	1.667	0.577	1.154	6	3
	W5	1.373	1.648	0.824	0.833	1.667	0.607	1.214	6	3
	W6	1.231	1.969	0.984	0.625	1.250	0.576	1.153	8	4
	Average	**1.272**	**1.847**	**0.924**	**0.694**	**1.389**	**0.570**	**1.139**	**7.3**	**3.7**
TD2-M1	W1	1.103	1.765	0.883	0.625	1.250	0.666	1.332	8	4
	W2	1.492	1.790	0.895	0.833	1.667	0.559	1.117	6	3
	W3	1.277	2.044	1.022	0.625	1.250	0.541	1.082	8	4
	W4	1.339	2.142	1.071	0.625	1.250	0.521	1.041	8	4
	W5	1.596	1.915	0.957	0.833	1.667	0.522	1.044	6	3
	W6	1.488	1.786	0.893	0.833	1.667	0.560	1.120	6	3
	Average	**1.382**	**1.907**	**0.953**	**0.729**	**1.458**	**0.564**	**1.127**	**7**	**3.5**
TD2-M2	W1	1.720	2.064	1.032	0.833	1.667	0.559	1.117	6	3
	W2	1.516	1.820	0.910	0.833	1.667	0.550	1.099	6	3
	W3	1.303	2.085	1.043	0.625	1.250	0.537	1.074	8	4
	W4	1.333	2.133	1.066	0.625	1.250	0.525	1.051	8	4
	W5	1.314	2.102	1.051	0.625	1.250	0.551	1.102	8	4
	W6	1.616	1.939	0.970	0.833	1.667	0.516	1.031	6	3
	Average	**1.467**	**2.024**	**1.012**	**0.729**	**1.458**	**0.539**	**1.079**	**7**	**3.5**

A.3 Spatiotemporal Parameters CP3

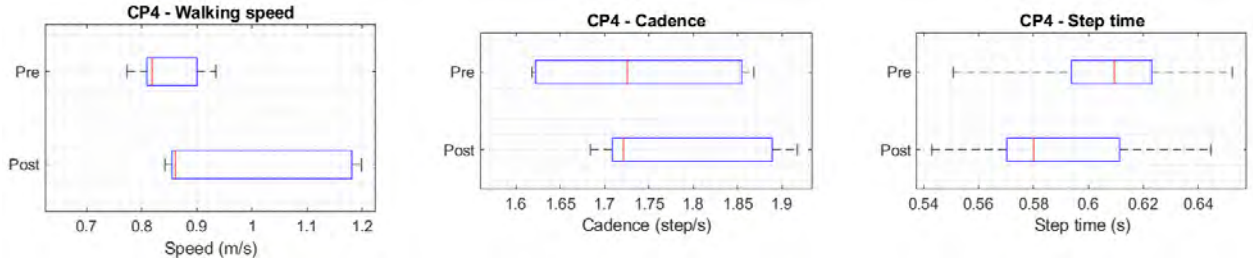






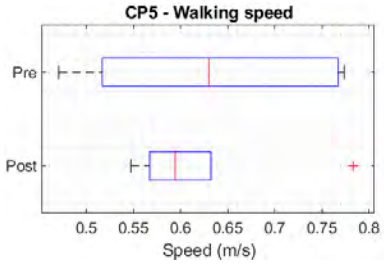
Dataset	Walk Nr.	Speed (m/s)	Cadence (s^{-1})		Length (m)		Time (s)		Number	
			Step	Stride	Step	Stride	Step	Stride	Step	Stride
CP3-Pre	W1	0.662	1.853	0.926	0.357	0.714	0.540	1.079	14	7
	W2	0.751	1.803	0.901	0.417	0.833	0.555	1.110	12	6
	W3	0.761	1.826	0.913	0.417	0.833	0.548	1.095	12	6
	W4	0.898	1.795	0.898	0.500	1.000	0.557	1.114	10	5
	W5	0.912	1.823	0.912	0.500	1.000	0.549	1.097	10	5
	W6	0.826	1.982	0.991	0.417	0.833	0.555	1.111	12	6
	Average	**0.801**	**1.847**	**0.923**	**0.435**	**0.869**	**0.550**	**1.100**	**11.7**	**5.8**
CP3-Post	W1	0.583	1.867	0.933	0.313	0.625	0.571	1.142	16	8
	W2	0.915	1.831	0.915	0.500	1.000	0.546	1.092	10	5
	W3	0.761	1.826	0.913	0.417	0.833	0.548	1.095	12	6
	W4	0.834	2.002	1.001	0.417	0.833	0.553	1.106	12	6
	W5	0.918	1.836	0.918	0.500	1.000	0.545	1.089	10	5
	W6	0.917	1.834	0.917	0.500	1.000	0.545	1.091	10	5
	Average	**0.821**	**1.866**	**0.933**	**0.441**	**0.882**	**0.553**	**1.106**	**11.7**	**5.8**
TD3-M1	W1	1.205	1.928	0.964	0.625	1.250	0.596	1.192	8	4
	W2	1.808	2.169	1.085	0.833	1.667	0.559	1.117	6	3
	W3	1.287	2.060	1.030	0.625	1.250	0.562	1.123	8	4
	W4	1.542	1.850	0.925	0.833	1.667	0.540	1.081	6	3
	W5	1.471	1.765	0.883	0.833	1.667	0.567	1.133	6	3
	W6	1.524	1.828	0.914	0.833	1.667	0.547	1.094	6	3
	Average	**1.473**	**1.933**	**0.967**	**0.764**	**1.528**	**0.563**	**1.127**	**6.7**	**3.3**
TD3-M2	W1	1.066	1.706	0.853	0.625	1.250	0.586	1.172	8	4
	W2	1.506	1.807	0.903	0.833	1.667	0.553	1.107	6	3
	W3	1.557	1.868	0.934	0.833	1.667	0.535	1.070	6	3
	W4	1.516	1.820	0.910	0.833	1.667	0.550	1.099	6	3
	W5	1.425	1.710	0.855	0.833	1.667	0.585	1.169	6	3
	W6	1.247	1.996	0.998	0.625	1.250	0.574	1.149	8	4
	Average	**1.386**	**1.818**	**0.909**	**0.764**	**1.528**	**0.566**	**1.131**	**6.7**	**3.3**

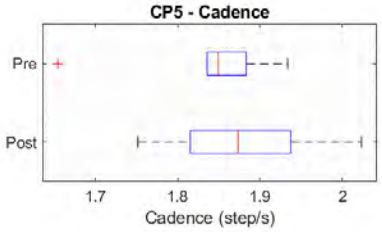
A.4 Spatiotemporal Parameters CP4

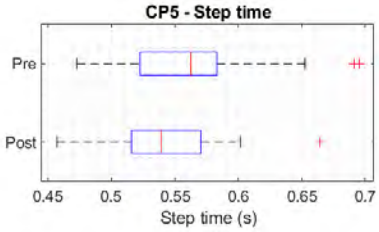


Dataset	Walk Nr.	Speed (m/s)	Cadence (s^{-1})		Length (m)		Time (s)		Number	
			Step	Stride	Step	Stride	Step	Stride	Step	Stride
CP4-Pre	W1	0.934	1.868	0.934	0.500	1.000	0.592	1.185	10	5
	W2	0.773	1.855	0.927	0.417	0.833	0.585	1.171	12	6
	W3	0.811	1.622	0.811	0.500	1.000	0.616	1.233	10	5
	W4	0.826	1.651	0.826	0.500	1.000	0.606	1.211	10	5
	W5	0.900	1.800	0.900	0.500	1.000	0.625	1.250	10	5
	W6	0.809	1.618	0.809	0.500	1.000	0.618	1.236	10	5
	Average	**0.842**	**1.736**	**0.868**	**0.486**	**0.972**	**0.606**	**1.213**	**10.3**	**5.2**
CP4-Post	W1	1.181	1.889	0.945	0.625	1.250	0.604	1.207	8	4
	W2	1.198	1.917	0.959	0.625	1.250	0.597	1.194	8	4
	W3	0.859	1.718	0.859	0.500	1.000	0.582	1.164	10	5
	W4	0.862	1.725	0.862	0.500	1.000	0.580	1.160	10	5
	W5	0.854	1.709	0.854	0.500	1.000	0.585	1.170	10	5
	W6	0.842	1.684	0.842	0.500	1.000	0.594	1.188	10	5
	Average	**0.966**	**1.774**	**0.887**	**0.542**	**1.083**	**0.589**	**1.179**	**9.3**	**4.7**
TD4-M1	W1	1.238	1.980	0.990	0.625	1.250	0.505	1.010	8	4
	W2	1.432	2.290	1.145	0.625	1.250	0.502	1.004	8	4
	W3	1.557	1.868	0.934	0.833	1.667	0.535	1.071	6	3
	W4	1.645	1.974	0.987	0.833	1.667	0.591	1.183	6	3
	W5	1.403	2.245	1.123	0.625	1.250	0.502	1.004	8	4
	W6	1.711	2.053	1.027	0.833	1.667	0.487	0.974	6	3
	Average	**1.498**	**2.069**	**1.034**	**0.729**	**1.458**	**0.518**	**1.036**	**7**	**3.5**
TD4-M2	W1	1.716	2.059	1.029	0.833	1.667	0.486	0.972	6	3
	W2	1.684	2.694	1.347	0.625	1.250	0.456	0.912	8	4
	W3	1.693	2.709	1.354	0.625	1.250	0.416	0.832	8	4
	W4	1.492	2.387	1.193	0.625	1.250	0.419	0.838	8	4
	W5	1.513	2.420	1.210	0.625	1.250	0.495	0.990	8	4
	W6	1.688	2.026	1.013	0.833	1.667	0.494	0.987	6	3
	Average	**1.631**	**2.382**	**1.191**	**0.694**	**1.389**	**0.458**	**0.917**	**7.3**	**3.7**

A.5 Spatiotemporal Parameters CP5

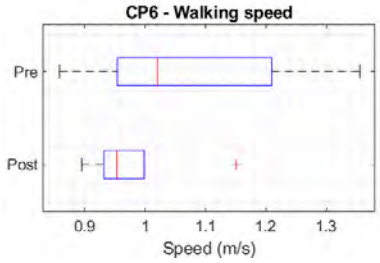


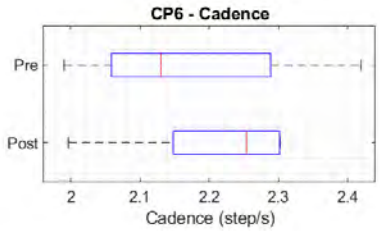


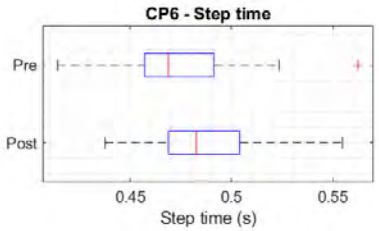


Dataset	Walk Nr.	Speed (m/s)	Cadence (s^{-1})		Length (m)		Time (s)		Number	
			Step	Stride	Step	Stride	Step	Stride	Step	Stride
CP5-Pre	W1	0.517	1.654	0.827	0.313	0.625	0.605	1.209	16	8
	W2	0.471	1.884	0.942	0.250	0.500	0.557	1.113	20	10
	W3	0.656	1.836	0.918	0.357	0.714	0.545	1.089	14	7
	W4	0.604	1.934	0.967	0.313	0.625	0.562	1.123	16	8
	W5	0.774	1.857	0.929	0.417	0.833	0.538	1.077	12	6
	W6	0.767	1.842	0.921	0.417	0.833	0.543	1.086	12	6
	Average	**0.631**	**1.834**	**0.917**	**0.344**	**0.688**	**0.560**	**1.120**	**15**	**7.5**
CP5-Post	W1	0.547	1.752	0.876	0.313	0.625	0.571	1.142	16	8
	W2	0.567	1.815	0.908	0.313	0.625	0.551	1.102	16	8
	W3	0.783	1.880	0.940	0.417	0.833	0.532	1.064	12	6
	W4	0.632	2.024	1.012	0.313	0.625	0.529	1.059	16	8
	W5	0.605	1.937	0.969	0.313	0.625	0.516	1.032	16	8
	W6	0.583	1.867	0.933	0.313	0.625	0.536	1.071	16	8
	Average	**0.620**	**1.879**	**0.940**	**0.330**	**0.660**	**0.539**	**1.079**	**15.3**	**7.7**
TD5-M1	W1	0.954	1.907	0.954	0.500	1.000	0.572	1.144	10	5
	W2	1.527	2.443	1.222	0.625	1.250	0.465	0.930	8	4
	W3	1.565	1.877	0.939	0.833	1.667	0.623	1.245	6	3
	W4	1.600	2.560	1.280	0.625	1.250	0.447	0.895	8	4
	W5	1.422	2.275	1.138	0.625	1.250	0.492	0.985	8	4
	W6	1.275	2.039	1.020	0.625	1.250	0.490	0.981	8	4
	Average	**1.390**	**2.184**	**1.092**	**0.639**	**1.278**	**0.513**	**1.026**	**8**	**4**
TD5-M2	W1	0.899	2.157	1.078	0.417	0.833	0.504	1.008	12	6
	W2	1.198	2.397	1.198	0.500	1.000	0.465	0.930	10	5
	W3	1.000	2.000	1.000	0.500	1.000	0.500	1.000	10	5
	W4	1.012	2.025	1.012	0.500	1.000	0.494	0.988	10	5
	W5	1.303	2.085	1.043	0.625	1.250	0.480	0.959	8	4
	W6	1.049	2.098	1.049	0.500	1.000	0.527	1.053	10	5
	Average	**1.077**	**2.127**	**1.063**	**0.507**	**1.014**	**0.496**	**0.991**	**10**	**5**

A.6 Spatiotemporal Parameters CP6



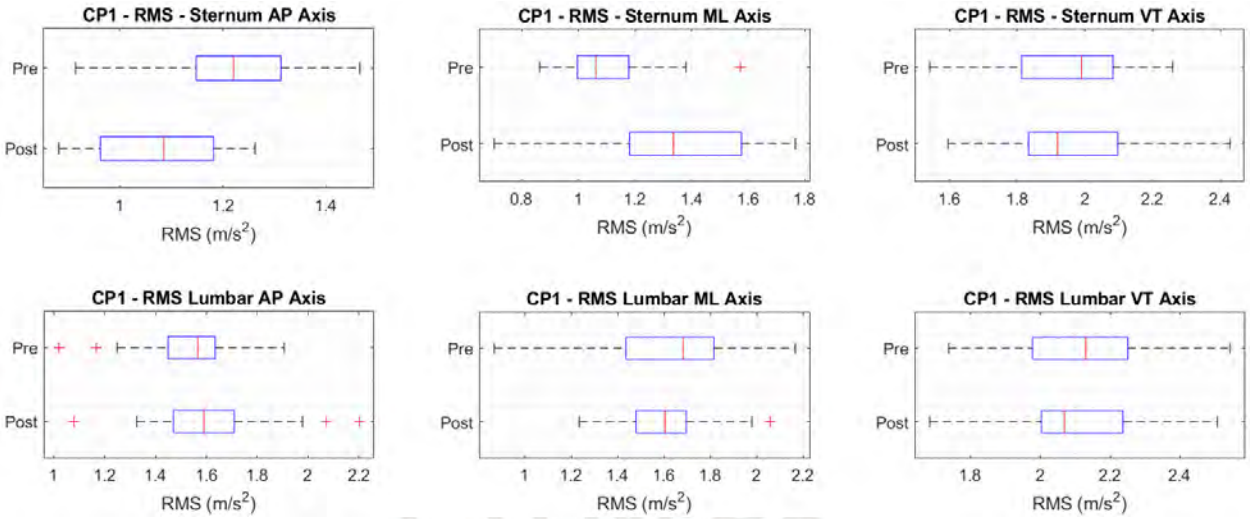




Dataset	Walk Nr.	Speed (m/s)	Cadence (s^{-1})		Length (m)		Time (s)		Number	
			Step	Stride	Step	Stride	Step	Stride	Step	Stride
CP6-Pre	W1	0.995	1.990	0.995	0.500	1.000	0.502	1.005	10	5
	W2	0.858	2.059	1.029	0.417	0.833	0.486	0.971	12	6
	W3	1.210	2.419	1.210	0.500	1.000	0.456	0.913	10	5
	W4	1.356	2.169	1.085	0.625	1.250	0.461	0.922	8	4
	W5	1.046	2.091	1.046	0.500	1.000	0.478	0.956	10	5
	W6	0.954	2.289	1.144	0.417	0.833	0.481	0.961	12	6
	Average	**1.070**	**2.170**	**1.085**	**0.493**	**0.986**	**0.478**	**0.957**	**10.3**	**5.2**
CP6-Post	W1	0.998	1.997	0.998	0.500	1.000	0.501	1.002	10	5
	W2	0.947	2.272	1.136	0.417	0.833	0.475	0.951	12	6
	W3	1.151	2.302	1.151	0.500	1.000	0.477	0.953	10	5
	W4	0.931	2.235	1.118	0.417	0.833	0.484	0.968	12	6
	W5	0.959	2.303	1.151	0.417	0.833	0.475	0.949	12	6
	W6	0.895	2.148	1.074	0.417	0.833	0.507	1.014	12	6
	Average	**0.980**	**2.209**	**1.105**	**0.444**	**0.889**	**0.486**	**0.973**	**11.3**	**5.7**
TD6-M1	W1	2.344	2.813	1.406	0.833	1.667	0.444	0.888	6	3
	W2	2.922	2.337	1.169	1.250	2.500	0.428	0.856	4	2
	W3	2.908	2.327	1.163	1.250	2.500	0.430	0.860	4	2
	W4	2.352	2.823	1.411	0.833	1.667	0.439	0.878	6	3
	W5	2.922	2.337	1.169	1.250	2.500	0.428	0.856	4	2
	W6	3.855	3.084	1.542	1.250	2.500	0.455	0.910	4	2
	Average	**2.884**	**2.620**	**1.310**	**1.111**	**2.222**	**0.438**	**0.876**	**4.7**	**2.3**
TD6-M2	W1	1.720	2.752	1.376	0.625	1.250	0.414	0.828	8	4
	W2	3.635	2.908	1.454	1.250	2.500	0.440	0.879	4	2
	W3	2.461	2.953	1.477	0.833	1.667	0.402	0.805	6	3
	W4	2.154	2.585	1.293	0.833	1.667	0.387	0.774	6	3
	W5	2.770	2.216	1.108	1.250	2.500	0.451	0.903	4	2
	W6	3.076	2.461	1.230	1.250	2.500	0.406	0.813	4	2
	Average	**2.636**	**2.646**	**1.323**	**1.007**	**2.014**	**0.414**	**0.827**	**5.3**	**2.7**

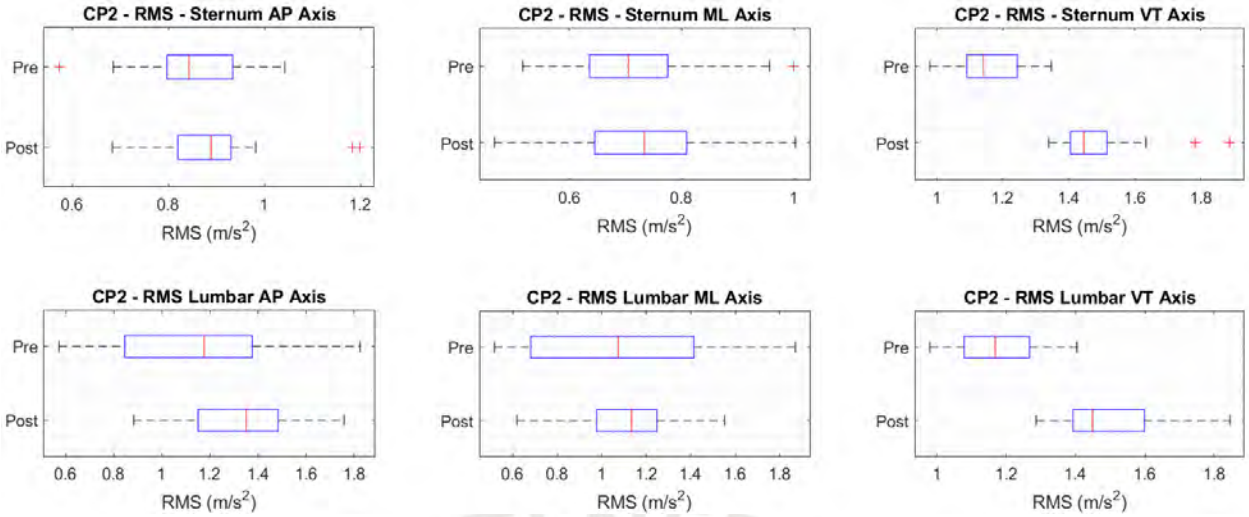
B RMS accelerations

B.1 RMS accelerations CP1



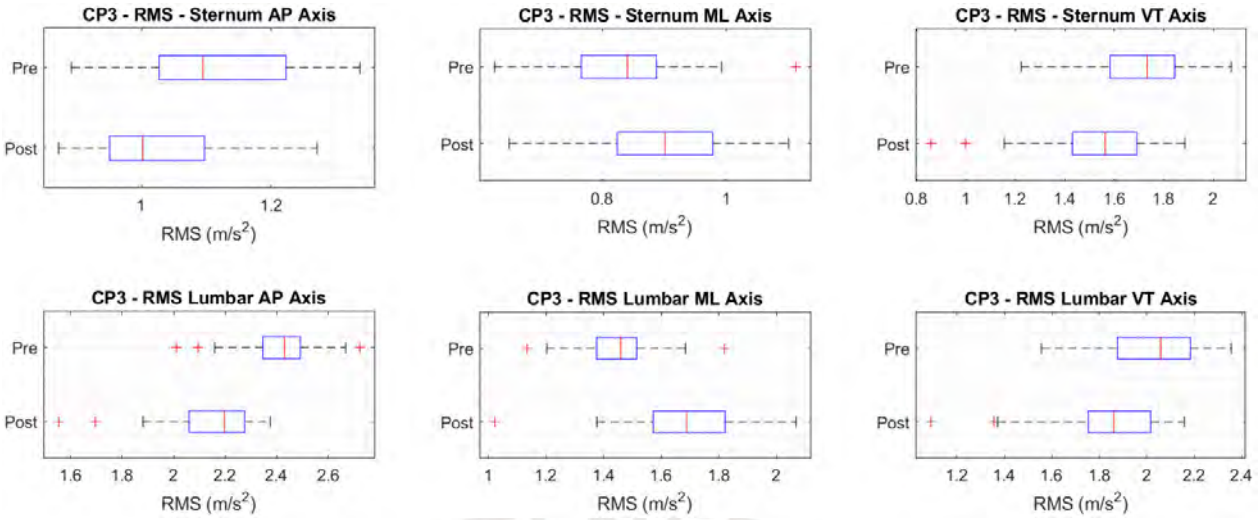
Dataset	Walk Nr.	Sternum RMS (m/s²)				Lumbar RMS (m/s²)			
		AP	ML	VT	Total	AP	ML	VT	Total
CP1-Pre	W1	1.313	1.129	1.789	2.490	1.561	1.725	2.064	3.110
	W2	1.338	0.933	2.150	2.699	1.687	1.914	2.365	3.479
	W3	1.191	1.004	2.073	2.593	1.686	1.900	2.402	3.496
	W4	1.178	0.963	1.987	2.503	1.542	1.809	2.105	3.175
	W5	1.305	1.019	1.953	2.560	1.593	1.709	2.190	3.202
	W6	1.218	1.117	1.930	2.541	1.658	1.757	2.018	3.148
	Average	**1.254**	**1.050**	**1.965**	**2.564**	**1.523**	**1.596**	**2.139**	**3.086**
CP1-Post	W1	1.088	1.521	1.851	2.631	1.685	1.716	2.011	3.136
	W2	1.175	1.358	2.074	2.744	1.517	1.473	2.314	3.134
	W3	1.084	1.366	2.058	2.697	1.460	1.649	2.063	3.017
	W4	1.042	1.002	1.897	2.385	1.567	1.597	2.008	3.006
	W5	1.011	1.338	1.903	2.537	1.871	1.646	2.051	3.227
	W6	1.099	1.279	2.050	2.654	1.508	1.805	2.044	3.116
	Average	**1.076**	**1.339**	**1.949**	**2.610**	**1.611**	**1.627**	**2.083**	**3.106**
TD1-M1	W1	1.400	1.251	2.297	2.967	1.830	2.405	2.345	3.825
	W2	1.238	1.024	2.065	2.616	1.553	2.094	2.204	3.414
	W3	1.055	0.853	1.936	2.365	1.584	2.036	2.208	3.395
	W4	1.095	0.947	1.915	2.401	1.639	2.155	2.144	3.454
	W5	1.287	0.947	1.900	2.482	1.537	2.087	2.164	3.377
	W6	1.298	0.941	1.717	2.349	1.535	2.051	1.984	3.241
	Average	**1.208**	**1.008**	**1.949**	**2.512**	**1.448**	**1.722**	**2.073**	**3.097**
TD1-M2	W1	1.421	1.083	2.346	2.949	2.115	2.028	2.849	4.087
	W2	1.514	1.079	2.561	3.165	1.879	1.978	2.897	3.980
	W3	1.538	1.170	2.251	2.967	1.821	2.037	2.717	3.853
	W4	1.369	1.093	2.357	2.936	1.575	1.824	2.726	3.638
	W5	1.454	1.166	2.347	2.997	1.953	1.966	2.780	3.925
	W6	1.589	0.928	2.073	2.772	1.621	1.698	2.525	3.448
	Average	**1.459**	**1.069**	**2.289**	**2.927**	**1.747**	**1.681**	**2.627**	**3.592**

B.2 RMS accelerations CP2



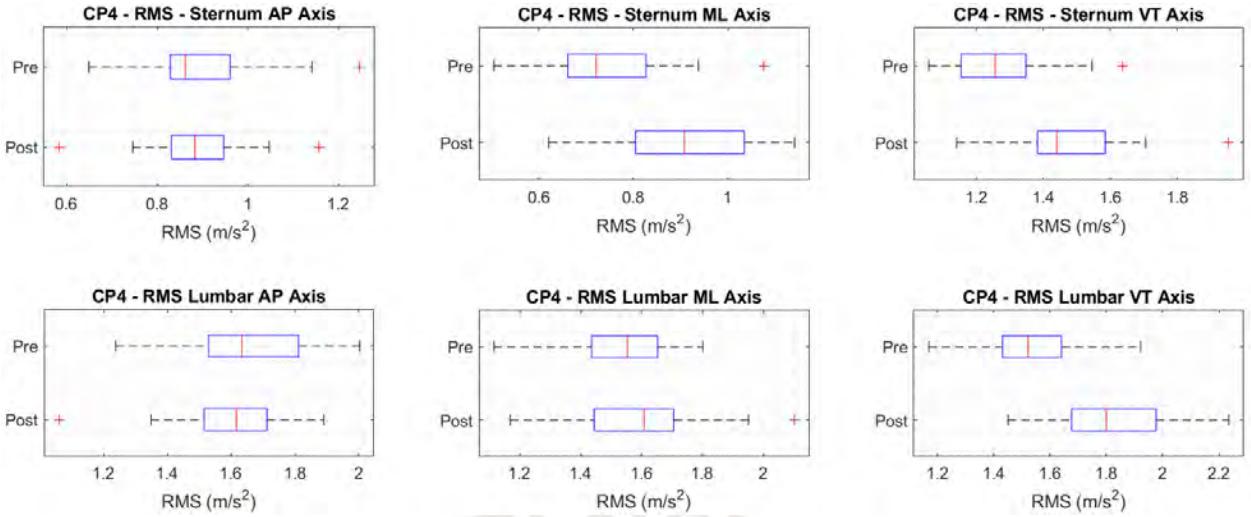
Dataset	Walk Nr.	Sternum RMS (m/s²)				Lumbar RMS (m/s²)			
		AP	ML	VT	Total	AP	ML	VT	Total
CP2-Pre	W1	0.836	0.705	1.076	1.535	1.305	1.321	1.183	2.202
	W2	0.824	0.592	1.060	1.468	1.312	1.350	0.998	2.130
	W3	0.810	0.889	1.241	1.729	1.395	1.313	1.318	2.325
	W4	0.891	0.703	1.198	1.650	1.365	1.286	1.167	2.209
	W5	0.905	0.618	1.143	1.584	1.503	1.758	1.297	2.652
	W6	0.896	0.854	1.214	1.734	1.565	1.523	1.307	2.545
	Average	**0.856**	**0.720**	**1.150**	**1.612**	**1.135**	**1.056**	**1.176**	**1.971**
CP2-Post	W1	0.766	0.791	1.618	1.957	1.464	1.011	1.615	2.403
	W2	0.837	0.579	1.595	1.892	1.450	1.351	1.672	2.593
	W3	0.859	0.684	1.456	1.824	1.325	1.149	1.521	2.322
	W4	1.042	0.891	1.458	2.001	1.478	1.228	1.423	2.391
	W5	0.997	0.660	1.393	1.836	1.424	1.194	1.328	2.284
	W6	0.916	0.804	1.399	1.855	1.457	1.212	1.442	2.381
	Average	**0.894**	**0.719**	**1.470**	**1.874**	**1.311**	**1.100**	**1.496**	**2.287**
TD2-M1	W1	1.478	1.037	1.307	2.229	1.558	1.465	1.485	2.603
	W2	1.600	1.257	1.737	2.675	1.586	1.528	1.887	2.900
	W3	1.339	1.025	1.758	2.437	1.548	1.465	1.799	2.789
	W4	1.662	1.063	1.787	2.662	1.555	1.576	1.905	2.921
	W5	1.898	1.032	2.062	2.987	1.716	1.697	2.147	3.231
	W6	1.187	1.036	1.609	2.252	1.729	1.587	1.699	2.897
	Average	**1.470**	**1.094**	**1.679**	**2.501**	**1.584**	**1.361**	**1.767**	**2.754**
TD2-M2	W1	1.318	1.005	1.444	2.198	1.938	1.351	1.522	2.810
	W2	1.186	1.227	1.464	2.248	1.847	1.344	1.606	2.792
	W3	1.190	1.014	1.338	2.058	1.946	1.475	1.455	2.843
	W4	1.210	1.017	1.372	2.093	1.787	1.335	1.469	2.671
	W5	1.427	1.114	1.423	2.303	1.804	1.351	1.609	2.769
	W6	0.915	0.830	1.412	1.876	1.983	1.269	1.636	2.867
	Average	**1.184**	**1.052**	**1.387**	**2.118**	**1.588**	**1.242**	**1.489**	**2.523**

B.3 RMS accelerations CP3



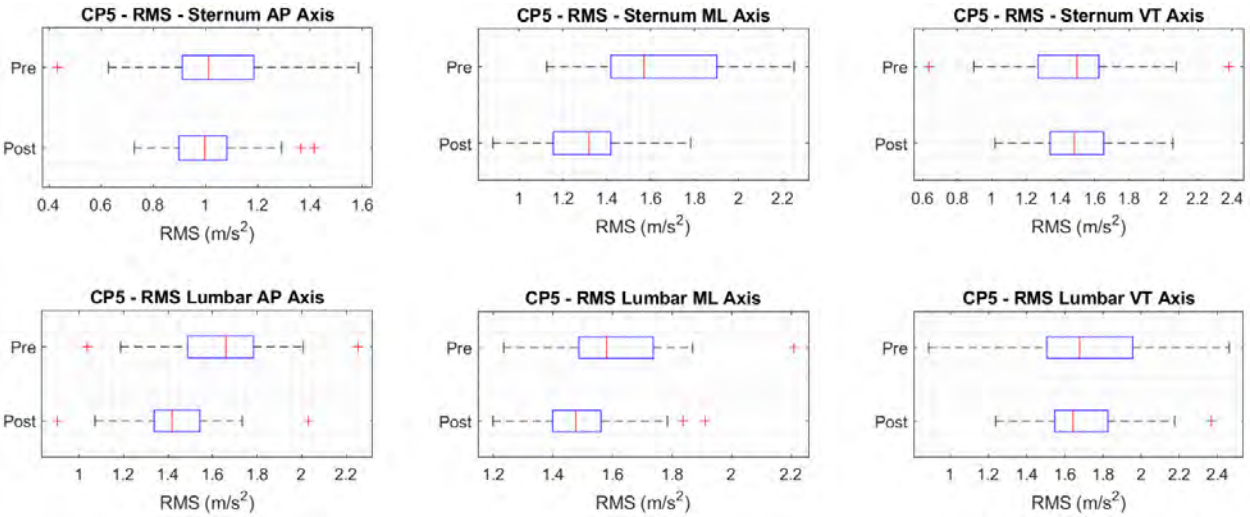
Dataset	Walk Nr.	Sternum RMS (m/s²)				Lumbar RMS (m/s²)			
		AP	ML	VT	Total	AP	ML	VT	Total
CP3-Pre	W1	1.064	0.793	1.620	2.094	2.274	1.434	2.009	3.356
	W2	1.061	0.869	1.650	2.145	2.359	1.373	1.995	3.381
	W3	1.137	0.860	1.689	2.211	2.484	1.479	2.064	3.552
	W4	1.125	0.872	1.718	2.231	2.389	1.394	1.919	3.367
	W5	1.245	0.883	1.811	2.369	2.496	1.452	2.171	3.613
	W6	1.089	0.797	1.783	2.236	2.507	1.550	2.132	3.638
	Average	**1.108**	**0.839**	**1.696**	**2.199**	**2.407**	**1.444**	**2.030**	**3.468**
CP3-Post	W1	1.025	0.794	1.399	1.907	2.008	1.479	1.675	3.004
	W2	1.049	0.932	1.632	2.153	2.136	1.614	1.936	3.304
	W3	1.026	0.914	1.602	2.111	2.178	1.717	1.869	3.344
	W4	1.137	0.933	1.687	2.238	2.288	1.815	2.038	3.561
	W5	0.960	0.897	1.546	2.030	2.292	1.749	1.868	3.435
	W6	1.009	0.903	1.606	2.101	2.215	1.767	1.836	3.377
	Average	**1.026**	**0.894**	**1.548**	**2.070**	**2.157**	**1.672**	**1.835**	**3.294**
TD3-M1	W1	0.897	0.991	1.367	1.912	1.749	1.234	1.278	2.493
	W2	0.822	1.022	1.566	2.043	1.677	1.269	1.514	2.591
	W3	0.900	0.760	1.476	1.888	1.776	1.445	1.378	2.672
	W4	0.776	0.757	1.623	1.952	1.931	1.304	1.477	2.759
	W5	0.872	0.735	1.496	1.881	1.991	1.395	1.388	2.800
	W6	0.778	0.732	1.382	1.747	1.864	1.420	1.365	2.712
	Average	**0.806**	**0.858**	**1.456**	**1.883**	**1.311**	**1.093**	**1.412**	**2.272**
TD3-M2	W1	0.935	0.743	1.491	1.910	2.218	1.761	1.484	3.197
	W2	0.865	0.847	1.473	1.907	2.062	1.788	1.492	3.110
	W3	0.905	0.854	1.685	2.094	2.176	1.932	1.689	3.365
	W4	0.747	0.749	1.443	1.789	2.238	1.913	1.415	3.266
	W5	0.830	0.949	1.567	2.012	2.145	1.978	1.613	3.334
	W6	0.833	0.881	1.628	2.030	2.169	1.912	1.668	3.338
	Average	**0.849**	**0.822**	**1.543**	**1.954**	**1.533**	**1.348**	**1.550**	**2.626**

B.4 RMS accelerations CP4



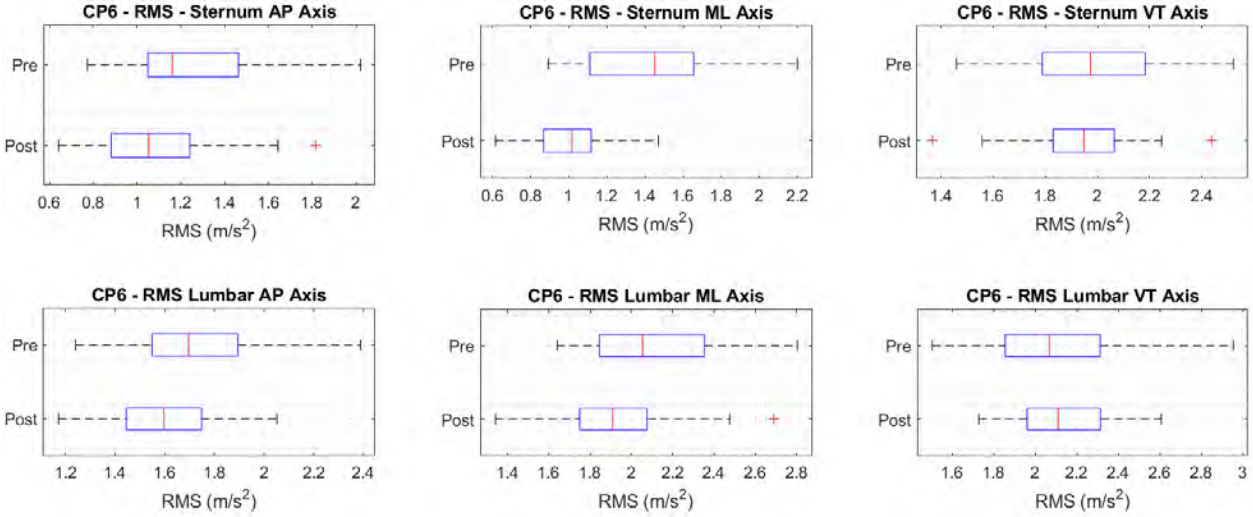
Dataset	Walk Nr.	Sternum RMS (m/s²)				Lumbar RMS (m/s²)			
		AP	ML	VT	Total	AP	ML	VT	Total
CP4-Pre	W1	0.962	0.666	1.451	1.864	1.617	1.493	1.723	2.795
	W2	1.008	0.797	1.299	1.827	1.690	1.580	1.514	2.765
	W3	0.835	0.787	1.286	1.724	1.758	1.675	1.542	2.877
	W4	1.002	0.681	1.267	1.753	1.714	1.476	1.489	2.708
	W5	0.850	0.649	1.207	1.613	1.547	1.528	1.563	2.678
	W6	0.820	0.758	1.223	1.656	1.397	1.463	1.514	2.527
	Average	**0.895**	**0.731**	**1.288**	**1.739**	**1.618**	**1.520**	**1.548**	**2.716**
CP4-Post	W1	0.875	0.871	1.551	1.983	1.610	1.522	1.819	2.867
	W2	0.931	0.960	1.456	1.977	1.730	1.560	1.826	2.960
	W3	0.886	1.049	1.434	1.986	1.657	1.558	1.842	2.927
	W4	0.861	0.955	1.394	1.897	1.631	1.511	1.826	2.878
	W5	0.947	0.914	1.463	1.968	1.464	1.763	1.863	2.953
	W6	0.865	0.869	1.479	1.922	1.679	1.596	1.838	2.958
	Average	**0.881**	**0.944**	**1.474**	**1.967**	**1.632**	**1.603**	**1.835**	**2.941**
TD4-M1	W1	0.713	0.612	1.882	2.104	1.494	1.379	2.033	2.875
	W2	0.849	0.567	1.907	2.163	1.676	1.665	1.955	3.067
	W3	0.729	0.592	1.950	2.164	1.666	1.756	1.988	3.132
	W4	0.808	0.590	1.798	2.057	1.692	1.703	1.801	3.001
	W5	0.784	0.615	1.920	2.163	1.608	1.543	1.987	2.985
	W6	0.838	0.585	1.966	2.216	1.671	1.548	1.999	3.031
	Average	**0.766**	**0.593**	**1.883**	**2.120**	**1.306**	**1.178**	**1.926**	**2.653**
TD4-M2	W1	0.734	0.532	1.744	1.966	1.339	1.429	1.897	2.726
	W2	0.771	0.605	1.904	2.142	1.623	1.579	2.015	3.031
	W3	0.676	0.517	1.923	2.103	1.638	1.570	2.129	3.111
	W4	0.896	0.454	1.774	2.039	1.746	1.623	1.896	3.046
	W5	0.767	0.480	1.810	2.024	1.637	1.496	2.021	3.001
	W6	0.805	0.586	1.817	2.072	1.648	1.479	2.152	3.088
	Average	**0.759**	**0.530**	**1.820**	**2.049**	**1.416**	**1.287**	**1.949**	**2.766**

B.5 RMS accelerations CP5



Dataset	Walk Nr.	Sternum RMS (m/s²)				Lumbar RMS (m/s²)			
		AP	ML	VT	Total	AP	ML	VT	Total
CP5-Pre	W1	0.897	1.582	1.195	2.177	1.531	1.568	1.401	2.601
	W2	1.046	1.528	1.299	2.262	1.540	1.589	1.564	2.710
	W3	0.948	1.884	1.554	2.620	1.754	1.584	1.743	2.937
	W4	1.116	1.895	1.522	2.674	1.626	1.600	1.767	2.885
	W5	1.079	1.750	1.779	2.719	1.797	1.701	1.976	3.167
	W6	1.291	1.515	1.699	2.617	1.734	1.695	1.920	3.093
	Average	**1.036**	**1.668**	**1.458**	**2.471**	**1.645**	**1.602**	**1.688**	**2.861**
CP5-Post	W1	0.973	1.264	1.363	2.098	1.308	1.538	1.543	2.541
	W2	1.022	1.343	1.330	2.149	1.410	1.461	1.560	2.561
	W3	1.005	1.301	1.658	2.335	1.454	1.483	1.755	2.719
	W4	1.108	1.372	1.621	2.396	1.486	1.544	1.794	2.794
	W5	1.013	1.269	1.637	2.306	1.527	1.476	1.776	2.768
	W6	0.992	1.353	1.640	2.346	1.389	1.481	1.862	2.755
	Average	**1.015**	**1.299**	**1.524**	**2.260**	**1.431**	**1.501**	**1.707**	**2.696**
TD5-M1	W1	1.070	0.877	1.546	2.074	1.433	1.248	1.764	2.593
	W2	1.429	0.791	1.726	2.376	1.676	1.207	2.024	2.891
	W3	1.774	0.958	1.931	2.791	1.818	1.460	2.104	3.141
	W4	1.759	1.129	1.926	2.842	1.957	1.533	2.208	3.325
	W5	1.750	1.099	1.646	2.642	1.858	1.449	1.977	3.076
	W6	1.570	0.915	2.065	2.751	1.802	1.674	2.360	3.409
	Average	**1.552**	**0.958**	**1.784**	**2.570**	**1.733**	**1.363**	**2.026**	**3.006**
TD5-M2	W1	0.970	0.811	1.464	1.934	1.665	1.527	1.729	2.845
	W2	1.077	0.654	1.735	2.144	1.724	1.437	1.864	2.917
	W3	1.198	0.864	1.495	2.102	1.621	1.498	1.731	2.806
	W4	0.961	0.710	1.676	2.059	1.719	1.504	1.732	2.866
	W5	0.893	0.977	1.788	2.225	1.769	1.798	2.026	3.236
	W6	1.106	1.136	1.684	2.313	1.934	1.702	1.891	3.196
	Average	**1.020**	**0.852**	**1.616**	**2.117**	**1.727**	**1.552**	**1.793**	**2.944**

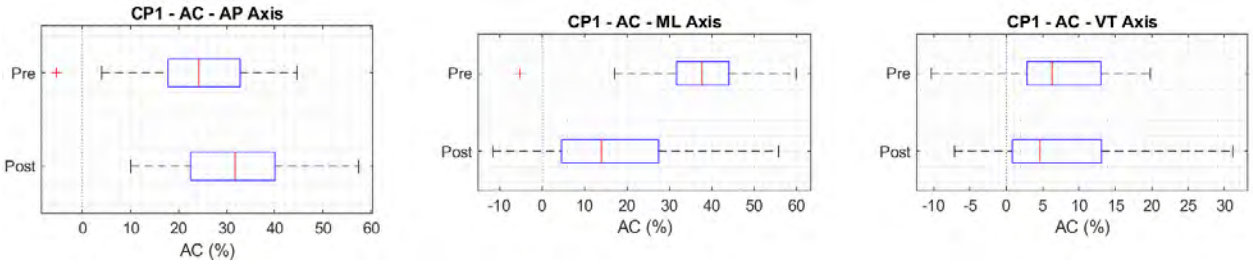
B.6 RMS accelerations CP6



Dataset	Walk Nr.	Sternum RMS (m/s²)				Lumbar RMS (m/s²)			
		AP	ML	VT	Total	AP	ML	VT	Total
CP-Pre	W1	0.963	1.237	1.709	2.319	1.625	2.108	1.794	3.210
	W2	1.197	1.392	2.014	2.725	1.745	2.072	2.029	3.385
	W3	1.513	1.948	2.109	3.245	2.045	2.687	2.433	4.162
	W4	1.357	1.504	2.197	2.989	1.744	2.019	2.226	3.475
	W5	1.671	1.710	2.066	3.160	1.908	2.145	2.415	3.751
	W6	1.271	1.368	1.839	2.621	1.629	2.068	2.042	3.331
	Average	**1.298**	**1.476**	**1.979**	**2.809**	**1.762**	**2.154**	**2.140**	**3.523**
CP-Post	W1	0.891	0.953	1.865	2.277	1.429	1.752	2.087	3.076
	W2	1.052	0.925	2.020	2.458	1.661	1.946	2.130	3.329
	W3	1.374	1.170	2.025	2.713	1.585	1.868	2.115	3.236
	W4	1.226	1.027	1.981	2.546	1.526	1.853	2.096	3.187
	W5	0.952	1.029	2.138	2.557	1.625	1.884	2.233	3.343
	W6	0.937	1.041	1.795	2.277	1.637	1.923	1.919	3.172
	Average	**1.065**	**1.005**	**1.953**	**2.459**	**1.583**	**1.915**	**2.112**	**3.268**
TD1	W1	2.160	2.445	4.363	5.448	2.703	2.360	4.033	5.398
	W2	2.043	2.377	5.012	5.911	2.583	2.530	4.673	5.909
	W3	2.386	2.329	5.058	6.058	2.810	2.351	4.728	5.981
	W4	1.992	2.133	4.460	5.330	2.489	2.304	4.230	5.422
	W5	2.271	2.364	4.598	5.647	2.827	2.610	4.163	5.669
	W6	1.957	2.175	3.702	4.719	2.206	1.941	3.489	4.562
	Average	**2.136**	**2.199**	**4.366**	**5.345**	**2.419**	**2.297**	**4.074**	**5.286**
TD2	W1	2.084	2.190	3.677	4.760	2.191	2.231	3.383	4.607
	W2	2.265	3.131	4.917	6.254	2.844	2.572	4.522	5.929
	W3	2.276	2.618	4.866	5.975	2.360	2.838	4.171	5.570
	W4	2.270	3.085	4.771	6.118	2.464	2.766	4.193	5.595
	W5	2.446	2.697	4.508	5.795	2.100	2.423	4.198	5.283
	W6	2.465	2.625	4.527	5.785	2.418	2.470	4.244	5.474
	Average	**2.267**	**2.628**	**4.408**	**5.629**	**2.318**	**2.556**	**4.228**	**5.475**

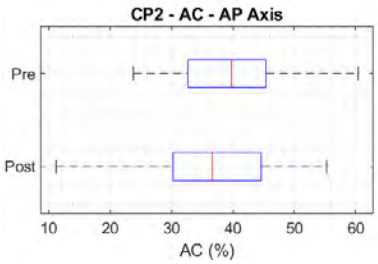
C. Attenuation coefficient

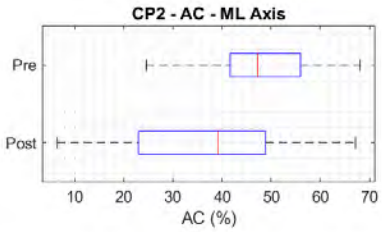
C.1 Attenuation coefficient CP1

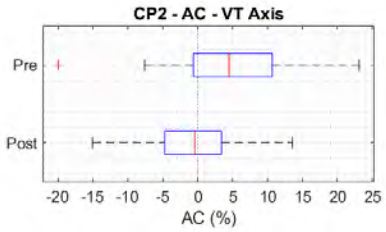


Dataset	Walk Nr.	AC (%)			
		AP	ML	VT	Total
CP1-Pre	W1	15.926	34.541	13.323	19.953
	W2	20.678	51.256	9.097	22.425
	W3	29.391	47.153	13.690	25.832
	W4	23.596	46.748	5.616	21.177
	W5	18.120	40.379	10.791	20.040
	W6	26.518	36.430	4.366	19.280
	Average	**22.372**	**42.751**	**9.481**	**21.451**
CP1-Post	W1	35.421	11.397	7.988	16.091
	W2	22.557	7.767	10.336	12.453
	W3	25.754	17.133	0.248	10.609
	W4	33.492	37.258	5.530	20.661
	W5	45.949	18.688	7.220	21.401
	W6	27.164	29.162	-0.310	14.824
	Average	**31.723**	**20.234**	**5.169**	**16.006**
TD1-M1	W1	23.467	48.008	2.036	22.440
	W2	20.285	51.094	6.329	23.367
	W3	33.382	58.093	12.307	30.363
	W4	33.180	56.054	10.651	30.476
	W5	16.302	54.647	12.222	26.499
	W6	15.474	54.121	13.446	27.505
	Average	**23.682**	**53.669**	**9.499**	**26.775**
TD1-M2	W1	32.827	46.597	17.644	27.843
	W2	19.445	45.471	11.597	20.481
	W3	15.516	42.564	17.156	23.002
	W4	13.071	40.074	13.544	19.290
	W5	25.535	40.693	15.569	23.640
	W6	2.015	45.340	17.915	19.617
	Average	**18.068**	**43.457**	**15.571**	**22.312**

C.2 Attenuation coefficient CP2

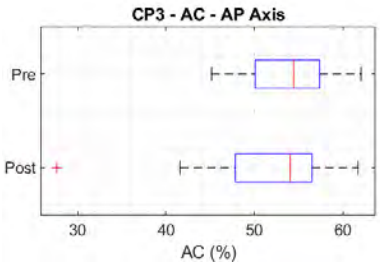


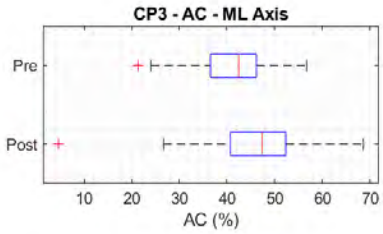


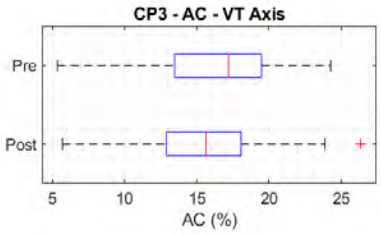


Dataset	Walk Nr.	AC (%)			
		AP	ML	VT	Total
CP2-Pre	W1	35.906	46.607	9.016	30.289
	W2	37.160	56.117	-6.236	31.102
	W3	41.910	32.263	5.834	25.662
	W4	34.743	45.347	-2.646	25.307
	W5	39.766	64.847	11.888	40.279
	W6	42.758	43.953	7.111	31.888
	Average	**38.707**	**48.189**	**4.161**	**30.754**
CP2-Post	W1	47.684	21.710	-0.161	18.554
	W2	42.285	57.151	4.601	27.032
	W3	35.150	40.448	4.260	21.428
	W4	29.524	27.450	-2.495	16.301
	W5	29.959	44.708	-4.906	19.612
	W6	37.177	33.662	3.008	22.113
	Average	**36.963**	**37.521**	**0.718**	**20.840**
TD2-M1	W1	5.116	29.224	12.013	14.395
	W2	-0.884	17.715	7.964	7.753
	W3	13.498	30.007	2.238	12.633
	W4	-6.864	32.551	6.193	8.868
	W5	-10.594	39.178	3.973	7.552
	W6	31.354	34.692	5.322	22.280
	Average	**5.271**	**30.561**	**6.284**	**12.247**
TD2-M2	W1	32.008	25.577	5.165	21.789
	W2	35.784	8.736	8.809	19.469
	W3	38.854	31.271	8.090	27.626
	W4	32.290	23.769	6.560	21.609
	W5	20.860	17.570	11.544	16.832
	W6	53.876	34.563	13.735	34.577
	Average	**35.612**	**23.581**	**8.984**	**23.650**

C.3 Attenuation coefficient CP3

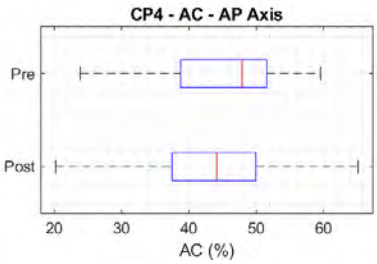


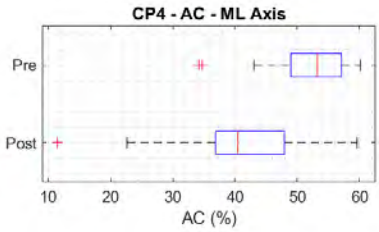


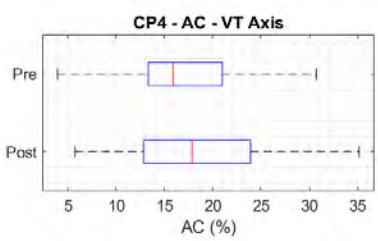


Dataset	Walk Nr.	AC (%)			
		AP	ML	VT	Total
CP3-Pre	W1	53.188	44.682	19.403	37.606
	W2	55.026	36.686	17.305	36.540
	W3	54.214	41.848	18.165	37.773
	W4	52.915	37.446	10.498	33.735
	W5	50.112	39.207	16.576	34.440
	W6	56.567	48.622	16.370	38.542
	Average	**53.670**	**41.415**	**16.386**	**36.439**
CP3-Post	W1	48.965	46.336	16.451	36.506
	W2	50.870	42.283	15.664	34.842
	W3	52.909	46.788	14.279	36.896
	W4	50.328	48.587	17.224	37.164
	W5	58.094	48.678	17.205	40.913
	W6	54.464	48.881	12.542	37.790
	Average	**52.605**	**46.925**	**15.561**	**37.352**
TD3-M1	W1	48.731	19.701	-6.945	23.329
	W2	50.986	19.456	-3.465	21.166
	W3	49.318	47.405	-7.059	29.344
	W4	59.832	41.959	-9.898	29.252
	W5	56.200	47.350	-7.813	32.804
	W6	58.254	48.442	-1.265	35.579
	Average	**53.887**	**37.386**	**-6.074**	**28.579**
TD3-M2	W1	57.837	57.789	-0.447	40.257
	W2	58.048	52.634	1.261	38.700
	W3	58.419	55.820	0.271	37.761
	W4	66.629	60.846	-1.988	45.222
	W5	61.288	52.001	2.858	39.665
	W6	61.611	53.926	2.424	39.196
	Average	**60.639**	**55.503**	**0.730**	**40.133**

C.4 Attenuation coefficient CP4

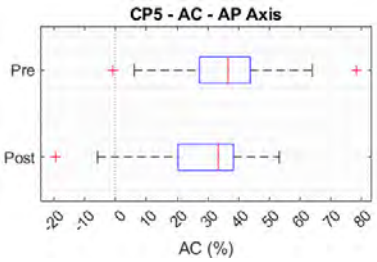


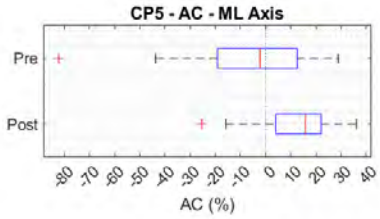


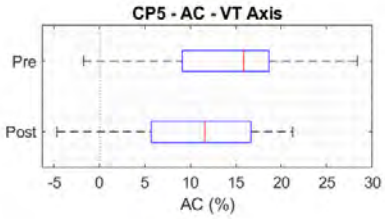


Dataset	Walk Nr.	AC (%)			
		AP	ML	VT	Total
CP4-Pre	W1	40.495	55.380	15.788	33.308
	W2	40.364	49.560	14.222	33.925
	W3	52.535	53.021	16.574	40.086
	W4	41.525	53.842	14.935	35.263
	W5	45.058	57.505	22.786	39.779
	W6	41.279	48.171	19.269	34.461
	Average	**43.543**	**52.913**	**17.262**	**36.137**
CP4-Post	W1	45.640	42.768	14.735	30.837
	W2	46.174	38.448	20.298	33.216
	W3	46.532	32.632	22.157	32.154
	W4	47.209	36.813	23.660	34.088
	W5	35.280	48.168	21.444	33.350
	W6	48.471	45.542	19.539	35.031
	Average	**44.884**	**40.728**	**20.305**	**33.113**
TD4-M1	W1	52.299	55.639	7.395	26.827
	W2	49.336	65.971	2.450	29.459
	W3	56.213	66.294	1.954	30.914
	W4	52.262	65.349	0.194	31.453
	W5	51.247	60.151	3.337	27.537
	W6	49.874	62.221	1.641	26.889
	Average	**51.872**	**62.604**	**2.829**	**28.846**
TD4-M2	W1	45.179	62.737	8.075	27.907
	W2	52.465	61.683	5.497	29.342
	W3	58.753	67.090	9.688	32.424
	W4	48.661	72.027	6.432	33.066
	W5	53.149	67.926	10.431	32.552
	W6	51.177	60.364	15.589	32.914
	Average	**51.564**	**65.304**	**9.285**	**31.367**

C.5 Attenuation coefficient CP5

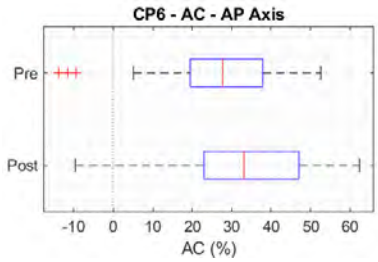


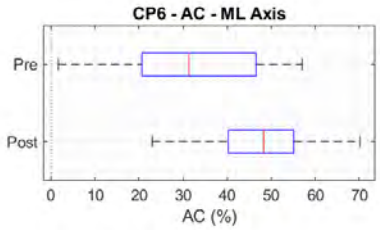


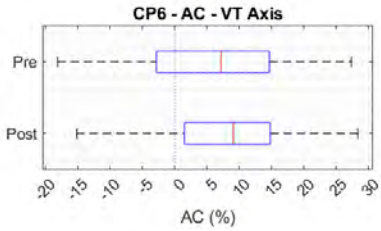


Dataset	Walk Nr.	AC (%)			
		AP	ML	VT	Total
CP5-Pre	W1	41.416	-0.887	14.658	16.326
	W2	32.066	3.831	16.946	16.524
	W3	45.934	-18.925	10.822	10.785
	W4	31.345	-18.411	13.866	7.315
	W5	39.946	-2.921	9.981	14.135
	W6	25.595	10.604	11.509	15.399
	Average	**36.051**	**-4.452**	**12.964**	**13.414**
CP5-Post	W1	25.574	17.781	11.664	17.408
	W2	27.544	8.094	14.730	16.092
	W3	30.888	12.306	5.479	14.125
	W4	25.398	11.136	9.605	14.263
	W5	33.662	13.997	7.818	16.704
	W6	28.591	8.597	11.936	14.836
	Average	**28.609**	**11.985**	**10.205**	**15.571**
TD5-M1	W1	25.370	29.732	12.370	20.001
	W2	14.712	34.480	14.703	17.810
	W3	2.428	34.390	8.263	11.132
	W4	10.135	26.376	12.741	14.514
	W5	5.809	24.170	16.748	14.111
	W6	12.874	45.375	12.495	19.307
	Average	**11.888**	**32.420**	**12.887**	**16.146**
TD5-M2	W1	41.753	46.922	15.327	32.018
	W2	37.519	54.524	6.910	26.500
	W3	26.099	42.326	13.670	25.094
	W4	44.081	52.787	3.196	28.177
	W5	49.542	45.666	11.742	31.240
	W6	42.802	33.264	10.948	27.623
	Average	**40.299**	**45.915**	**10.299**	**28.442**

C.6 Attenuation coefficient CP6



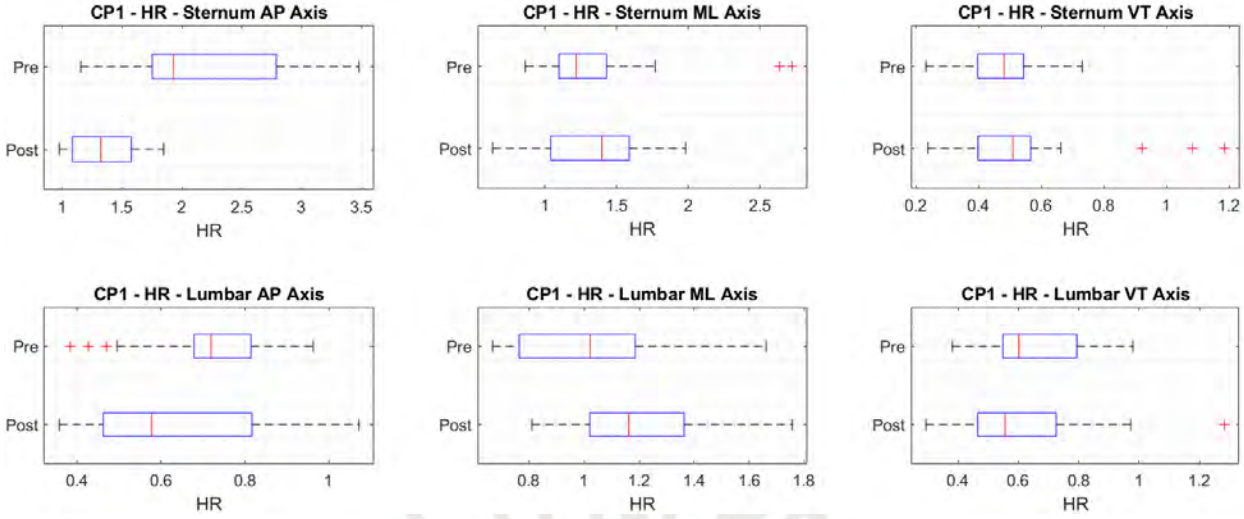




Dataset	Walk Nr.	AC (%)			
		AP	ML	VT	Total
CP6-Pre	W1	40.780	41.300	4.767	27.763
	W2	31.397	32.794	0.758	19.478
	W3	26.011	27.516	13.317	22.029
	W4	22.172	25.496	1.300	13.985
	W5	12.438	20.265	14.460	15.774
	W6	21.951	33.847	9.960	21.335
	Average	**25.792**	**30.203**	**7.427**	**20.061**
CP6-Post	W1	37.605	45.598	10.608	26.001
	W2	36.649	52.475	5.167	26.158
	W3	13.264	37.354	4.255	16.176
	W4	19.621	44.587	5.521	20.118
	W5	41.400	45.388	4.220	23.510
	W6	42.754	45.865	6.463	28.216
	Average	**31.882**	**45.211**	**6.039**	**23.363**
TD6-M1	W1	20.094	-3.607	-8.167	-0.911
	W2	20.904	6.066	-7.258	-0.049
	W3	15.076	0.955	-6.983	-1.283
	W4	19.961	7.452	-5.434	1.698
	W5	19.684	9.428	-10.456	0.386
	W6	11.278	-12.040	-6.093	-3.437
	Average	**17.833**	**1.376**	**-7.399**	**-0.599**
TD6-M2	W1	4.907	1.849	-8.671	-3.311
	W2	20.356	-21.724	-8.739	-5.480
	W3	3.551	7.779	-16.659	-7.287
	W4	7.901	-11.535	-13.802	-9.357
	W5	-16.450	-11.304	-7.388	-9.695
	W6	-1.938	-6.248	-6.680	-5.683
	Average	**3.054**	**-6.864**	**-10.323**	**-6.802**

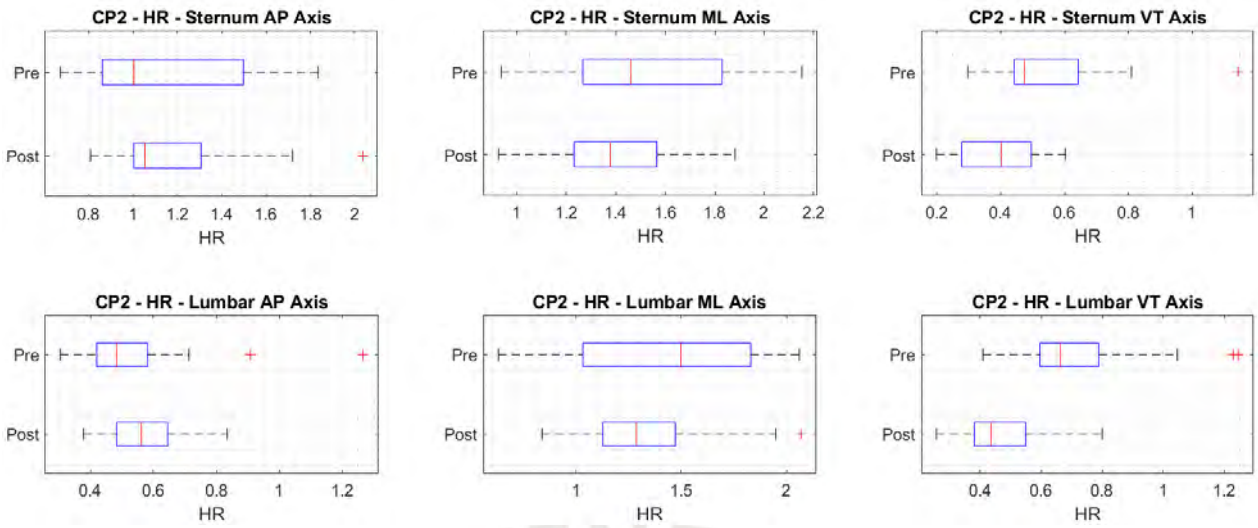
D Harmonic ratio

D.1 Harmonic ratio CP1



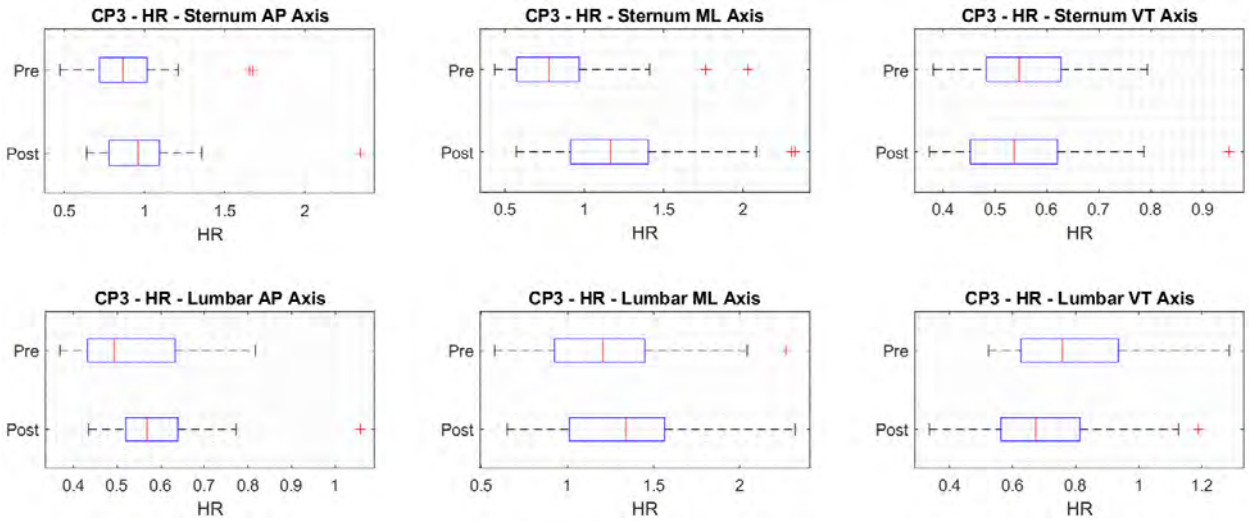
Dataset	Walk Nr.	Sternum HR			Lumbar HR		
		AP	ML	VT	AP	ML	VT
CP1-Pre	W1	1.564	1.877	0.494	1.564	1.877	0.494
	W2	2.123	1.089	0.494	2.123	1.089	0.494
	W3	1.833	1.310	0.456	1.833	1.310	0.456
	W4	2.935	1.427	0.480	2.935	1.427	0.480
	W5	2.414	1.124	0.479	2.414	1.124	0.479
	W6	2.399	1.210	0.447	2.399	1.210	0.447
	Average	**2.211**	**1.339**	**0.475**	**2.211**	**1.339**	**0.475**
CP1-Post	W1	1.299	1.541	0.524	1.299	1.541	0.524
	W2	1.567	1.351	0.653	1.567	1.351	0.653
	W3	1.306	1.323	0.408	1.306	1.323	0.408
	W4	1.208	1.089	0.695	1.208	1.089	0.695
	W5	1.249	1.099	0.438	1.249	1.099	0.438
	W6	1.250	1.388	0.433	1.250	1.388	0.433
	Average	**1.313**	**1.299**	**0.525**	**1.313**	**1.299**	**0.525**
TD1-M1	W1	0.807	1.711	0.625	0.807	1.711	0.625
	W2	0.913	1.054	0.641	0.913	1.054	0.641
	W3	0.923	1.437	0.512	0.923	1.437	0.512
	W4	1.356	1.122	0.238	1.356	1.122	0.238
	W5	1.236	1.473	0.290	1.236	1.473	0.290
	W6	0.893	1.440	0.342	0.893	1.440	0.342
	Average	**1.021**	**1.373**	**0.441**	**1.021**	**1.373**	**0.441**
TD1-M2	W1	0.834	1.412	0.580	0.834	1.412	0.580
	W2	0.987	1.461	0.497	0.987	1.461	0.497
	W3	0.754	0.995	0.546	0.754	0.995	0.546
	W4	0.831	1.159	0.543	0.831	1.159	0.543
	W5	0.918	1.521	0.304	0.918	1.521	0.304
	W6	0.997	1.688	0.246	0.997	1.688	0.246
	Average	**0.887**	**1.373**	**0.453**	**0.887**	**1.373**	**0.453**

D.2 Harmonic ratio CP2



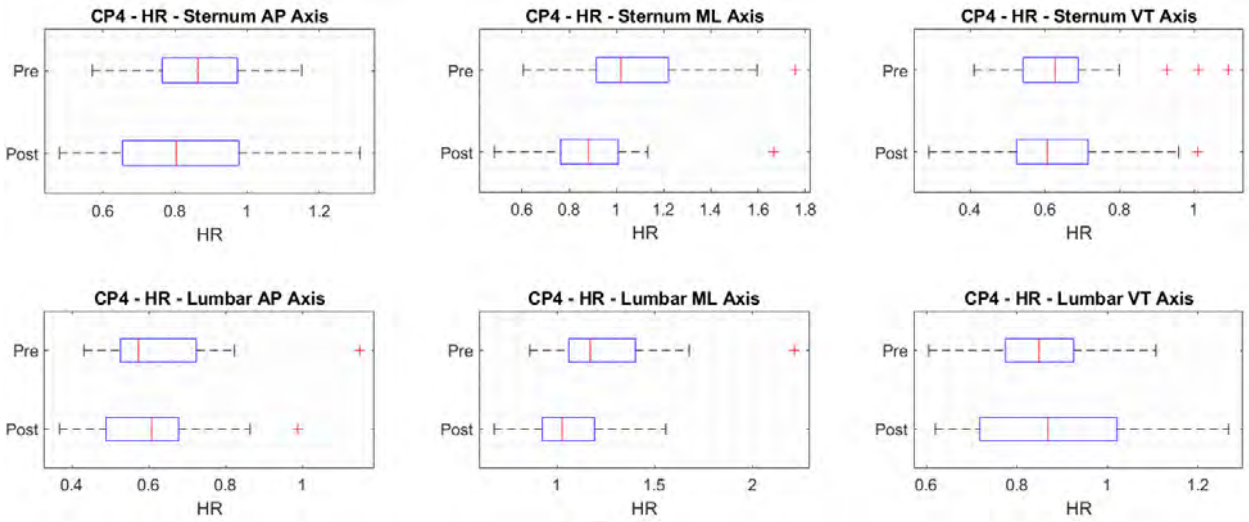
Dataset	Walk Nr.	Sternum HR			Lumbar HR		
		AP	ML	VT	AP	ML	VT
CP2-Pre	W1	0.906	1.648	0.365	0.906	1.648	0.365
	W2	1.469	1.778	0.471	1.469	1.778	0.471
	W3	1.124	1.619	0.428	1.124	1.619	0.428
	W4	1.429	1.474	0.464	1.429	1.474	0.464
	W5	1.149	1.279	0.591	1.149	1.279	0.591
	W6	0.850	1.411	0.497	0.850	1.411	0.497
	Average	**1.154**	**1.535**	**0.469**	**1.154**	**1.535**	**0.469**
CP2-Post	W1	1.159	1.396	0.419	1.159	1.396	0.419
	W2	1.583	1.251	0.355	1.583	1.251	0.355
	W3	1.127	1.335	0.514	1.127	1.335	0.514
	W4	0.991	1.699	0.273	0.991	1.699	0.273
	W5	0.960	1.487	0.428	0.960	1.487	0.428
	W6	1.225	1.384	0.344	1.225	1.384	0.344
	Average	**1.174**	**1.425**	**0.389**	**1.174**	**1.425**	**0.389**
TD2-M1	W1	0.751	1.165	0.767	0.751	1.165	0.767
	W2	0.585	1.363	0.532	0.585	1.363	0.532
	W3	0.633	1.138	0.361	0.633	1.138	0.361
	W4	0.567	1.545	0.332	0.567	1.545	0.332
	W5	0.774	1.422	0.584	0.774	1.422	0.584
	W6	0.649	0.975	0.507	0.649	0.975	0.507
	Average	**0.660**	**1.268**	**0.514**	**0.660**	**1.268**	**0.514**
TD2-M2	W1	0.747	0.723	0.694	0.747	0.723	0.694
	W2	0.563	1.061	0.569	0.563	1.061	0.569
	W3	0.752	0.950	0.630	0.752	0.950	0.630
	W4	0.422	0.820	0.411	0.422	0.820	0.411
	W5	0.543	0.836	0.629	0.543	0.836	0.629
	W6	0.612	0.967	0.289	0.612	0.967	0.289
	Average	**0.607**	**0.893**	**0.537**	**0.607**	**0.893**	**0.537**

D.3 Harmonic ratio CP3



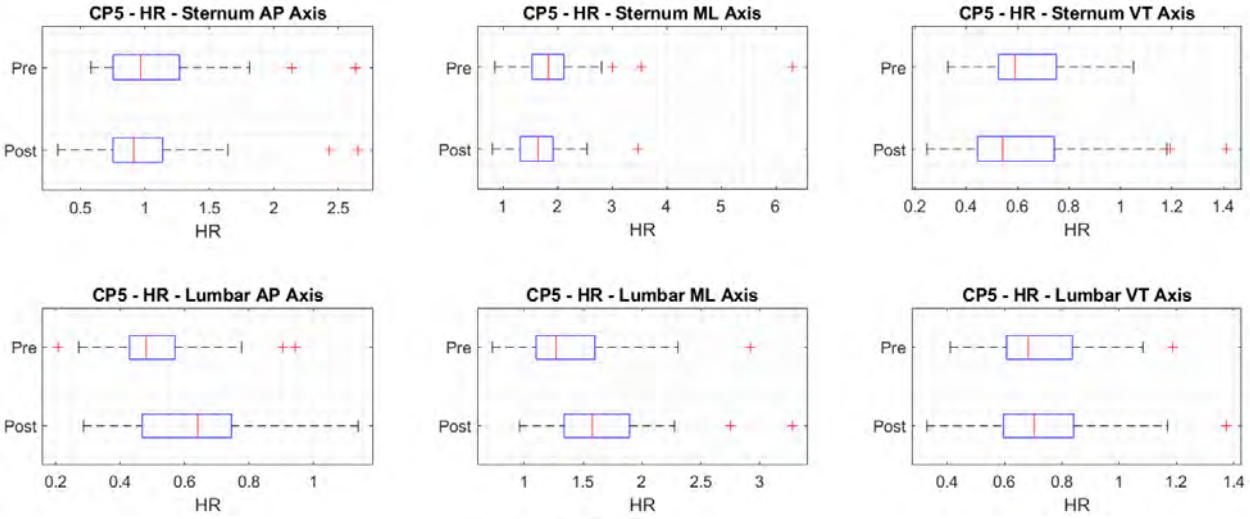
Dataset	Walk Nr.	Sternum HR			Lumbar HR		
		AP	ML	VT	AP	ML	VT
CP3-Pre	W1	0.857	0.696	0.552	0.857	0.696	0.552
	W2	0.841	0.780	0.590	0.841	0.780	0.590
	W3	0.747	0.670	0.583	0.747	0.670	0.583
	W4	0.900	0.802	0.549	0.900	0.802	0.549
	W5	0.740	0.731	0.543	0.740	0.731	0.543
	W6	1.139	1.218	0.527	1.139	1.218	0.527
	Average	**0.871**	**0.816**	**0.557**	**0.871**	**0.816**	**0.557**
CP3-Post	W1	0.939	1.108	0.517	0.939	1.108	0.517
	W2	0.911	1.290	0.527	0.911	1.290	0.527
	W3	0.887	1.102	0.493	0.887	1.102	0.493
	W4	1.119	1.282	0.592	1.119	1.282	0.592
	W5	0.888	0.986	0.531	0.888	0.986	0.531
	W6	1.146	1.469	0.651	1.146	1.469	0.651
	Average	**0.982**	**1.206**	**0.552**	**0.982**	**1.206**	**0.552**
TD3-M1	W1	1.116	2.359	0.347	1.116	2.359	0.347
	W2	1.041	2.087	0.607	1.041	2.087	0.607
	W3	1.217	1.400	0.518	1.217	1.400	0.518
	W4	1.872	1.815	0.207	1.872	1.815	0.207
	W5	1.459	1.438	0.334	1.459	1.438	0.334
	W6	1.103	1.746	0.383	1.103	1.746	0.383
	Average	**1.301**	**1.808**	**0.399**	**1.301**	**1.808**	**0.399**
TD3-M2	W1	0.989	1.110	0.542	0.989	1.110	0.542
	W2	1.477	0.725	0.486	1.477	0.725	0.486
	W3	1.507	0.837	0.917	1.507	0.837	0.917
	W4	1.336	1.084	0.448	1.336	1.084	0.448
	W5	1.029	0.903	0.886	1.029	0.903	0.886
	W6	1.048	0.952	0.517	1.048	0.952	0.517
	Average	**1.231**	**0.935**	**0.633**	**1.231**	**0.935**	**0.633**

D.4 Harmonic ratio CP4



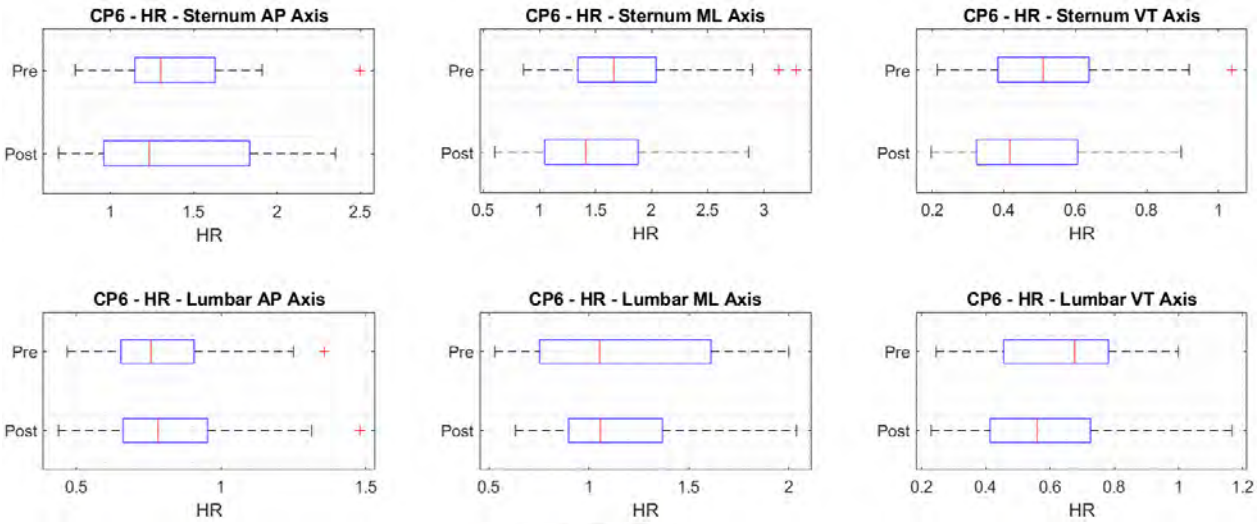
Dataset	Walk Nr.	Sternum HR			Lumbar HR		
		AP	ML	VT	AP	ML	VT
CP4-Pre	W1	0.758	1.129	0.633	0.758	1.129	0.633
	W2	0.813	1.130	0.554	0.813	1.130	0.554
	W3	1.037	1.117	0.603	1.037	1.117	0.603
	W4	0.830	1.118	0.506	0.830	1.118	0.506
	W5	0.946	1.014	0.727	0.946	1.014	0.727
	W6	0.832	1.028	0.823	0.832	1.028	0.823
	Average	**0.869**	**1.090**	**0.641**	**0.869**	**1.090**	**0.641**
CP4-Post	W1	0.825	0.981	0.548	0.825	0.981	0.548
	W2	0.672	0.870	0.699	0.672	0.870	0.699
	W3	0.847	0.950	0.659	0.847	0.950	0.659
	W4	0.909	0.845	0.631	0.909	0.845	0.631
	W5	0.756	1.013	0.625	0.756	1.013	0.625
	W6	0.911	0.665	0.611	0.911	0.665	0.611
	Average	**0.820**	**0.887**	**0.629**	**0.820**	**0.887**	**0.629**
TD4-M1	W1	0.936	0.944	0.571	0.936	0.944	0.571
	W2	1.270	1.059	0.433	1.270	1.059	0.433
	W3	1.489	0.847	0.574	1.489	0.847	0.574
	W4	1.230	1.068	1.122	1.230	1.068	1.122
	W5	0.995	1.046	0.728	0.995	1.046	0.728
	W6	1.409	0.742	0.368	1.409	0.742	0.368
	Average	**1.221**	**0.951**	**0.633**	**1.221**	**0.951**	**0.633**
TD4-M2	W1	0.984	1.451	0.871	0.984	1.451	0.871
	W2	1.436	1.018	0.520	1.436	1.018	0.520
	W3	1.152	1.666	0.713	1.152	1.666	0.713
	W4	1.474	0.899	0.708	1.474	0.899	0.708
	W5	1.602	1.246	0.513	1.602	1.246	0.513
	W6	2.309	1.414	0.296	2.309	1.414	0.296
	Average	**1.493**	**1.282**	**0.604**	**1.493**	**1.282**	**0.604**

D.5 Harmonic ratio CP5



Dataset	Walk Nr.	Sternum HR			Lumbar HR		
		AP	ML	VT	AP	ML	VT
CP5-Pre	W1	1.138	2.359	0.641	1.138	2.359	0.641
	W2	0.975	2.377	0.608	0.975	2.377	0.608
	W3	0.913	1.859	0.577	0.913	1.859	0.577
	W4	1.193	1.768	0.743	1.193	1.768	0.743
	W5	1.165	1.521	0.585	1.165	1.521	0.585
	W6	0.790	1.235	0.682	0.790	1.235	0.682
	Average	**1.029**	**1.853**	**0.639**	**1.029**	**1.853**	**0.639**
CP5-Post	W1	1.029	1.839	0.546	1.029	1.839	0.546
	W2	0.873	1.880	0.714	0.873	1.880	0.714
	W3	1.318	1.357	0.695	1.318	1.357	0.695
	W4	0.869	1.714	0.505	0.869	1.714	0.505
	W5	0.936	1.473	0.552	0.936	1.473	0.552
	W6	1.025	1.521	0.749	1.025	1.521	0.749
	Average	**1.008**	**1.631**	**0.627**	**1.008**	**1.631**	**0.627**
TD5-M1	W1	0.768	1.905	0.830	0.768	1.905	0.830
	W2	0.523	2.221	0.370	0.523	2.221	0.370
	W3	0.422	1.575	1.124	0.422	1.575	1.124
	W4	0.689	2.600	0.574	0.689	2.600	0.574
	W5	0.625	1.626	0.627	0.625	1.626	0.627
	W6	0.725	1.878	0.617	0.725	1.878	0.617
	Average	**0.625**	**1.968**	**0.690**	**0.625**	**1.968**	**0.690**
TD5-M2	W1	0.861	1.936	0.478	0.861	1.936	0.478
	W2	1.305	1.452	0.425	1.305	1.452	0.425
	W3	0.734	1.860	0.606	0.734	1.860	0.606
	W4	0.772	1.255	0.782	0.772	1.255	0.782
	W5	1.088	2.214	0.782	1.088	2.214	0.782
	W6	0.919	2.206	0.906	0.919	2.206	0.906
	Average	**0.947**	**1.821**	**0.663**	**0.947**	**1.821**	**0.663**

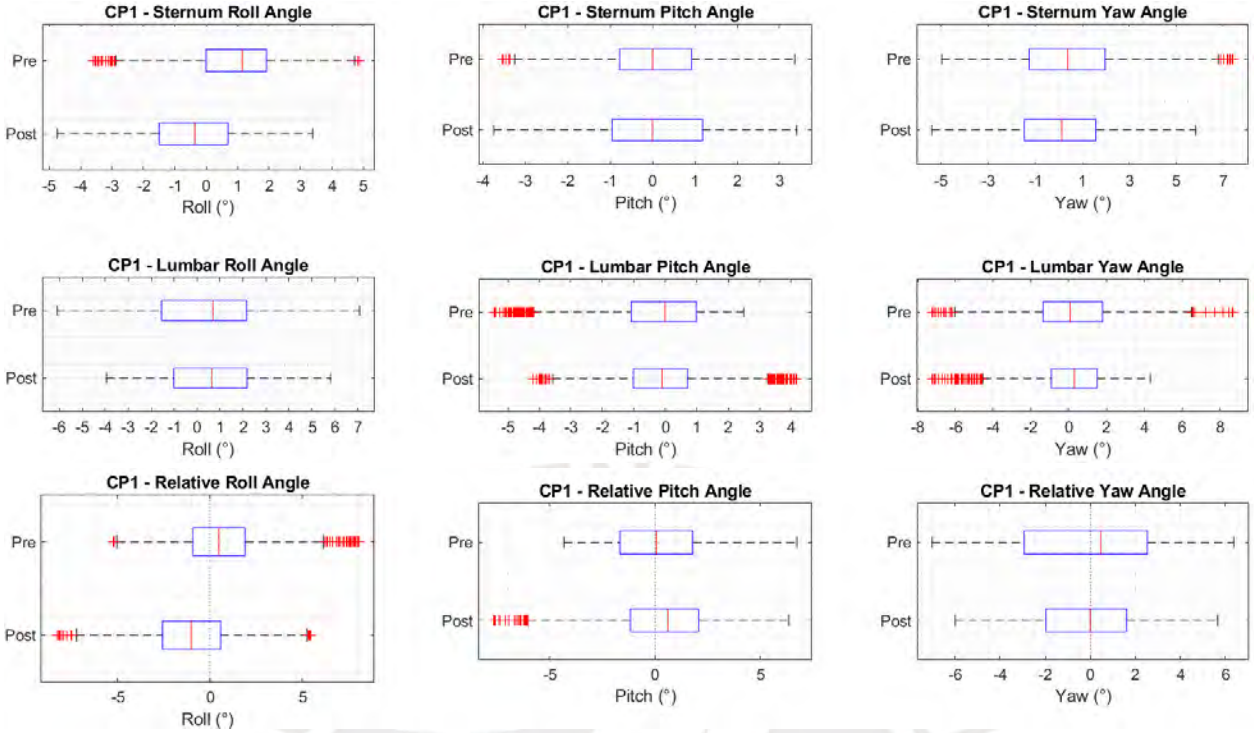
D.6 Harmonic ratio CP6



Dataset	Walk Nr.	Sternum HR			Lumbar HR		
		AP	ML	VT	AP	ML	VT
CP6-Pre	W1	1.243	1.439	0.604	1.243	1.439	0.604
	W2	1.477	1.946	0.541	1.477	1.946	0.541
	W3	1.465	1.941	0.476	1.465	1.941	0.476
	W4	1.323	1.810	0.532	1.323	1.810	0.532
	W5	1.366	1.490	0.661	1.366	1.490	0.661
	W6	1.471	1.942	0.426	1.471	1.942	0.426
	Average	**1.391**	**1.761**	**0.540**	**1.391**	**1.761**	**0.540**
CP6-Post	W1	0.935	1.015	0.860	0.935	1.015	0.860
	W2	1.417	1.522	0.368	1.417	1.522	0.368
	W3	1.560	1.285	0.497	1.560	1.285	0.497
	W4	1.481	1.356	0.513	1.481	1.356	0.513
	W5	1.040	1.693	0.403	1.040	1.693	0.403
	W6	1.602	1.951	0.526	1.602	1.951	0.526
	Average	**1.339**	**1.470**	**0.528**	**1.339**	**1.470**	**0.528**
TD6-M1	W1	1.970	2.280	0.287	1.970	2.280	0.287
	W2	1.943	2.212	0.269	1.943	2.212	0.269
	W3	2.516	3.171	0.311	2.516	3.171	0.311
	W4	1.820	2.180	0.271	1.820	2.180	0.271
	W5	1.275	2.291	0.460	1.275	2.291	0.460
	W6	1.783	1.982	0.264	1.783	1.982	0.264
	Average	**1.884**	**2.353**	**0.310**	**1.884**	**2.353**	**0.310**
TD6-M2	W1	1.082	1.621	0.741	1.082	1.621	0.741
	W2	1.443	1.843	0.247	1.443	1.843	0.247
	W3	1.106	1.652	0.606	1.106	1.652	0.606
	W4	1.159	1.613	0.811	1.159	1.613	0.811
	W5	0.875	2.070	0.254	0.875	2.070	0.254
	W6	1.143	1.603	0.581	1.143	1.603	0.581
	Average	**1.135**	**1.734**	**0.540**	**1.135**	**1.734**	**0.540**

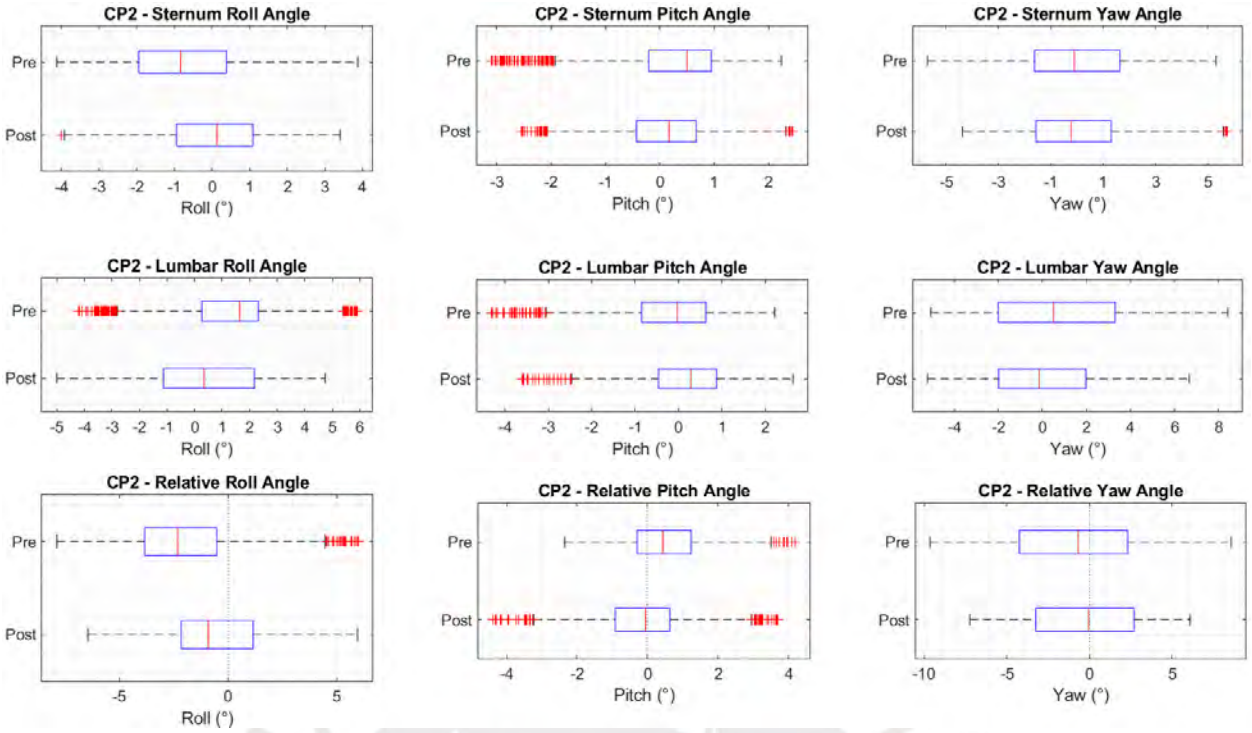
E. Absolute and Relative Angles

E.1 Absolute and Relative angles CP1



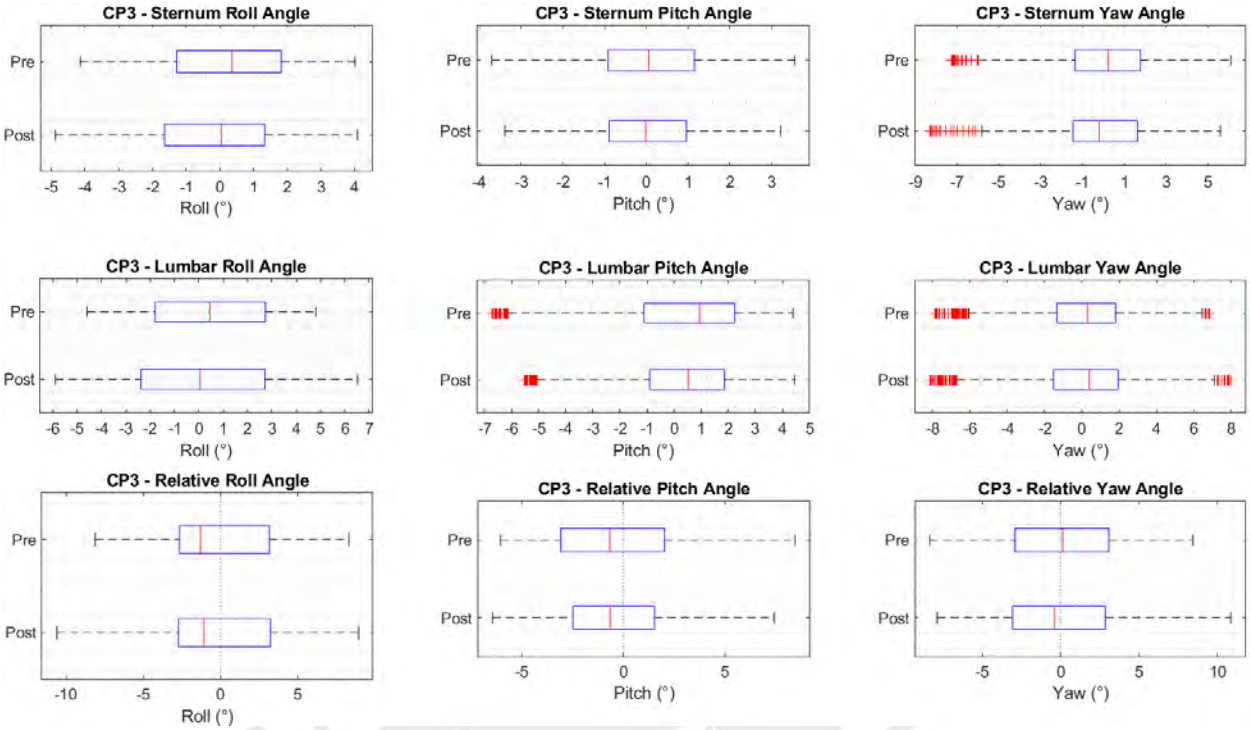
Dataset	Walk Nr.	Sternum Angles (°)			Lumbar Angles (°)			Relative Angles (°)		
		Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw
CP1-Pre	W1	0.646	-0.098	-0.322	0.646	-0.098	-0.322	-0.123	0.198	-0.603
	W2	0.550	0.287	0.369	0.550	0.287	0.369	0.866	0.678	0.454
	W3	1.245	0.124	0.656	1.245	0.124	0.656	1.005	0.600	0.325
	W4	0.978	0.046	0.633	0.978	0.046	0.633	0.557	0.342	-0.143
	W5	1.171	0.090	0.417	1.171	0.090	0.417	0.557	0.342	-0.143
	W6	1.222	-0.026	0.548	1.222	-0.026	0.548	0.687	0.244	0.082
	Average	**0.969**	**0.070**	**0.384**	**0.969**	**0.070**	**0.384**	**0.553**	**0.380**	**-0.055**
CP1-Post	W1	-0.991	-0.045	-0.296	-0.991	-0.045	-0.296	-1.606	0.097	-0.392
	W2	-0.390	0.073	-0.046	-0.390	0.073	-0.046	-0.980	0.268	-0.084
	W3	-0.158	0.082	-0.189	-0.158	0.082	-0.189	-0.975	0.171	-0.529
	W4	-0.088	0.054	0.279	-0.088	0.054	0.279	-0.403	0.376	-0.100
	W5	0.011	0.066	0.119	0.011	0.066	0.119	-0.403	0.376	-0.100
	W6	-0.918	0.127	0.949	-0.918	0.127	0.949	-1.582	0.360	0.401
	Average	**-0.422**	**0.060**	**0.136**	**-0.422**	**0.060**	**0.136**	**-1.013**	**0.266**	**-0.146**
TD1-M1	W1	-0.080	0.211	-0.036	-0.080	0.211	-0.036	1.465	0.029	0.138
	W2	0.649	0.152	0.335	0.649	0.152	0.335	1.936	-0.026	0.290
	W3	0.492	0.135	0.069	0.492	0.135	0.069	1.473	0.202	0.224
	W4	-0.107	0.067	0.129	-0.107	0.067	0.129	0.675	0.159	0.199
	W5	-0.423	0.120	0.090	-0.423	0.120	0.090	0.675	0.159	0.199
	W6	-0.098	0.176	0.420	-0.098	0.176	0.420	0.974	0.293	0.219
	Average	**0.072**	**0.144**	**0.168**	**0.072**	**0.144**	**0.168**	**1.165**	**0.140**	**0.209**
TD1-M2	W1	-0.338	0.200	-0.012	-0.338	0.200	-0.012	-0.125	0.234	0.362
	W2	-0.224	-0.020	0.131	-0.224	-0.020	0.131	0.206	0.192	0.360
	W3	0.017	0.109	-0.345	0.017	0.109	-0.345	0.572	0.081	0.155
	W4	-0.136	0.119	0.029	-0.136	0.119	0.029	0.406	0.039	-0.352
	W5	0.187	0.113	-0.533	0.187	0.113	-0.533	0.406	0.039	-0.352
	W6	0.163	0.062	0.498	0.163	0.062	0.498	0.363	0.441	0.443
	Average	**-0.055**	**0.097**	**-0.039**	**-0.055**	**0.097**	**-0.039**	**0.288**	**0.177**	**0.107**

E.2 Absolute and Relative angles CP2



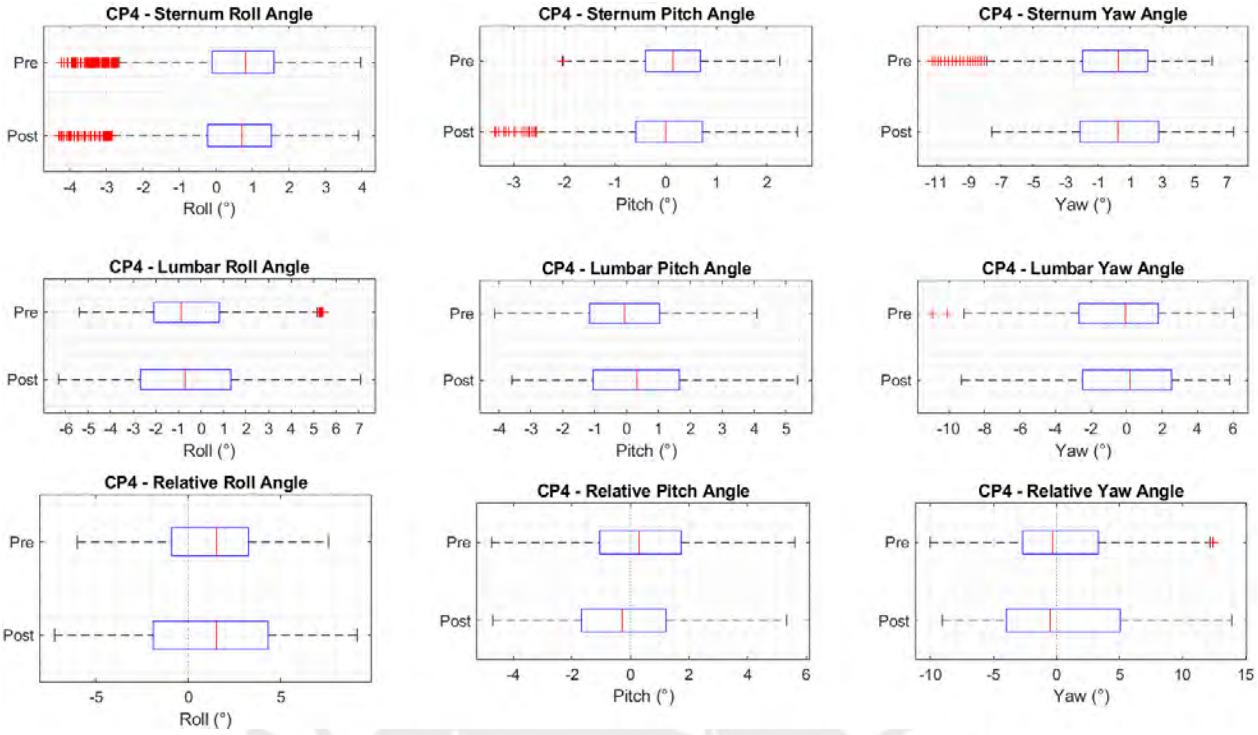
Dataset	Walk Nr.	Sternum Angles (°)			Lumbar Angles (°)			Relative Angles (°)		
		Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw
CP2-Pre	W1	-0.634	0.481	-0.284	-0.634	0.481	-0.284	-1.983	0.650	-1.136
	W2	-0.970	0.117	0.269	-0.970	0.117	0.269	-2.217	0.230	-0.273
	W3	-1.326	0.368	-0.719	-1.326	0.368	-0.719	-2.592	0.466	-1.609
	W4	-0.685	0.191	0.585	-0.685	0.191	0.585	-1.293	0.357	-1.246
	W5	-0.014	0.175	-0.456	-0.014	0.175	-0.456	-1.293	0.357	-1.246
	W6	-0.986	0.430	0.217	-0.986	0.430	0.217	-2.413	0.730	-0.566
	Average	**-0.769**	**0.294**	**-0.065**	**-0.769**	**0.294**	**-0.065**	**-1.979**	**0.472**	**-0.995**
CP2-Post	W1	0.122	0.165	-0.319	0.122	0.165	-0.319	-0.618	0.008	-0.184
	W2	0.558	-0.016	0.183	0.558	-0.016	0.183	0.283	-0.266	-0.075
	W3	0.194	0.061	-0.049	0.194	0.061	-0.049	-0.529	-0.233	-0.533
	W4	-0.431	0.206	0.167	-0.431	0.206	0.167	-0.597	0.108	-0.345
	W5	-0.133	0.328	-0.234	-0.133	0.328	-0.234	-0.597	0.108	-0.345
	W6	-0.125	0.024	0.030	-0.125	0.024	0.030	-0.481	-0.322	-0.213
	Average	**0.031**	**0.128**	**-0.037**	**0.031**	**0.128**	**-0.037**	**-0.406**	**-0.113**	**-0.283**
TD2-M1	W1	0.637	0.015	-0.056	0.637	0.015	-0.056	-0.333	0.360	-0.181
	W2	0.048	0.028	0.022	0.048	0.028	0.022	-0.673	0.164	0.241
	W3	0.301	-0.011	-0.257	0.301	-0.011	-0.257	-0.552	0.392	-0.695
	W4	-0.139	0.033	0.009	-0.139	0.033	0.009	-1.099	-0.057	-0.475
	W5	0.033	-0.307	-0.125	0.033	-0.307	-0.125	-1.099	-0.057	-0.475
	W6	0.313	-0.020	-0.082	0.313	-0.020	-0.082	-0.801	0.310	-0.210
	Average	**0.199**	**-0.044**	**-0.081**	**0.199**	**-0.044**	**-0.081**	**-0.724**	**0.205**	**-0.299**
TD2-M2	W1	0.102	-0.166	-0.446	0.102	-0.166	-0.446	-0.736	0.314	-0.816
	W2	-0.289	-0.389	0.496	-0.289	-0.389	0.496	-1.026	0.294	0.242
	W3	0.038	-0.152	-0.024	0.038	-0.152	-0.024	-0.688	0.588	-0.285
	W4	-0.241	-0.275	0.467	-0.241	-0.275	0.467	-0.838	0.582	-0.603
	W5	-0.015	-0.208	0.057	-0.015	-0.208	0.057	-0.838	0.582	-0.603
	W6	-0.884	-0.114	0.087	-0.884	-0.114	0.087	-2.054	0.465	-0.014
	Average	**-0.215**	**-0.218**	**0.106**	**-0.215**	**-0.218**	**0.106**	**-1.007**	**0.482**	**-0.352**

E.3 Absolute and Relative angles CP3



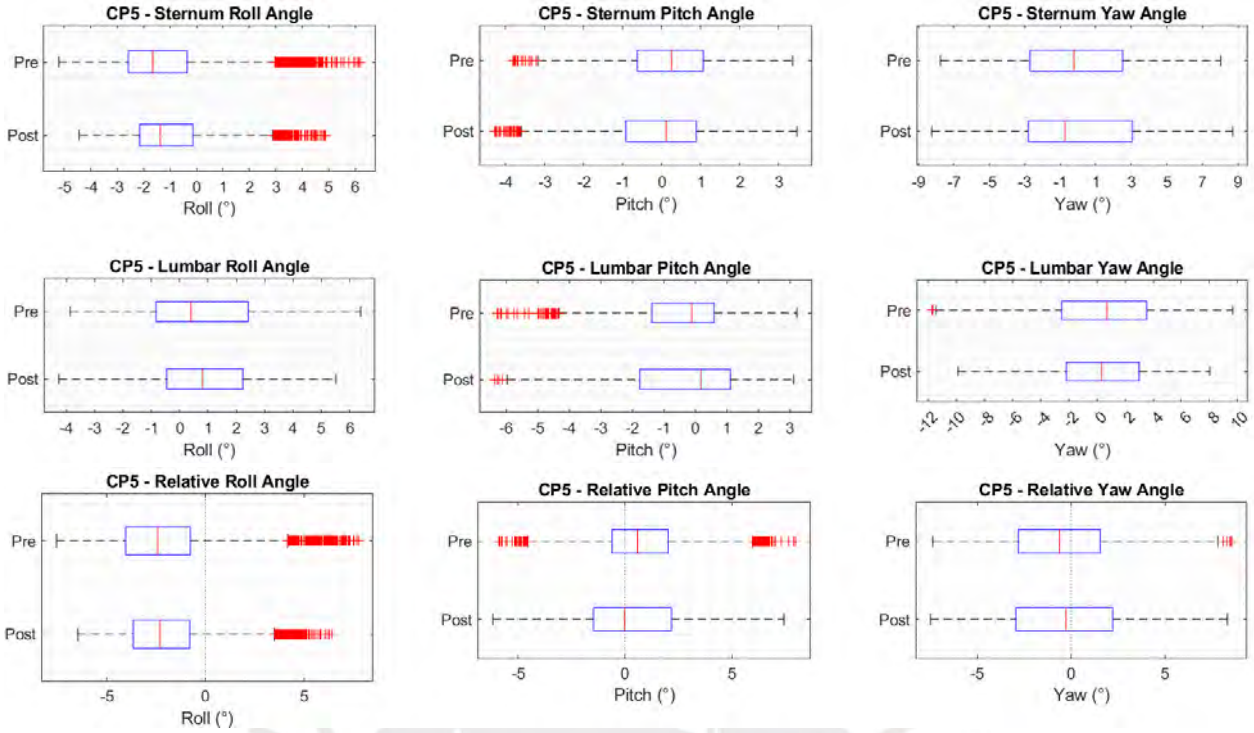
Dataset	Walk Nr.	Sternum Angles (°)			Lumbar Angles (°)			Relative Angles (°)		
		Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw
CP3-Pre	W1	0.196	0.105	0.205	0.196	0.105	0.205	-0.120	-0.181	0.142
	W2	0.090	0.030	0.281	0.090	0.030	0.281	-0.567	-0.234	0.336
	W3	0.414	0.177	0.361	0.414	0.177	0.361	-0.052	-0.118	0.273
	W4	0.300	0.076	0.437	0.300	0.076	0.437	0.028	-0.102	0.205
	W5	0.528	0.199	0.254	0.528	0.199	0.254	0.028	-0.102	0.205
	W6	0.306	0.092	0.022	0.306	0.092	0.022	-0.147	-0.170	-0.096
	Average	**0.306**	**0.113**	**0.260**	**0.306**	**0.113**	**0.260**	**-0.148**	**-0.155**	**0.179**
CP3-Post	W1	-0.270	0.088	-0.085	-0.270	0.088	-0.085	-0.296	-0.231	-0.006
	W2	-0.177	-0.030	0.043	-0.177	-0.030	0.043	-0.533	-0.344	0.014
	W3	0.055	0.006	0.256	0.055	0.006	0.256	-0.269	-0.333	0.415
	W4	-0.032	0.067	0.111	-0.032	0.067	0.111	0.031	-0.472	-0.027
	W5	0.235	-0.041	0.031	0.235	-0.041	0.031	0.031	-0.472	-0.027
	W6	-0.449	-0.026	-0.005	-0.449	-0.026	-0.005	-0.675	-0.405	-0.161
	Average	**-0.106**	**0.011**	**0.059**	**-0.106**	**0.011**	**0.059**	**-0.286**	**-0.363**	**0.043**
TD3-M1	W1	-1.205	0.274	-0.345	-1.205	0.274	-0.345	-1.963	0.884	-1.238
	W2	-1.571	0.260	0.359	-1.571	0.260	0.359	-2.313	0.702	0.092
	W3	-0.486	0.134	-0.400	-0.486	0.134	-0.400	-1.351	0.761	-0.846
	W4	-0.330	-0.211	0.247	-0.330	-0.211	0.247	-1.619	0.758	-1.490
	W5	-0.424	0.080	-0.591	-0.424	0.080	-0.591	-1.619	0.758	-1.490
	W6	-0.174	-0.016	-0.007	-0.174	-0.016	-0.007	-1.335	0.529	0.041
	Average	**-0.698**	**0.087**	**-0.123**	**-0.698**	**0.087**	**-0.123**	**-1.685**	**0.740**	**-0.870**
TD3-M2	W1	-0.174	0.151	0.067	-0.174	0.151	0.067	-1.760	0.892	-0.881
	W2	0.098	-0.054	0.208	0.098	-0.054	0.208	-1.442	0.673	0.406
	W3	0.628	-0.219	-0.063	0.628	-0.219	-0.063	-0.937	0.499	-0.602
	W4	0.191	-0.082	0.465	0.191	-0.082	0.465	-1.253	0.512	-0.742
	W5	0.387	-0.159	-0.076	0.387	-0.159	-0.076	-1.253	0.512	-0.742
	W6	0.779	-0.147	0.623	0.779	-0.147	0.623	-0.774	0.645	0.419
	Average	**0.318**	**-0.085**	**0.204**	**0.318**	**-0.085**	**0.204**	**-1.256**	**0.638**	**-0.371**

E.4 Absolute and Relative angles CP4



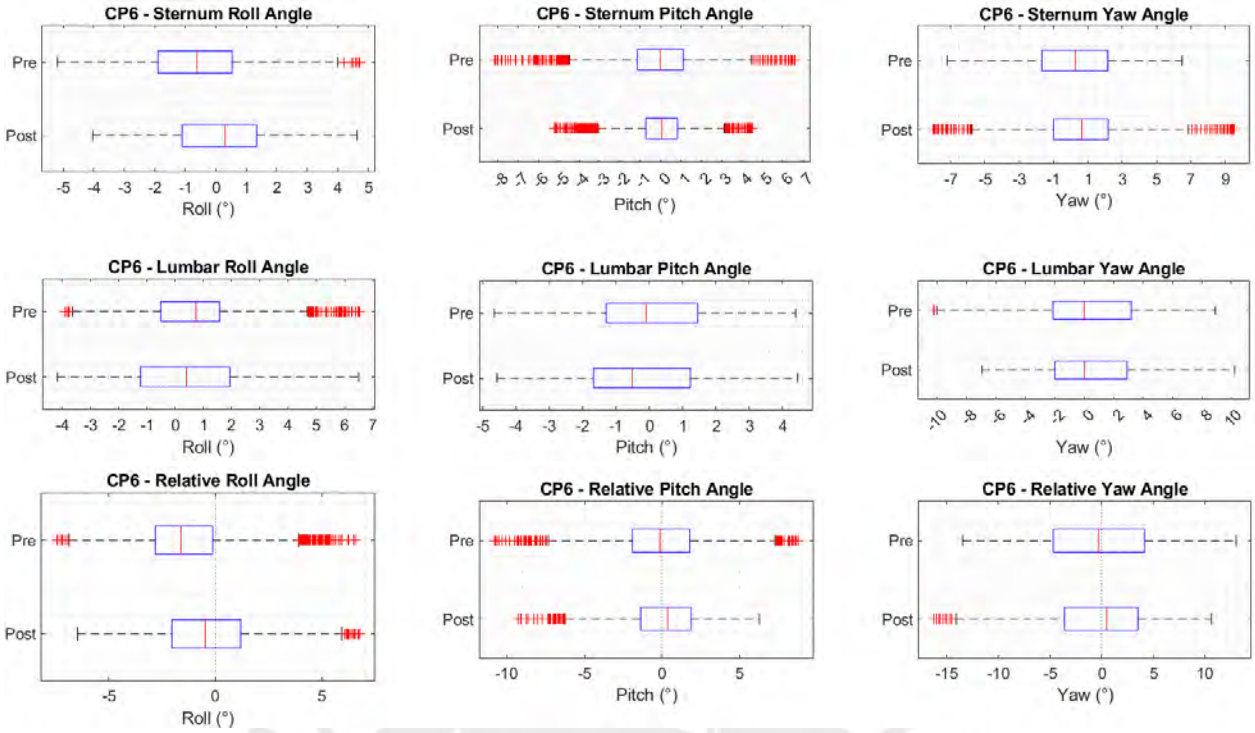
Dataset	Walk Nr.	Sternum Angles (°)			Lumbar Angles (°)			Relative Angles (°)		
		Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw
CP4-Pre	W1	1.071	0.206	0.171	1.071	0.206	0.171	1.594	0.337	0.966
	W2	0.651	0.122	-0.345	0.651	0.122	-0.345	1.041	0.207	0.106
	W3	0.627	0.078	-0.090	0.627	0.078	-0.090	1.269	0.171	0.776
	W4	0.587	0.082	0.073	0.587	0.082	0.073	1.205	0.243	0.779
	W5	0.634	0.096	0.180	0.634	0.096	0.180	1.205	0.243	0.779
	W6	0.323	0.222	0.151	0.323	0.222	0.151	0.880	0.312	0.131
	Average	**0.649**	**0.134**	**0.023**	**0.649**	**0.134**	**0.023**	**1.188**	**0.250**	**0.570**
CP4-Post	W1	0.673	0.051	0.519	0.673	0.051	0.519	1.255	-0.147	0.620
	W2	0.710	0.134	0.591	0.710	0.134	0.591	1.541	-0.188	0.676
	W3	0.708	0.113	0.211	0.708	0.113	0.211	1.361	-0.086	0.648
	W4	0.504	0.044	0.179	0.504	0.044	0.179	1.416	-0.203	0.682
	W5	0.742	0.040	0.171	0.742	0.040	0.171	1.416	-0.203	0.682
	W6	0.284	0.018	0.011	0.284	0.018	0.011	0.913	-0.074	0.087
	Average	**0.604**	**0.067**	**0.280**	**0.604**	**0.067**	**0.280**	**1.308**	**-0.148**	**0.556**
TD4-M1	W1	0.169	0.063	-0.111	0.169	0.063	-0.111	-0.423	0.163	-0.367
	W2	-0.202	-0.074	0.350	-0.202	-0.074	0.350	-0.652	-0.004	0.201
	W3	0.193	0.009	-0.029	0.193	0.009	-0.029	-0.461	0.218	-0.437
	W4	-0.004	-0.068	0.206	-0.004	-0.068	0.206	-0.015	0.157	0.334
	W5	0.506	0.026	0.286	0.506	0.026	0.286	-0.015	0.157	0.334
	W6	0.252	-0.035	0.456	0.252	-0.035	0.456	-0.439	0.058	0.228
	Average	**0.152**	**-0.013**	**0.193**	**0.152**	**-0.013**	**0.193**	**-0.330**	**0.126**	**0.041**
TD4-M2	W1	-0.031	0.064	-0.104	-0.031	0.064	-0.104	-0.274	0.274	-0.365
	W2	0.512	0.054	0.213	0.512	0.054	0.213	0.000	0.215	0.053
	W3	0.516	-0.015	0.333	0.516	-0.015	0.333	0.107	0.059	0.439
	W4	0.797	-0.061	0.384	0.797	-0.061	0.384	0.497	-0.236	0.342
	W5	0.939	-0.139	0.479	0.939	-0.139	0.479	0.497	-0.236	0.342
	W6	1.172	-0.083	0.310	1.172	-0.083	0.310	0.894	0.086	0.308
	Average	**0.651**	**-0.030**	**0.269**	**0.651**	**-0.030**	**0.269**	**0.296**	**0.017**	**0.194**

E.5 Absolute and Relative angles CP5



Dataset	Walk Nr.	Sternum Angles (°)			Lumbar Angles (°)			Relative Angles (°)		
		Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw
CP5-Pre	W1	-1.584	0.116	-0.362	-1.584	0.116	-0.362	-2.556	0.629	-1.033
	W2	-1.441	0.276	0.137	-1.441	0.276	0.137	-2.312	0.773	-0.175
	W3	-1.409	0.306	-0.595	-1.409	0.306	-0.595	-2.295	0.783	-0.894
	W4	-1.245	0.085	0.231	-1.245	0.085	0.231	-1.738	0.588	-0.748
	W5	-1.212	0.192	-0.482	-1.212	0.192	-0.482	-1.738	0.588	-0.748
	W6	-0.758	0.279	0.300	-0.758	0.279	0.300	-1.423	0.677	0.162
	Average	**-1.275**	**0.209**	**-0.128**	**-1.275**	**0.209**	**-0.128**	**-2.080**	**0.682**	**-0.576**
CP5-Post	W1	-1.146	-0.110	-0.068	-1.146	-0.110	-0.068	-2.079	0.297	-0.519
	W2	-1.342	-0.069	0.242	-1.342	-0.069	0.242	-2.188	0.233	0.203
	W3	-0.664	-0.061	-0.157	-0.664	-0.061	-0.157	-1.567	0.384	-0.609
	W4	-0.824	0.139	0.162	-0.824	0.139	0.162	-1.910	0.163	-0.454
	W5	-0.954	-0.202	-0.084	-0.954	-0.202	-0.084	-1.910	0.163	-0.454
	W6	-1.027	0.093	0.158	-1.027	0.093	0.158	-1.946	0.426	0.092
	Average	**-0.993**	**-0.035**	**0.042**	**-0.993**	**-0.035**	**0.042**	**-1.953**	**0.275**	**-0.274**
TD5-M1	W1	0.781	-0.713	0.695	0.781	-0.713	0.695	0.166	-0.803	0.498
	W2	1.574	-0.870	1.026	1.574	-0.870	1.026	1.153	-0.895	0.525
	W3	1.463	-0.780	0.810	1.463	-0.780	0.810	1.051	-0.673	0.261
	W4	1.916	-0.712	0.638	1.916	-0.712	0.638	0.938	-0.739	0.368
	W5	1.350	-0.889	0.141	1.350	-0.889	0.141	0.938	-0.739	0.368
	W6	1.750	-0.572	0.254	1.750	-0.572	0.254	1.153	-0.628	-0.346
	Average	**1.472**	**-0.756**	**0.594**	**1.472**	**-0.756**	**0.594**	**0.844**	**-0.748**	**0.282**
TD5-M2	W1	1.727	-0.627	0.847	1.727	-0.627	0.847	1.127	-0.488	0.704
	W2	1.169	-0.127	0.117	1.169	-0.127	0.117	0.355	0.212	-0.261
	W3	1.588	-0.717	1.008	1.588	-0.717	1.008	0.872	-0.603	0.508
	W4	1.611	-0.601	0.078	1.611	-0.601	0.078	0.816	-0.322	0.372
	W5	1.698	-0.424	0.824	1.698	-0.424	0.824	0.816	-0.322	0.372
	W6	1.918	-0.555	0.324	1.918	-0.555	0.324	1.262	-0.484	0.043
	Average	**1.619**	**-0.509**	**0.533**	**1.619**	**-0.509**	**0.533**	**0.897**	**-0.354**	**0.310**

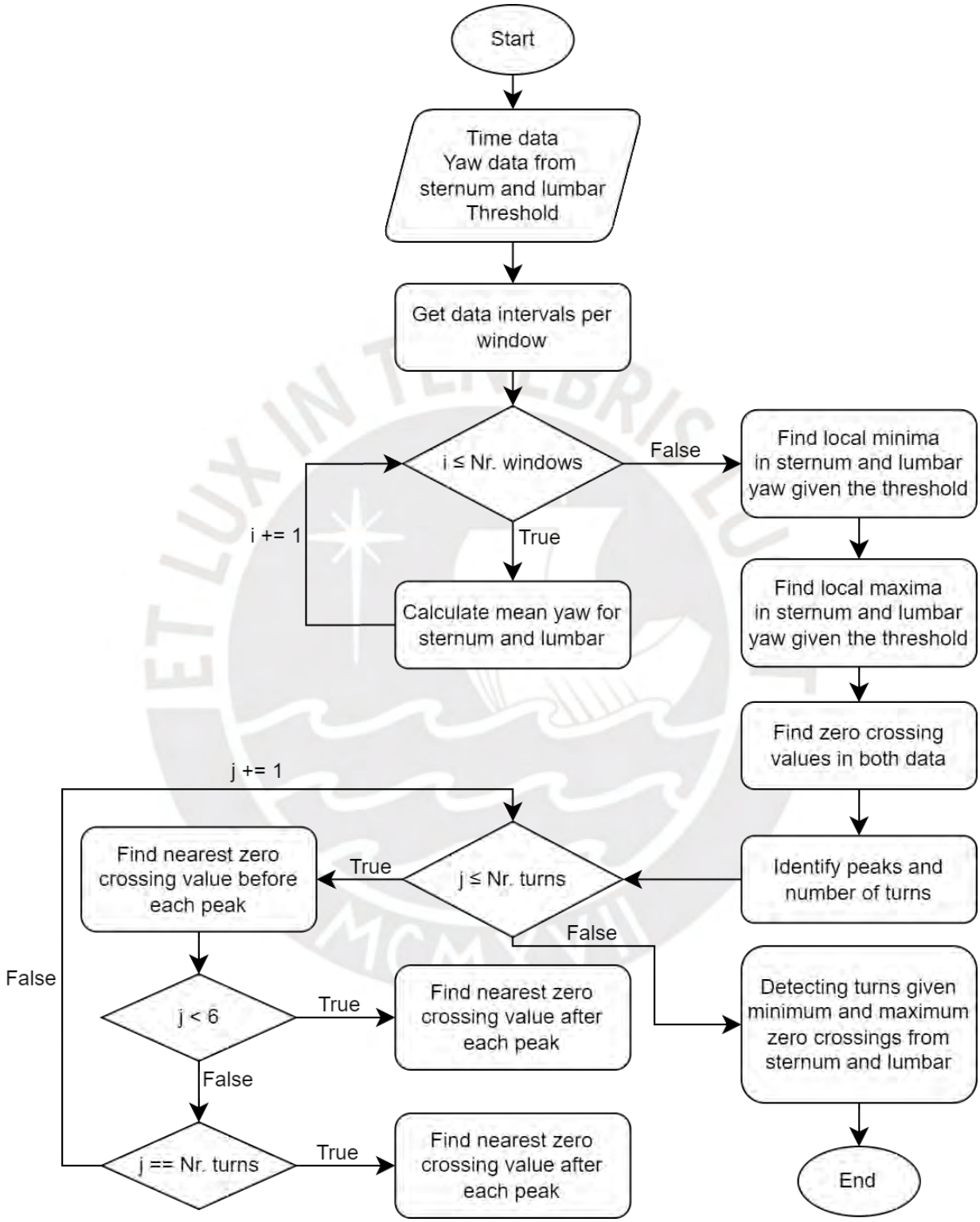
E.6 Absolute and Relative angles CP6



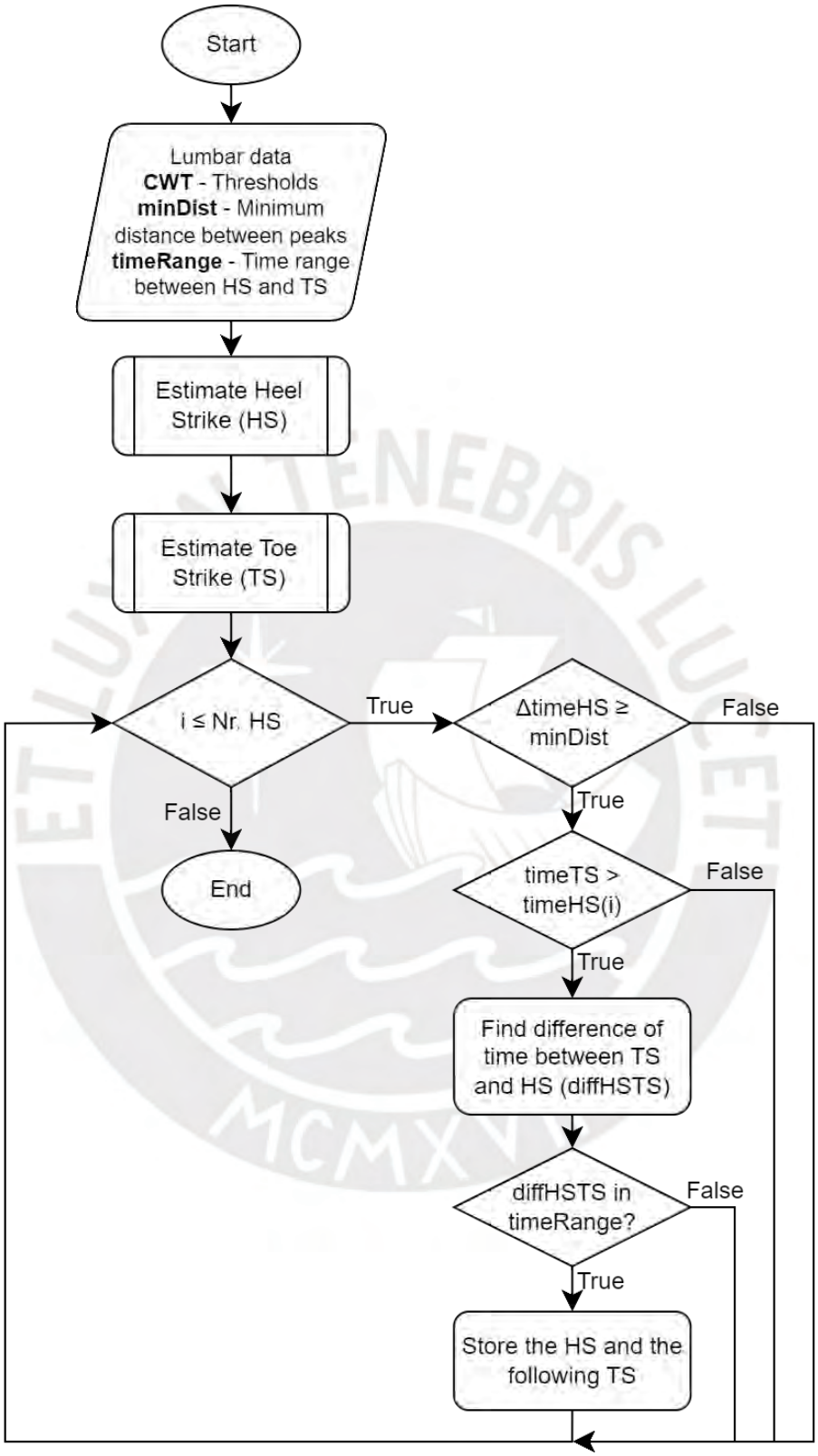
Dataset	Walk Nr.	Sternum Angles (°)			Lumbar Angles (°)			Relative Angles (°)		
		Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw
CP6-Pre	W1	-0.390	0.020	-0.120	-0.390	0.020	-0.120	-1.319	0.026	-0.384
	W2	-0.476	-0.137	0.579	-0.476	-0.137	0.579	-1.156	-0.090	-0.008
	W3	-0.757	-0.155	-0.378	-0.757	-0.155	-0.378	-1.410	-0.188	-0.892
	W4	-0.961	0.239	0.046	-0.961	0.239	0.046	-1.108	-0.075	-0.367
	W5	-0.592	-0.020	0.140	-0.592	-0.020	0.140	-1.108	-0.075	-0.367
	W6	-0.761	-0.292	0.328	-0.761	-0.292	0.328	-1.605	-0.483	-0.134
	Average	**-0.656**	**-0.058**	**0.099**	**-0.656**	**-0.058**	**0.099**	**-1.282**	**-0.148**	**-0.332**
CP6-Post	W1	0.298	0.214	0.084	0.298	0.214	0.084	-0.198	0.428	-0.520
	W2	0.053	-0.094	0.372	0.053	-0.094	0.372	-0.616	0.183	-0.053
	W3	0.151	0.001	0.847	0.151	0.001	0.847	-0.351	0.156	0.174
	W4	0.155	0.197	0.817	0.155	0.197	0.817	0.025	0.051	-0.082
	W5	0.557	-0.183	0.464	0.557	-0.183	0.464	0.025	0.051	-0.082
	W6	-0.235	-0.161	0.940	-0.235	-0.161	0.940	-0.557	-0.038	0.295
	Average	**0.163**	**-0.004**	**0.587**	**0.163**	**-0.004**	**0.587**	**-0.281**	**0.134**	**-0.043**
TD6-M1	W1	0.938	-0.749	-0.339	0.938	-0.749	-0.339	0.642	-0.791	-0.822
	W2	1.089	-0.058	-0.388	1.089	-0.058	-0.388	0.634	0.277	-0.359
	W3	1.257	0.099	-0.447	1.257	0.099	-0.447	1.115	0.190	-0.161
	W4	1.055	-0.251	-0.188	1.055	-0.251	-0.188	0.388	0.389	-0.756
	W5	0.675	0.139	-0.387	0.675	0.139	-0.387	0.388	0.389	-0.756
	W6	0.978	-0.499	0.143	0.978	-0.499	0.143	-0.034	-0.877	-0.359
	Average	**0.999**	**-0.220**	**-0.268**	**0.999**	**-0.220**	**-0.268**	**0.550**	**-0.067**	**-0.554**
TD6-M2	W1	0.152	0.291	-0.240	0.152	0.291	-0.240	0.094	0.155	0.150
	W2	0.399	-0.404	-0.432	0.399	-0.404	-0.432	0.727	-1.017	-0.491
	W3	0.869	-0.520	0.516	0.869	-0.520	0.516	0.918	-0.420	0.366
	W4	1.011	-0.410	-0.098	1.011	-0.410	-0.098	0.857	-0.389	-0.130
	W5	0.664	-0.422	-0.590	0.664	-0.422	-0.590	0.857	-0.389	-0.130
	W6	1.683	-0.653	-0.260	1.683	-0.653	-0.260	1.413	-0.645	-0.026
	Average	**0.796**	**-0.353**	**-0.184**	**0.796**	**-0.353**	**-0.184**	**0.739**	**-0.369**	**-0.001**

F. Flowcharts

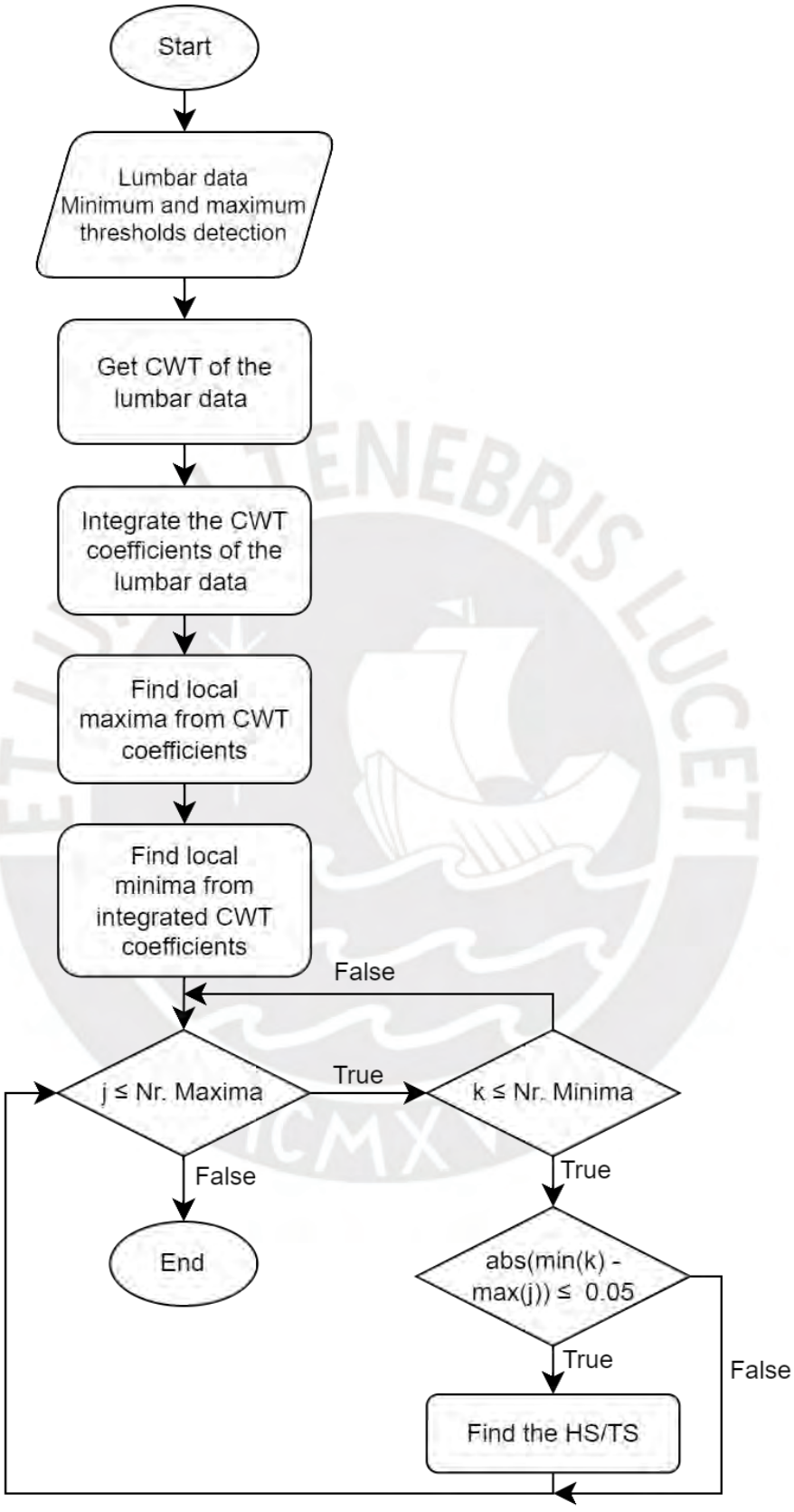
F.1 Automatic walks and turns detection



F.2 Partially automatic gait cycles detection



F.3 Estimation of HS or TS



G. Code

G.1 Main code

```
%% Masther thesis: Main code
% Thesis name:
% A method of estimation of relevant trunk control parameters
% for cerebral palsy rehabilitation in children during independent walking
% based on inertial measurement unit data
% Author: Maria Alejandra Guzman Alfaro
% Last updated: 01.11.2024
CP1 = main(1);
CP2 = main(2);
CP3 = main(3);
CP4 = main(4);
CP5 = main(5);
CP6 = main(6);

function results = main(patient)
%% I. Data collection
% 1. Calling libraries of functions
preproc = libPreProcessing; proc = libProcessing; param = libParameters;
% 2. Loading arguments/variables necessary for each data file processing
fs = 128; % Sampling frequency (Hz)
args = preproc.loadArguments(patient);
% 3. Reading csv files
[pre, post, td1, td2] = preproc.loadPatientData(patient);

%% II. Raw data management
%% 1. Gathering and preprocessing data
pre.raw = preproc.collectDataIMUs(pre.csv, pre.Nr);
post.raw = preproc.collectDataIMUs(post.csv, post.Nr);
td1.raw = preproc.collectDataIMUs(td1.csv, td1.Nr);
td2.raw = preproc.collectDataIMUs(td2.csv, td2.Nr);
%% 2. Initial angles calculation
[pre.vars] = preproc.getRPYAngles(0.2, pre.raw);
[post.vars] = preproc.getRPYAngles(0.2, post.raw);
[td1.vars] = preproc.getRPYAngles(0.2, td1.raw);
[td2.vars] = preproc.getRPYAngles(0.2, td2.raw);
%% 3. Coordinate transformation (Initial frame to global frame)
pre.glob = preproc.getGlobalFrameData(pre.vars, pre.raw);
post.glob = preproc.getGlobalFrameData(post.vars, post.raw);
td1.glob = preproc.getGlobalFrameData(td1.vars, td1.raw);
td2.glob = preproc.getGlobalFrameData(td2.vars, td2.raw);
%% 4. Filtering data and Mean subtraction
window = 6; % Window number
% Filtering data
pre.filt = preproc.globalDataFiltering(pre.glob, window, fs);
post.filt = preproc.globalDataFiltering(post.glob, window, fs);
td1.filt = preproc.globalDataFiltering(td1.glob, window, fs);
td2.filt = preproc.globalDataFiltering(td2.glob, window, fs);

%% III. Walks and turns management
%% 1. Orientation calculation
pre.angle = proc.getRPYAllIMUs(pre.filt, pre.Nr);
post.angle = proc.getRPYAllIMUs(post.filt, post.Nr);
td1.angle = proc.getRPYAllIMUs(td1.filt, td1.Nr);
td2.angle = proc.getRPYAllIMUs(td2.filt, td2.Nr);
%% 2. Turn detection
pre.turn = proc.turnDetectionWithYaw(pre.filt.time,   pre,  args.turn.pre);
post.turn = proc.turnDetectionWithYaw(post.filt.time, post, args.turn.post);
```

```
td1.turn = proc.turnDetectionWithYaw(td1.filt.time,   td1,  args.turn.td1);
td2.turn = proc.turnDetectionWithYaw(td2.filt.time,   td2,  args.turn.td2);
%% 3. Walks and turns cutting
pre = proc.getWalkSegments(pre);
post = proc.getWalkSegments(post);
td1 = proc.getWalkSegments(td1);
td2 = proc.getWalkSegments(td2);

%% IV. Gait cycles management
%% 1. Gait cycle detection
pre = proc.computeGaitEventsPerDataset(pre, args.GD.pre);
post = proc.computeGaitEventsPerDataset(post, args.GD.post);
td1 = proc.computeGaitEventsPerDataset(td1, args.GD.td1);
td2 = proc.computeGaitEventsPerDataset(td2, args.GD.td2);
%% 2. Gait cycle cutting
pre = proc.cutGaitCycleDataset(pre);
post = proc.cutGaitCycleDataset(post);
td1 = proc.cutGaitCycleDataset(td1);
td2 = proc.cutGaitCycleDataset(td2);

%% V. Parameters calculation
%% 1. Absolute angles
pre = param.getAbsoluteAnglesPerDataset(pre);
post = param.getAbsoluteAnglesPerDataset(post);
td1 = param.getAbsoluteAnglesPerDataset(td1);
td2 = param.getAbsoluteAnglesPerDataset(td2);
%% 2. Stride Spatiotemporal parameters
pre = param.getSpatiotemporalParametersPerDataset(pre);
post = param.getSpatiotemporalParametersPerDataset(post);
td1 = param.getSpatiotemporalParametersPerDataset(td1);
td2 = param.getSpatiotemporalParametersPerDataset(td2);
%% 3. Root Mean Square (RMS)
pre = param.getRMSPerDataset(pre);
post = param.getRMSPerDataset(post);
td1 = param.getRMSPerDataset(td1);
td2 = param.getRMSPerDataset(td2);
%% 4. Attenuation Coefficient (AC)
pre = param.getACPerDataset(pre);
post = param.getACPerDataset(post);
td1 = param.getACPerDataset(td1);
td2 = param.getACPerDataset(td2);
%% 5. Harmonic Ratio (HR)
pre = param.getHRPerDataset(pre);
post = param.getHRPerDataset(post);
td1 = param.getHRPerDataset(td1);
td2 = param.getHRPerDataset(td2);
%% 6. Relative angles
pre = param.getRelativeAnglesPerDataset(pre);
post = param.getRelativeAnglesPerDataset(post);
td1 = param.getRelativeAnglesPerDataset(td1);
td2 = param.getRelativeAnglesPerDataset(td2);
%% VI. Results
% Save results
param.saveParametersPerDataset(pre, post, td1, td2)
results.pre = pre; results.post = post;
results.td1 = td1; results.td2 = td2;
end
```

G.2 Preprocessing library

```
%% Pre-Processing functions library
% Author: Maria Alejandra Guzman Alfaro
% Last updated: 01.11.2024
function preproc = libPreProcessing
preproc.loadArguments=@loadArguments;
preproc.loadPatientData=@loadPatientData;
preproc.collectDataIMUs=@collectDataIMUs;
preproc.getRPYAngles=@getRPYAngles;
preproc.getGlobalFrameData=@getGlobalFrameData;
preproc.getRotationQuaternion=@getRotationQuaternion;
preproc.globalDataFiltering=@globalDataFiltering;
preproc.cutSignal=@cutSignal;
preproc.cutGlobalData=@cutGlobalData;
preproc.cutGCData=@cutGCData;
preproc.cutHSData=@cutHSData;
end
%% I. Data collection
%% 1. Load argument from patients
function args = loadArguments(patient)
% Getting arguments per patient
switch patient
    case 1
        % Participant 1: CP1
        args.patient = "CP1";
        % Turn thresholds per dataset:
        args.turn.pre = 12; args.turn.post = 10;
        args.turn.td1 = 10; args.turn.td2 = 15;
        % Gait events detection thresholds (per Dataset):
        % - Pre dataset
        args.GD.pre.HSTS = [1; 10; 1.5; 10];
        args.GD.pre.minHS = 0.3; args.GD.pre.rangeThres = 0.11;
        % - Post dataset
        args.GD.post.HSTS = [1; 10; 1.5; 10];
        args.GD.post.minHS = 0.3; args.GD.post.rangeThres = 0.11;
        % - TD Pre dataset
        args.GD.td1.HSTS = [1; 10; 1.5; 25];
        args.GD.td1.minHS = 0.3; args.GD.td1.rangeThres = 0.15;
        % - TD Post dataset
        args.GD.td2.HSTS = [1; 10; 1.5; 5];
        args.GD.td2.minHS = 0.3; args.GD.td2.rangeThres = 0.15;
    case 2
        % Participant 2: CP2
        args.patient = "CP2";
        % Turn thresholds per dataset:
        args.turn.pre = 12; args.turn.post = 10;
        args.turn.td1 = 10; args.turn.td2 = 15;
        % Initial contact detection thresholds (per Walk):
        % - Pre dataset
        args.GD.pre.HSTS = [1; 5; 1; 5];
        args.GD.pre.minHS = 0.2; args.GD.pre.rangeThres = 0.2;
        % - Post dataset
        args.GD.post.HSTS = [0.5; 5; 1.5; 5];
        args.GD.post.minHS = 0.3; args.GD.post.rangeThres = 0.16;
        % - TD Pre dataset
        args.GD.td1.HSTS = [2; 5; 0.5; 5];
        args.GD.td1.minHS = 0.3; args.GD.td1.rangeThres = 0.16;
        % - TD Post dataset
        args.GD.td2.HSTS = [2; 5; 0.5; 5];
        args.GD.td2.minHS = 0.3; args.GD.td2.rangeThres = 0.16;
    case 3
        % Participant 3: CP3
        args.patient = "CP3";
        % Turn thresholds per dataset:
        args.turn.pre = 12; args.turn.post = 10;
        args.turn.td1 = 10; args.turn.td2 = 15;
        % Initial contact detection thresholds (per Walk):
```

```
    % - Pre dataset
    args.GD.pre.HSTS = [0.5; 5; 1.5; 5];
    args.GD.pre.minHS = 0.3; args.GD.pre.rangeThres = 0.25;
    % - Post dataset
    args.GD.post.HSTS = [0.5; 5; 1.5; 5];
    args.GD.post.minHS = 0.3; args.GD.post.rangeThres = 0.25;
    % - TD Pre dataset
    args.GD.td1.HSTS = [1.5; 5; 0.5; 5];
    args.GD.td1.minHS = 0.35; args.GD.td1.rangeThres = 0.15;
    % - TD Post dataset
    args.GD.td2.HSTS = [2; 5; -1; 5];
    args.GD.td2.minHS = 0.4; args.GD.td2.rangeThres = 0.2;
case 4
    % Participant 4: CP4
    args.patient = "CP4";
    % Turn thresholds per dataset:
    args.turn.pre = 12; args.turn.post = 10;
    args.turn.td1 = 10; args.turn.td2 = 15;
    % Initial contact detection thresholds (per Walk):
    % - Pre dataset
    args.GD.pre.HSTS = [1.5; 5; 1; 5];
    args.GD.pre.minHS = 0.5; args.GD.pre.rangeThres = 0.24;
    % - Post dataset
    args.GD.post.HSTS = [1; 5; 1; 5];
    args.GD.post.minHS = 0.5; args.GD.post.rangeThres = 0.23;
    % - TD Pre dataset
    args.GD.td1.HSTS = [1; 5; 1; 5];
    args.GD.td1.minHS = 0.4; args.GD.td1.rangeThres = 0.23;
    % - TD Post dataset
    args.GD.td2.HSTS = [1; 5; 1; 5];
    args.GD.td2.minHS = 0.3; args.GD.td2.rangeThres = 0.23;
case 5
    % Participant 5: CP5
    args.patient = "CP5";
    % Turn thresholds per dataset:
    args.turn.pre = 12; args.turn.post = 10;
    args.turn.td1 = 15; args.turn.td2 = 15;
    % Initial contact detection thresholds (per Walk):
    % - Pre dataset
    args.GD.pre.HSTS = [1.25; 3; -0.5; 5];
    args.GD.pre.minHS = 0.4; args.GD.pre.rangeThres = 0.2;
    % - Post dataset
    args.GD.post.HSTS = [1.25; 3; -0.5; 5];
    args.GD.post.minHS = 0.4; args.GD.post.rangeThres = 0.2;
    % - TD Pre dataset
    args.GD.td1.HSTS = [1; 3; -0.5; 5];
    args.GD.td1.minHS = 0.4; args.GD.td1.rangeThres = 0.2;
    % - TD Post dataset
    args.GD.td2.HSTS = [1; 3; -0.5; 5];
    args.GD.td2.minHS = 0.4; args.GD.td2.rangeThres = 0.2;
case 6
    % Participant 6: CP6
    args.patient = "CP6";
    % Turn thresholds per dataset:
    args.turn.pre = 12; args.turn.post = 12;
    args.turn.td1 = 15; args.turn.td2 = 15;
    % Initial contact detection thresholds (per Walk):
    % - Pre dataset
    args.GD.pre.HSTS = [0.5; 5; 1; 5];
    args.GD.pre.minHS = 0.39; args.GD.pre.rangeThres = 0.15;
    % - Post dataset
    args.GD.post.HSTS = [0.5; 5; 1; 5];
    args.GD.post.minHS = 0.42; args.GD.post.rangeThres = 0.2;
    % - TD Pre dataset
    args.GD.td1.HSTS = [2; 3; 2; 5];
    args.GD.td1.minHS = 0.3; args.GD.td1.rangeThres = 0.15;
    % - TD Post dataset
    args.GD.td2.HSTS = [0.75; 3; -0.5; 5];
```

```
        args.GD.td2.minHS = 0.3; args.GD.td2.rangeThres = 0.15;
end
end
%% 2. Store participants data
function [pre, post, td1, td2] = loadPatientData(patient)
% Dataset name
pre.name = "CP-Pre"; post.name = "CP-Post";
td1.name = "TD1"; td2.name = "TD2";
% Getting dataset path
switch patient
    case 1
        % Participant 1:
        prePath = 'data/CP1/20230717-084536_Walk.xlsx';
        postPath = 'data/CP1/20230725-153640_Walk.xlsx';
        td1Path = 'data/CP1/20231130-163601_Walk-TD.xlsx';
        td2Path = 'data/CP1/20231211-162339_Walk-TD.xlsx';
        pre.patient = "CP1"; post.patient = "CP1"; pre.doc = "CP1"; post.doc = "CP1";
        td1.patient = "TD1"; td2.patient = "TD1"; td1.doc = "TD1"; td2.doc = "TD1";
    case 2
        % Participant 2:
        prePath = 'data/CP2/20230605-084828_Walk.xlsx';
        postPath = 'data/CP2/20230615-163306_Walk.xlsx';
        td1Path = 'data/CP2/20230911-143742_Walk-TD.xlsx';
        td2Path = 'data/CP2/20230922-142722_Walk-TD.xlsx';
        pre.patient = "CP2"; post.patient = "CP2"; pre.doc = "CP2"; post.doc = "CP2";
        td1.patient = "TD2"; td2.patient = "TD2"; td1.doc = "TD2"; td2.doc = "TD2";
    case 3
        % Participant 3:
        prePath = 'data/CP3/20231113-083739_Walk.xlsx';
        postPath = 'data/CP3/20231123-134703_Walk.xlsx';
        td1Path = 'data/CP3/20231219-101059_Walk-TD.xlsx';
        td2Path = 'data/CP3/20231229-094736_Walk-TD.xlsx';
        pre.patient = "CP3"; post.patient = "CP3"; pre.doc = "CP3"; post.doc = "CP3";
        td1.patient = "TD3"; td2.patient = "TD3"; td1.doc = "TD3"; td2.doc = "TD3";
    case 4
        % Participant 4:
        prePath = 'data/CP4/20230710-083803_Walk.xlsx';
        postPath = 'data/CP4/20230720-143552_Walk.xlsx';
        td1Path = 'data/CP4/20240606-130552_Walk-TD.xlsx';
        td2Path = 'data/CP4/20240617-125730_Walk-TD.xlsx';
        pre.patient = "CP4"; post.patient = "CP4"; pre.doc = "CP4"; post.doc = "CP4";
        td1.patient = "TD4"; td2.patient = "TD4"; td1.doc = "TD4"; td2.doc = "TD4";
    case 5
        % Participant 5:
        prePath = 'data/CP5/20231120-091114_Walk.xlsx';
        postPath = 'data/CP5/20231130-133929_Walk.xlsx';
        td1Path = 'data/CP5/20240222-135523_Walk-TD.xlsx';
        td2Path = 'data/CP5/20240304-085228_Walk-TD.xlsx';
        pre.patient = "CP5"; post.patient = "CP5"; pre.doc = "CP5"; post.doc = "CP5";
        td1.patient = "TD5"; td2.patient = "TD5"; td1.doc = "TD5"; td2.doc = "TD5";
    case 6
        % Participant 6:
        prePath = 'data/CP6/20240216-145723_Walk.xlsx';
        postPath = 'data/CP6/20240226-132731_Walk.xlsx';
        td1Path = 'data/CP6/20240527-131636_Walk-TD.xlsx';
        td2Path = 'data/CP6/20240607-130858_Walk-TD.xlsx';
        pre.patient = "CP6"; post.patient = "CP6"; pre.doc = "CP6"; post.doc = "CP6";
        td1.patient = "TD6"; td2.patient = "TD6"; td1.doc = "TD6"; td2.doc = "TD6";
end
% Getting dataset csv raw data
pre.csv = readtable(prePath);   pre.Nr = size(pre.csv, 1);
post.csv = readtable(postPath); post.Nr = size(post.csv, 1);
td1.csv = readtable(td1Path);   td1.Nr = size(td1.csv, 1);
td2.csv = readtable(td2Path);   td2.Nr = size(td2.csv, 1);
end
%% II. Raw data management
%% 1. Collect data from IMUs
function raw = collectDataIMUs(data, range)
```

```
% Collect data from IMUs
% All acceleration IMU data (in g)
acc.st = [data.S13578_Sternum__Acc0';   data.S13578_Sternum__Acc1';
data.S13578_Sternum__Acc2']';
acc.lu = [data.S14540_Lumbar__Acc0';    data.S14540_Lumbar__Acc1';
data.S14540_Lumbar__Acc2']';
acc.lf = [data.S14422_LeftFoot__Acc0';  data.S14422_LeftFoot__Acc1';
data.S14422_LeftFoot__Acc2']';
acc.rf = [data.S14433_RightFoot__Acc0'; data.S14433_RightFoot__Acc1';
data.S14433_RightFoot__Acc2']';
% All gyroscope IMU data
gyr.st = [data.S13578_Sternum__Gyr0';   data.S13578_Sternum__Gyr1';
data.S13578_Sternum__Gyr2']';
gyr.lu = [data.S14540_Lumbar__Gyr0';    data.S14540_Lumbar__Gyr1';
data.S14540_Lumbar__Gyr2']';
gyr.lf = [data.S14422_LeftFoot__Gyr0';  data.S14422_LeftFoot__Gyr1';
data.S14422_LeftFoot__Gyr2']';
gyr.rf = [data.S14433_RightFoot__Gyr0'; data.S14433_RightFoot__Gyr1';
data.S14433_RightFoot__Gyr2']';
% All magnetic IMU data
mag.st = [data.S13578_Sternum__Mag0';   data.S13578_Sternum__Mag1';
data.S13578_Sternum__Mag2']';
mag.lu = [data.S14540_Lumbar__Mag0';    data.S14540_Lumbar__Mag1';
data.S14540_Lumbar__Mag2']';
mag.lf = [data.S14422_LeftFoot__Mag0';  data.S14422_LeftFoot__Mag1';
data.S14422_LeftFoot__Mag2']';
mag.rf = [data.S14433_RightFoot__Mag0'; data.S14433_RightFoot__Mag1';
data.S14433_RightFoot__Mag2']';
% Time data
raw.time = linspace(0,range(end)/128,range(end));
% Give back raw data struct
raw.acc.sternum = acc.st; raw.acc.lumbar = acc.lu;
raw.acc.lfoot = acc.lf;   raw.acc.rfoot = acc.rf;
raw.gyr.sternum = gyr.st; raw.gyr.lumbar = gyr.lu;
raw.gyr.lfoot = gyr.lf;   raw.gyr.rfoot = gyr.rf;
raw.mag.sternum = mag.st; raw.mag.lumbar = mag.lu;
raw.mag.lfoot = mag.lf;   raw.mag.rfoot = mag.rf;
end
%% 2. Calculation of RPY angles throughout its calculation in smaller windows
function vars = getRPYAngles(wdw, struct)
% Inputs:
% - wdw: Window or interval of time to iterate
% - struct: Data struct with:
% + struct.acc: Acceleration struct
% + struct.gyr: Angular velocity struct
% + struct.mag: Magnetic field struct
% + struct.time: Time struct
acc = struct.acc; gyr = struct.gyr; mag = struct.mag; t = struct.time;
t_eval = buffer(t, round(wdw*128), 12); % Creating array with intervals of 32 elements
% Initializing arrays
gPerWindow = []; rollPerWindow = []; pitchPerWindow = []; yawPerWindow = [];
% Creating windows of 0.25 s to capture RPY angles from data
for i = 1:size(t_eval,1)
    % Since we have the time range we need the frames or timestamps
    rng = nonzeros(round(t_eval(:,i)*128));
    % IMU: STERNUM
    mean_acc.sternum = mean(acc.sternum(rng,:));
    gEstimated.sternum = sqrt(mean_acc.sternum(1)^2 + mean_acc.sternum(2)^2 +
mean_acc.sternum(3)^2);
    % Giving the acceleration, gyroscope data and time range
    [roll.sternum, pitch.sternum, yaw.sternum] = getRPYwithComplementaryFilter(acc.sternum,
gyr.sternum, mag.sternum, rng);
    % Getting estimated variables per window
    gPerWindow.sternum(i) = gEstimated.sternum;      rollPerWindow.sternum(i) =
mean(roll.sternum);
    pitchPerWindow.sternum(i) = mean(pitch.sternum); yawPerWindow.sternum(i) =
mean(yaw.sternum);
    % IMU: LUMBAR
```

```
    mean_acc.lumbar = mean(acc.lumbar(rng,:));
    gEstimated.lumbar = sqrt(mean_acc.lumbar(1)^2 + mean_acc.lumbar(2)^2 +
mean_acc.lumbar(3)^2);
    % Giving the acceleration, gyroscope data and time range
    [roll.lumbar, pitch.lumbar, yaw.lumbar] = getRPYwithComplementaryFilter(acc.lumbar,
gyr.lumbar, mag.lumbar, rng);
    % Getting estimated variables per window
    gPerWindow.lumbar(i) = gEstimated.lumbar;      rollPerWindow.lumbar(i) =
mean(roll.lumbar);
    pitchPerWindow.lumbar(i) = mean(pitch.lumbar); yawPerWindow.lumbar(i) = mean(yaw.lumbar);
    % IMU: RIGHT FOOT
    mean_acc.rfoot = mean(acc.rfoot(rng,:));
    gEstimated.rfoot = sqrt(mean_acc.rfoot(1)^2 + mean_acc.rfoot(2)^2 + mean_acc.rfoot(3)^2);
    % Giving the acceleration, gyroscope data and time range
    [roll.rfoot, pitch.rfoot, yaw.rfoot] = getRPYwithComplementaryFilter(acc.rfoot, gyr.rfoot,
mag.rfoot, rng);
    % Getting estimated variables per window
    gPerWindow.rfoot(i) = gEstimated.rfoot;      rollPerWindow.rfoot(i) = mean(roll.rfoot);
    pitchPerWindow.rfoot(i) = mean(pitch.rfoot); yawPerWindow.rfoot(i) = mean(yaw.rfoot);
    % IMU: LEFT FOOT
    mean_acc.lfoot = mean(acc.lfoot(rng,:));
    gEstimated.lfoot = sqrt(mean_acc.lfoot(1)^2 + mean_acc.lfoot(2)^2 + mean_acc.lfoot(3)^2);
    % Giving the acceleration, gyroscope data and time range
    [roll.lfoot, pitch.lfoot, yaw.lfoot] = getRPYwithComplementaryFilter(acc.lfoot, gyr.lfoot,
mag.lfoot, rng);
    % Getting estimated variables per window
    gPerWindow.lfoot(i) = gEstimated.lfoot;      rollPerWindow.lfoot(i) = mean(roll.lfoot);
    pitchPerWindow.lfoot(i) = mean(pitch.lfoot); yawPerWindow.lfoot(i) = mean(yaw.lfoot);
end
% Getting mean values per angle and estimated gravity
% Right foot data
meanGravity.rfoot = mean(gPerWindow.rfoot);   meanRoll.rfoot = mean(rollPerWindow.rfoot);
meanPitch.rfoot = mean(pitchPerWindow.rfoot); meanYaw.rfoot = mean(yawPerWindow.rfoot);
% Left foot data
meanGravity.lfoot = mean(gPerWindow.lfoot);   meanRoll.lfoot = mean(rollPerWindow.lfoot);
meanPitch.lfoot = mean(pitchPerWindow.lfoot); meanYaw.lfoot = mean(yawPerWindow.lfoot);
% Sternum data
meanGravity.sternum = mean(gPerWindow.sternum);   meanRoll.sternum =
mean(rollPerWindow.sternum);
meanPitch.sternum = mean(pitchPerWindow.sternum); meanYaw.sternum =
mean(yawPerWindow.sternum);
% Lumbar data
meanGravity.lumbar = mean(gPerWindow.lumbar);   meanRoll.lumbar = mean(rollPerWindow.lumbar);
meanPitch.lumbar = mean(pitchPerWindow.lumbar); meanYaw.lumbar = mean(yawPerWindow.lumbar);
% Storing data in 'vars' struct
vars.g = meanGravity; vars.roll = meanRoll; vars.pitch = meanPitch; vars.yaw = meanYaw;
end
% Euler angles calculation with Complementary Filter
function [roll_deg, pitch_deg, yaw_deg] = getRPYwithComplementaryFilter(acc_data, gyr_data,
mag_data, rng)
fs = 128;
% Acceleration data
ax = acc_data(rng, 1); ay = acc_data(rng, 2); az = acc_data(rng, 3);
% Angular velocity data
gyr_data(isnan(gyr_data))=0;
% Convert gyroscope from °/s to rad/s
gyr_data = gyr_data*pi/180;
gx = gyr_data(rng, 1); gy = gyr_data(rng, 2); gz = gyr_data(rng, 3);
% Magnetic field data
mx = mag_data(:,1); my = mag_data(:,2); mz = mag_data(:,3);
Mx = zeros(1, length(ax)); My = zeros(1, length(ax));
i = 1; dt = 1/fs; % Time differential
% Initializing arrays
pitch = zeros(1, length(ax)); roll = zeros(1, length(ax)); yaw = zeros(1, length(ax));
% Calculation of inital RPY angles with geometry
% Getting roll from accelerometer data
roll(i) = atan2(ay(i),az(i));
% Getting pitch from accelerometer data
```

```
pitch(i) = atan2(-ax(i), (ay(i)*sin(roll(i)) + az(i)*cos(roll(i))));
% Getting yaw from magnetometer data
Mx(i) = mx(i)*cos(pitch(i)) + my(i)*sin(pitch(i))*sin(roll(i)) +
mz(i)*sin(pitch(i))*cos(roll(i));
My(i) = my(i)*cos(roll(i)) - mz(i)*sin(roll(i));
yaw(i) = atan2(My(i), Mx(i));
% Coefficient for the complementary filter
alpha = 0.95;
% Estimation of roll, pitch and yaw per iteration
for i = 2:length(ax)
    roll(i) = atan2(ay(i),az(i)); % Roll in x axis
    pitch(i) = atan2(-ax(i), (ay(i)*sin(roll(i)) + az(i)*cos(roll(i)))); % Pitch in y axis
    Mx(i) = mx(i)*cos(pitch(i)) + my(i)*sin(pitch(i))*sin(roll(i)) +
mz(i)*sin(pitch(i))*cos(roll(i));
    My(i) = my(i)*cos(roll(i)) - mz(i)*sin(roll(i));
    yaw(i) = atan2(My(i), Mx(i)); % Yaw in z axis
    % Complementary filter with gyroscope data
    roll(i) = alpha*(roll(i-1) + gx(i-1)*dt) + (1 - alpha)*roll(i);
    pitch(i) = alpha*(pitch(i-1) + gy(i-1)*dt) + (1 - alpha)*pitch(i);
    yaw(i) = alpha*(yaw(i-1) + gz(i-1)*dt) + (1 - alpha)*yaw(i);
end
% Converting angles from radians to degrees
roll_deg = rad2deg(roll);   % Roll angle around x axis
pitch_deg = rad2deg(pitch); % Pitch angle around y axis
yaw_deg = rad2deg(yaw);     % Yaw angle around z axis
end
%% 3. Rotation of data from initial to global frame
function glob = getGlobalFrameData(struct, raw)
roll = struct.roll; pitch = struct.pitch; yaw = struct.yaw;
% Calling quaternions library
qfuns = libQuaternions;
% Doing the data rotation with quaternions per IMU
% Sternum IMU
glob.qRot.sternum = getRotationQuaternion(qfuns, roll.sternum, pitch.sternum, yaw.sternum);
[glob.sternum.acc, glob.sternum.gyr, glob.sternum.mag] =
qfuns.imuDataRotation(raw.acc.sternum, ...
    raw.gyr.sternum, raw.mag.sternum, glob.qRot.sternum);
% Lumbar IMU
glob.qRot.lumbar = getRotationQuaternion(qfuns, roll.lumbar, pitch.lumbar, yaw.lumbar);
[glob.lumbar.acc, glob.lumbar.gyr, glob.lumbar.mag] = qfuns.imuDataRotation(raw.acc.lumbar,
...
    raw.gyr.lumbar, raw.mag.lumbar, glob.qRot.lumbar);
% Right Foot IMU
glob.qRot.rfoot = getRotationQuaternion(qfuns, roll.rfoot, pitch.rfoot, yaw.rfoot);
[glob.rfoot.acc, glob.rfoot.gyr, glob.rfoot.mag] = qfuns.imuDataRotation(raw.acc.rfoot, ...
    raw.gyr.rfoot, raw.mag.rfoot, glob.qRot.rfoot);
% Left Foot IMU
glob.qRot.lfoot = getRotationQuaternion(qfuns, roll.lfoot, pitch.lfoot, yaw.lfoot);
[glob.lfoot.acc, glob.lfoot.gyr, glob.lfoot.mag] = qfuns.imuDataRotation(raw.acc.lfoot, ...
    raw.gyr.lfoot, raw.mag.lfoot, glob.qRot.lfoot);
% Transfering time data
glob.time = raw.time;
end
function q = getRotationQuaternion(qfuns, roll, pitch, yaw)
% Creating quaternion for each RPY angle and axis
qRoll = qfuns.create_quaternion(roll, [1 0 0]);
qPitch = qfuns.create_quaternion(pitch, [0 1 0]);
qYaw = qfuns.create_quaternion(yaw, [0 0 1]);
% Calculating the rotated quaternion (q = qRoll*qPitch*qYaw)
q = qfuns.quaternionProduct(qYaw, qfuns.quaternionProduct(qPitch, qRoll));
end
%% 4. Filters signal with Moving Average Window filter
% Filtering global structs
function filt = globalDataFiltering(struct, fc, fs)
% When type is "mov" fc converts into window variable
% Assign time
filt.time = struct.time;
% Assign acceleration and angular velocity per IMU
```

```
filt.sternum.acc = preprocessing(struct.sternum.acc, fc, fs, "mov");
filt.sternum.gyr = preprocessing(struct.sternum.gyr, fc, fs, "mov");
filt.sternum.mag = preprocessing(struct.sternum.mag, fc, fs, "mov");
filt.lumbar.acc = preprocessing(struct.lumbar.acc, fc, fs, "mov");
filt.lumbar.gyr = preprocessing(struct.lumbar.gyr, fc, fs, "mov");
filt.lumbar.mag = preprocessing(struct.lumbar.mag, fc, fs, "mov");
filt.rfoot.acc = preprocessing(struct.rfoot.acc, fc, fs, "mov");
filt.rfoot.gyr = preprocessing(struct.rfoot.gyr, fc, fs, "mov");
filt.rfoot.mag = preprocessing(struct.rfoot.mag, fc, fs, "mov");
filt.lfoot.acc = preprocessing(struct.lfoot.acc, fc, fs, "mov");
filt.lfoot.gyr = preprocessing(struct.lfoot.gyr, fc, fs, "mov");
filt.lfoot.mag = preprocessing(struct.lfoot.mag, fc, fs, "mov");
end
function signal = preprocessing(data, fc, fs, filterType)
% First, convert NaN data to 0
data(isnan(data)) = 0;
data = data - mean(data(1:3*fs,:));
% Finally, filter
if filterType == "band"
    signal = bandpass(data,[0.5 fc], fs);
elseif filterType == "mov"
    signal = movmean(data, fc);
end
end
%% Other functions:
%% Cut raw data given a crop range
function signal = cutSignal(struct,crop_range)
% --- Input ---
% struct: Struct data with 3 channels (axis)
% crop_range: Vector of 2 elements
% Acceleration struct
signal.acc.sternum = struct.acc.sternum(crop_range(1):crop_range(2),:);
signal.acc.lumbar = struct.acc.lumbar(crop_range(1):crop_range(2),:);
signal.acc.lfoot = struct.acc.lfoot(crop_range(1):crop_range(2),:);
signal.acc.rfoot = struct.acc.rfoot(crop_range(1):crop_range(2),:);
% Gyroscope struct
signal.gyr.sternum = struct.gyr.sternum(crop_range(1):crop_range(2),:);
signal.gyr.lumbar = struct.gyr.lumbar(crop_range(1):crop_range(2),:);
signal.gyr.lfoot = struct.gyr.lfoot(crop_range(1):crop_range(2),:);
signal.gyr.rfoot = struct.gyr.rfoot(crop_range(1):crop_range(2),:);
% Magnetometer struct
signal.mag.sternum = struct.mag.sternum(crop_range(1):crop_range(2),:);
signal.mag.lumbar = struct.mag.lumbar(crop_range(1):crop_range(2),:);
signal.mag.lfoot = struct.mag.lfoot(crop_range(1):crop_range(2),:);
signal.mag.rfoot = struct.mag.rfoot(crop_range(1):crop_range(2),:);
% Time struct
signal.time = struct.time(:,crop_range(1):crop_range(2));
% struct
signal.acc = acc; signal.gyr = gyr; signal.mag = mag; signal.time = time;
end
%% Cut global frame data given a crop range
function signal = cutGlobalData(struct,crop_range)
% --- Input ---
% struct: Struct data with 3 channels (axis)
% crop_range: Vector of 2 elements
% Acceleration struct
signal.sternum.acc = struct.filt.sternum.acc(crop_range(1):crop_range(2),:);
signal.lumbar.acc = struct.filt.lumbar.acc(crop_range(1):crop_range(2),:);
signal.lfoot.acc = struct.filt.lfoot.acc(crop_range(1):crop_range(2),:);
signal.rfoot.acc = struct.filt.rfoot.acc(crop_range(1):crop_range(2),:);
% Gyroscope struct
signal.sternum.gyr = struct.filt.sternum.gyr(crop_range(1):crop_range(2),:);
signal.lumbar.gyr = struct.filt.lumbar.gyr(crop_range(1):crop_range(2),:);
signal.lfoot.gyr = struct.filt.lfoot.gyr(crop_range(1):crop_range(2),:);
signal.rfoot.gyr = struct.filt.rfoot.gyr(crop_range(1):crop_range(2),:);
% Magnetometer struct
signal.sternum.mag = struct.filt.sternum.mag(crop_range(1):crop_range(2),:);
signal.lumbar.mag = struct.filt.lumbar.mag(crop_range(1):crop_range(2),:);
```

```
signal.lfoot.mag = struct.filt.lfoot.mag(crop_range(1):crop_range(2),:);
signal.rfoot.mag = struct.filt.rfoot.mag(crop_range(1):crop_range(2),:);
% Roll angle struct
signal.sternum.roll = struct.angle.sternum.roll(:,crop_range(1):crop_range(2));
signal.lumbar.roll = struct.angle.lumbar.roll(:,crop_range(1):crop_range(2));
signal.lfoot.roll = struct.angle.lfoot.roll(:,crop_range(1):crop_range(2));
signal.rfoot.roll = struct.angle.rfoot.roll(:,crop_range(1):crop_range(2));
% Pitch angle struct
signal.sternum.pitch = struct.angle.sternum.pitch(:,crop_range(1):crop_range(2));
signal.lumbar.pitch = struct.angle.lumbar.pitch(:,crop_range(1):crop_range(2));
signal.lfoot.pitch = struct.angle.lfoot.pitch(:,crop_range(1):crop_range(2));
signal.rfoot.pitch = struct.angle.rfoot.pitch(:,crop_range(1):crop_range(2));
% Yaw angle struct
signal.sternum.yaw = struct.angle.sternum.yaw(:,crop_range(1):crop_range(2));
signal.lumbar.yaw = struct.angle.lumbar.yaw(:,crop_range(1):crop_range(2));
signal.lfoot.yaw = struct.angle.lfoot.yaw(:,crop_range(1):crop_range(2));
signal.rfoot.yaw = struct.angle.rfoot.yaw(:,crop_range(1):crop_range(2));
% Quaternion struct
signal.sternum.q = struct.angle.sternum.q(crop_range(1):crop_range(2),:);
signal.lumbar.q = struct.angle.lumbar.q(crop_range(1):crop_range(2),:);
signal.lfoot.q = struct.angle.lfoot.q(crop_range(1):crop_range(2),:);
signal.rfoot.q = struct.angle.rfoot.q(crop_range(1):crop_range(2),:);
% Yaw angle struct
signal.sternum.R = struct.angle.sternum.R(:,:,crop_range(1):crop_range(2));
signal.lumbar.R = struct.angle.lumbar.R(:,:,crop_range(1):crop_range(2));
signal.lfoot.R = struct.angle.lfoot.R(:,:,crop_range(1):crop_range(2));
signal.rfoot.R = struct.angle.rfoot.R(:,:,crop_range(1):crop_range(2));
% Time struct
signal.time = struct.filt.time(crop_range(1):crop_range(2));
end
%% Cut GC data given a crop range
function signal = cutGCData(struct,crop_range)
% --- Input ---
% struct: Struct data with 3 channels (axis)
% crop_range: Vector of 2 elements
% Acceleration struct
signal.sternum.acc = struct.sternum.acc(crop_range(1):crop_range(2),:);
signal.lumbar.acc = struct.lumbar.acc(crop_range(1):crop_range(2),:);
signal.lfoot.acc = struct.lfoot.acc(crop_range(1):crop_range(2),:);
signal.rfoot.acc = struct.rfoot.acc(crop_range(1):crop_range(2),:);
% Gyroscope struct
signal.sternum.gyr = struct.sternum.gyr(crop_range(1):crop_range(2),:);
signal.lumbar.gyr = struct.lumbar.gyr(crop_range(1):crop_range(2),:);
signal.lfoot.gyr = struct.lfoot.gyr(crop_range(1):crop_range(2),:);
signal.rfoot.gyr = struct.rfoot.gyr(crop_range(1):crop_range(2),:);
% Magnetometer struct
signal.sternum.mag = struct.sternum.mag(crop_range(1):crop_range(2),:);
signal.lumbar.mag = struct.lumbar.mag(crop_range(1):crop_range(2),:);
signal.lfoot.mag = struct.lfoot.mag(crop_range(1):crop_range(2),:);
signal.rfoot.mag = struct.rfoot.mag(crop_range(1):crop_range(2),:);
% Roll angle struct
signal.sternum.roll = struct.sternum.roll(:,crop_range(1):crop_range(2));
signal.lumbar.roll = struct.lumbar.roll(:,crop_range(1):crop_range(2));
signal.lfoot.roll = struct.lfoot.roll(:,crop_range(1):crop_range(2));
signal.rfoot.roll = struct.rfoot.roll(:,crop_range(1):crop_range(2));
% Pitch angle struct
signal.sternum.pitch = struct.sternum.pitch(:,crop_range(1):crop_range(2));
signal.lumbar.pitch = struct.lumbar.pitch(:,crop_range(1):crop_range(2));
signal.lfoot.pitch = struct.lfoot.pitch(:,crop_range(1):crop_range(2));
signal.rfoot.pitch = struct.rfoot.pitch(:,crop_range(1):crop_range(2));
% Yaw angle struct
signal.sternum.yaw = struct.sternum.yaw(:,crop_range(1):crop_range(2));
signal.lumbar.yaw = struct.lumbar.yaw(:,crop_range(1):crop_range(2));
signal.lfoot.yaw = struct.lfoot.yaw(:,crop_range(1):crop_range(2));
signal.rfoot.yaw = struct.rfoot.yaw(:,crop_range(1):crop_range(2));
% Quaternion struct
signal.sternum.q = struct.sternum.q(crop_range(1):crop_range(2),:);
signal.lumbar.q = struct.lumbar.q(crop_range(1):crop_range(2),:);
```

```
signal.lfoot.q = struct.lfoot.q(crop_range(1):crop_range(2),:);
signal.rfoot.q = struct.rfoot.q(crop_range(1):crop_range(2),:);
% Yaw angle struct
signal.sternum.R = struct.sternum.R(:,:,crop_range(1):crop_range(2));
signal.lumbar.R = struct.lumbar.R(:,:,crop_range(1):crop_range(2));
signal.lfoot.R = struct.lfoot.R(:,:,crop_range(1):crop_range(2));
signal.rfoot.R = struct.rfoot.R(:,:,crop_range(1):crop_range(2));
% Time struct
signal.time = struct.time(crop_range(1):crop_range(2));
end
%% Cut data given a crop range (angles)
function signal = cutHSData(struct,crop_range)
% --- Input ---
% struct: Struct data with 3 channels (axis)
% crop_range: Vector of 2 elements
signal = struct;
% Acceleration struct
signal.sternum.acc = struct.sternum.acc(crop_range(1):crop_range(2),:);
signal.lumbar.acc = struct.lumbar.acc(crop_range(1):crop_range(2),:);
signal.lfoot.acc = struct.lfoot.acc(crop_range(1):crop_range(2),:);
signal.rfoot.acc = struct.rfoot.acc(crop_range(1):crop_range(2),:);
% Gyroscope struct
signal.sternum.gyr = struct.sternum.gyr(crop_range(1):crop_range(2),:);
signal.lumbar.gyr = struct.lumbar.gyr(crop_range(1):crop_range(2),:);
signal.lfoot.gyr = struct.lfoot.gyr(crop_range(1):crop_range(2),:);
signal.rfoot.gyr = struct.rfoot.gyr(crop_range(1):crop_range(2),:);
% Magnetometer struct
signal.sternum.mag = struct.sternum.mag(crop_range(1):crop_range(2),:);
signal.lumbar.mag = struct.lumbar.mag(crop_range(1):crop_range(2),:);
signal.lfoot.mag = struct.lfoot.mag(crop_range(1):crop_range(2),:);
signal.rfoot.mag = struct.rfoot.mag(crop_range(1):crop_range(2),:);
% Roll angle struct
signal.sternum.roll = struct.sternum.roll(:,crop_range(1):crop_range(2));
signal.lumbar.roll = struct.lumbar.roll(:,crop_range(1):crop_range(2));
signal.lfoot.roll = struct.lfoot.roll(:,crop_range(1):crop_range(2));
signal.rfoot.roll = struct.rfoot.roll(:,crop_range(1):crop_range(2));
% Pitch angle struct
signal.sternum.pitch = struct.sternum.pitch(:,crop_range(1):crop_range(2));
signal.lumbar.pitch = struct.lumbar.pitch(:,crop_range(1):crop_range(2));
signal.lfoot.pitch = struct.lfoot.pitch(:,crop_range(1):crop_range(2));
signal.rfoot.pitch = struct.rfoot.pitch(:,crop_range(1):crop_range(2));
% Yaw angle struct
signal.sternum.yaw = struct.sternum.yaw(:,crop_range(1):crop_range(2));
signal.lumbar.yaw = struct.lumbar.yaw(:,crop_range(1):crop_range(2));
signal.lfoot.yaw = struct.lfoot.yaw(:,crop_range(1):crop_range(2));
signal.rfoot.yaw = struct.rfoot.yaw(:,crop_range(1):crop_range(2));
% Quaternion struct
signal.sternum.q = struct.sternum.q(crop_range(1):crop_range(2),:);
signal.lumbar.q = struct.lumbar.q(crop_range(1):crop_range(2),:);
signal.lfoot.q = struct.lfoot.q(crop_range(1):crop_range(2),:);
signal.rfoot.q = struct.rfoot.q(crop_range(1):crop_range(2),:);
% Yaw angle struct
signal.sternum.R = struct.sternum.R(:,:,crop_range(1):crop_range(2));
signal.lumbar.R = struct.lumbar.R(:,:,crop_range(1):crop_range(2));
signal.lfoot.R = struct.lfoot.R(:,:,crop_range(1):crop_range(2));
signal.rfoot.R = struct.rfoot.R(:,:,crop_range(1):crop_range(2));
% Time struct
signal.time = struct.time(crop_range(1):crop_range(2));
end
```

G.3 Processing library

```
%% Processing functions library
% Author: Maria Alejandra Guzman Alfaro
% Last updated: 01.11.2024
function proc = libProcessing
proc.getRPYAllIMUs=@getRPYAllIMUs;
proc.getRPYPerDataset=@getRPYPerDataset;
proc.turnDetectionWithYaw=@turnDetectionWithYaw;
proc.getWalkSegments=@getWalkSegments;
proc.computeGaitEventsPerDataset=@computeGaitEventsPerDataset;
proc.cutGaitCycleDataset=@cutGaitCycleDataset;
proc.cutGaitCyclesPerWalk=@cutGaitCyclesPerWalk;
end

%% III. Walks and turns management
%% 1. Orientation calculation
% Get the Euler angles of all the IMUs
function orientation = getRPYAllIMUs(struct, Nr)
q1 = [1 0 0 0]; q2 = [1 0 0 0]; q3 = [1 0 0 0]; q4 = [1 0 0 0];
for i = 1:Nr
    % Sternum IMU
    [orientation.sternum.roll(i), orientation.sternum.pitch(i), orientation.sternum.yaw(i),
...
        q1, R1] = getRPYCFwithQuaternions(struct.sternum, i, q1);
    % Lumbar IMU
    [orientation.lumbar.roll(i), orientation.lumbar.pitch(i), orientation.lumbar.yaw(i), ...
        q2, R2] = getRPYCFwithQuaternions(struct.lumbar, i, q2);
    % Right Foot IMU
    [orientation.rfoot.roll(i), orientation.rfoot.pitch(i), orientation.rfoot.yaw(i), ...
        q3, R3] = getRPYCFwithQuaternions(struct.rfoot, i, q3);
    % Left Foot IMU
    [orientation.lfoot.roll(i), orientation.lfoot.pitch(i), orientation.lfoot.yaw(i), ...
        q4, R4] = getRPYCFwithQuaternions(struct.lfoot, i, q4);
    % Storing orientation quaternions
    orientation.sternum.q(i,:) = q1; orientation.lumbar.q(i,:) = q2;
    orientation.rfoot.q(i,:) = q3;   orientation.lfoot.q(i,:) = q4;
    orientation.sternum.R(:,:,i) = R1; orientation.lumbar.R(:,:,i) = R2;
    orientation.rfoot.R(:,:,i) = R3;   orientation.lfoot.R(:,:,i) = R4;
end
end
% Gets Euler angles, quaternions and rotation matrices for each IMU
function [r, p, y, q, R] = getRPYCFwithQuaternions(struct, i, q)
preproc = libPreProcessing; qfuns = libQuaternions; % Libraries
% Angular velocity, acceleration and magnetometer data
gx = struct.gyr(i,1); gy = struct.gyr(i,2); gz = struct.gyr(i,3);
ax = struct.acc(i,1); ay = struct.acc(i,2); az = struct.acc(i,3);
mx = struct.mag(i,1); my = struct.mag(i,2); mz = struct.mag(i,3);
alpha = 0.95; dt = 1/128;  % Constants
dq = [0, gx, gy, gz]*dt/2; % Angular velocity rate
% Update quaternion with gyroscope data
qGyr = q + qfuns.quaternionProduct(q, dq);
% Normalize the quaternion from gyroscope
qGyr = qGyr/norm(qGyr);
% Estimate pitch (theta) and roll (phi) from the accelerometer
roll = atan2(ay, az);
pitch = atan2(-ax, (ay*sin(roll) + az*cos(roll)));
% Getting yaw from magnetometer data
MX = mx*cos(pitch) + my*sin(pitch)*sin(roll) + mz*sin(pitch)*cos(roll);
MY = my*cos(roll) - mz*sin(roll);
yaw = atan2(MY, MX);
% Convert accelerometer and magnetometer angles to a quaternion
qFilter = preproc.getRotationQuaternion(qfuns, roll, pitch, yaw);
% Complementary filter: combine gyro, accelerometer and magnetometer quaternions
q = alpha*qGyr + (1-alpha)*qFilter;
% Normalization of the quaternion (unit quaternion)
q = q/norm(q);
% Converting quaternion to Rotation matrix
```

```
R = quat2rotm(q);
% Converting quaternion to RPY euler angles
rpy = rad2deg(quat2eul(q,'XYZ')); r = rpy(1); p = rpy(2); y = rpy(3);
end
function struct = getRPYPerDataset(struct)
struct.W1.orientation = getRPYAllIMUs(struct.W1, struct.W1.Nr);
struct.W2.orientation = getRPYAllIMUs(struct.W2, struct.W2.Nr);
struct.W3.orientation = getRPYAllIMUs(struct.W3, struct.W3.Nr);
struct.W4.orientation = getRPYAllIMUs(struct.W4, struct.W4.Nr);
struct.W5.orientation = getRPYAllIMUs(struct.W5, struct.W5.Nr);
struct.W6.orientation = getRPYAllIMUs(struct.W6, struct.W6.Nr);
end
%% 2. Turn detection using only Yaw orientation data
function turn = turnDetectionWithYaw(t, struct, thresh)
wdw = 0.25;     % window time
nrTurns = 5;    % initial turn number
% Gathering orientation and time variables
yawSternum = struct.angle.sternum.yaw; yawLumbar = struct.angle.lumbar.yaw;
% Creating array with intervals
t_eval = buffer(t, round(wdw*128), 12);
for i = 1:size(t_eval,2)
    % Since we have the time range we need the frames or timestamps
    range = nonzeros(round(t_eval(:,i)*128));
    yawPerWindow.sternum = yawSternum(1,range);
    yawPerWindow.lumbar = yawLumbar(1,range);
    % Get mean yaw per window
    meanYaw.sternum(i) = mean(yawPerWindow.sternum);
    meanYaw.lumbar(i) = mean(yawPerWindow.lumbar);
end
% Find local minima and maxima from both sternum and lumbar data
[~, minLocs.sternum] = findpeaks(-meanYaw.sternum, 'MinPeakHeight', thresh,
'MinPeakDistance',20);
[~, minLocs.lumbar] = findpeaks(-meanYaw.lumbar, 'MinPeakHeight', thresh,
'MinPeakDistance',20);
[~, maxLocs.sternum] = findpeaks(meanYaw.sternum, 'MinPeakHeight', thresh,
'MinPeakDistance',20);
[~, maxLocs.lumbar] = findpeaks(meanYaw.lumbar, 'MinPeakHeight', thresh,
'MinPeakDistance',20);
% Assure that the arrays have same length
if length(minLocs.sternum)> length(minLocs.lumbar)
    minLocs.lumbar = [minLocs.lumbar minLocs.sternum(end)];
elseif length(minLocs.lumbar)> length(minLocs.sternum)
    minLocs.sternum = [minLocs.sternum minLocs.lumbar(end)];
elseif length(maxLocs.sternum)> length(maxLocs.lumbar)
    maxLocs.lumbar = [maxLocs.lumbar maxLocs.sternum(end)];
elseif length(maxLocs.lumbar)> length(maxLocs.sternum)
    maxLocs.sternum = [maxLocs.sternum maxLocs.lumbar(end)];
end
% sort location values
locs.sternum = sort([minLocs.sternum maxLocs.sternum]);
locs.lumbar = sort([minLocs.lumbar maxLocs.lumbar]);
% Find zero crossing values
idx.sternum = find(meanYaw.sternum(2:end).*meanYaw.sternum(1:end-1) < 0);
idx.lumbar = find(meanYaw.lumbar(2:end).*meanYaw.lumbar(1:end-1) < 0);
% In case there is a detection of a sixth turn
if length(minLocs.sternum) == 6 || length(minLocs.lumbar) == 6 || length(maxLocs.sternum) == 6
|| ...
        length(maxLocs.lumbar) == 6 || (length(minLocs.sternum) + length(maxLocs.sternum)) ==
6
    nrTurns = 6;
end
for j = 1:nrTurns
    beforePeak.sternum(j) = find(idx.sternum < locs.sternum(j), 1, 'last');
    beforePeak.lumbar(j) = find(idx.lumbar  < locs.lumbar(j), 1, 'last');
    if j < 6
        afterPeak.sternum(j) = find(idx.sternum > locs.sternum(j), 1, 'first');
        afterPeak.lumbar(j) = find(idx.lumbar > locs.lumbar(j), 1, 'first');
```

```
        zeroCrossing.sternum = sort([idx.sternum(beforePeak.sternum)
idx.sternum(afterPeak.sternum)]);
        zeroCrossing.lumbar = sort([idx.lumbar(beforePeak.lumbar)
idx.lumbar(afterPeak.lumbar)]);
    elseif nrTurns == 6 && j == nrTurns
        zeroCrossing.sternum = sort([idx.sternum(beforePeak.sternum)
idx.sternum(afterPeak.sternum)]);
        zeroCrossing.lumbar = sort([idx.lumbar(beforePeak.lumbar)
idx.lumbar(afterPeak.lumbar)]);
    end
end
t1.sternum = zeroCrossing.sternum(1:2);  t1.lumbar = zeroCrossing.lumbar(1:2);
t2.sternum = zeroCrossing.sternum(3:4);  t2.lumbar = zeroCrossing.lumbar(3:4);
t3.sternum = zeroCrossing.sternum(5:6);  t3.lumbar = zeroCrossing.lumbar(5:6);
t4.sternum = zeroCrossing.sternum(7:8);  t4.lumbar = zeroCrossing.lumbar(7:8);
t5.sternum = zeroCrossing.sternum(9:10); t5.lumbar = zeroCrossing.lumbar(9:10);
T1(1) = max(t1.sternum(1), t1.lumbar(1)); T1(2) = min(t1.sternum(2), t1.lumbar(2));
T2(1) = max(t2.sternum(1), t2.lumbar(1)); T2(2) = min(t2.sternum(2), t2.lumbar(2));
T3(1) = max(t3.sternum(1), t3.lumbar(1)); T3(2) = min(t3.sternum(2), t3.lumbar(2));
T4(1) = max(t4.sternum(1), t4.lumbar(1)); T4(2) = min(t4.sternum(2), t4.lumbar(2));
T5(1) = max(t5.sternum(1), t5.lumbar(1)); T5(2) = min(t5.sternum(2), t5.lumbar(2));
turn.t1(1) = max(t_eval(:,T1(1))); turn.t1(2) = min(t_eval(:,T1(2)));
turn.t2(1) = max(t_eval(:,T2(1))); turn.t2(2) = min(t_eval(:,T2(2)));
turn.t3(1) = max(t_eval(:,T3(1))); turn.t3(2) = min(t_eval(:,T3(2)));
turn.t4(1) = max(t_eval(:,T4(1))); turn.t4(2) = min(t_eval(:,T4(2)));
turn.t5(1) = max(t_eval(:,T5(1))); turn.t5(2) = min(t_eval(:,T5(2)));
turn.Nr = 5;
if nrTurns == 6
    t6.sternum = zeroCrossing.sternum(11); t6.lumbar = zeroCrossing.lumbar(11);
    T6 = max(t6.sternum, t6.lumbar); turn.t6 = max(t_eval(:,T6)); turn.Nr = 6;
end
end
%% 3. Walks and turns cutting
function result = getWalkSegments(struct)
preproc = libPreProcessing;
% Convert time data to frames
turn.t1 = round(struct.turn.t1*128); turn.t2 = round(struct.turn.t2*128);
turn.t3 = round(struct.turn.t3*128); turn.t4 = round(struct.turn.t4*128);
turn.t5 = round(struct.turn.t5*128); result = struct;
if struct.turn.Nr == 6; turn.t6 = round(struct.turn.t6*128); end
% 1) Cut every Walk segment
result.T1 = preproc.cutGlobalData(struct, turn.t1);
result.T2 = preproc.cutGlobalData(struct, turn.t2);
result.T3 = preproc.cutGlobalData(struct, turn.t3);
result.T4 = preproc.cutGlobalData(struct, turn.t4);
result.T5 = preproc.cutGlobalData(struct, turn.t5);
result.W1 = preproc.cutGlobalData(struct, [1 turn.t1(1)]);
result.W2 = preproc.cutGlobalData(struct, [turn.t1(2) turn.t2(1)]);
result.W3 = preproc.cutGlobalData(struct, [turn.t2(2) turn.t3(1)]);
result.W4 = preproc.cutGlobalData(struct, [turn.t3(2) turn.t4(1)]);
result.W5 = preproc.cutGlobalData(struct, [turn.t4(2) turn.t5(1)]);
if struct.turn.Nr == 6
    result.W6 = preproc.cutGlobalData(struct, [turn.t5(2) turn.t6]);
else
    result.W6 = preproc.cutGlobalData(struct, [turn.t5(2) struct.Nr]);
end
offset = 3*128;
% 2) Cutting the first (static) and last parts of the Walk
t_ini = detect_motion(result.W1.rfoot.acc(offset:end,:), 7);
if struct.turn.Nr ~= 6; t_end = struct.Nr - detect_motion(flipud(result.W6.rfoot.acc), 7); end
% 3) Save on another struct
result.W1 = preproc.cutGlobalData(struct, [t_ini+offset turn.t1(1)]);
if struct.turn.Nr ~= 6; result.W6 = preproc.cutGlobalData(struct, [turn.t5(2) t_end]); end
% Add name (for titles)
result.T1.name = "Turn Nr.1"; result.T2.name = "Turn Nr.2";
result.T3.name = "Turn Nr.3"; result.T4.name = "Turn Nr.4";
result.T5.name = "Turn Nr.5";
result.W1.name = "Walk Nr.1"; result.W2.name = "Walk Nr.2";
```

```
result.W3.name = "Walk Nr.3"; result.W4.name = "Walk Nr.4";
result.W5.name = "Walk Nr.5"; result.W6.name = "Walk Nr.6";
% Calculate number of samples
result.T1.Nr = length(result.T1.time); result.T2.Nr = length(result.T2.time);
result.T3.Nr = length(result.T3.time); result.T4.Nr = length(result.T4.time);
result.T5.Nr = length(result.T5.time);
result.W1.Nr = length(result.W1.time); result.W2.Nr = length(result.W2.time);
result.W3.Nr = length(result.W3.time); result.W4.Nr = length(result.W4.time);
result.W5.Nr = length(result.W5.time); result.W6.Nr = length(result.W6.time);
end
function idx = detect_motion(acc, thres)
% RMS acceleration
rms = norm([acc(:,1) acc(:,2) acc(:,3)]);
% Find the walking start point
idx = find(rms>thres, 1, 'first');
end

%% IV. Gait cycles management
%% 1. Gait cycle detection
% Method 1: Get HS through drift correction
function struct = driftCorrectionPerDataset(struct, filtThreshold)
[struct.W1.IC.time, struct.W1.IC.frame] = driftCorrectionPerWalk(struct.W1, filtThreshold);
[struct.W2.IC.time, struct.W2.IC.frame] = driftCorrectionPerWalk(struct.W2, filtThreshold);
[struct.W3.IC.time, struct.W3.IC.frame] = driftCorrectionPerWalk(struct.W3, filtThreshold);
[struct.W4.IC.time, struct.W4.IC.frame] = driftCorrectionPerWalk(struct.W4, filtThreshold);
[struct.W5.IC.time, struct.W5.IC.frame] = driftCorrectionPerWalk(struct.W5, filtThreshold);
[struct.W6.IC.time, struct.W6.IC.frame] = driftCorrectionPerWalk(struct.W6, filtThreshold);
end
function [timeHS, framHS] = driftCorrectionPerWalk(W, filtThreshold)
HS = W.HS; % Input data
acc = W.sternum.acc(HS(1):HS(end),3); time = W.time(HS(1):HS(end));
% Velocity a displacement data
vel = cumtrapz(time, acc); pos = cumtrapz(time, vel);
% Initial and final points
idxP1 = 1; idxP2 = length(pos);
P1 = [time(idxP1), pos(idxP1)];
P2 = [time(idxP2), pos(idxP2)];
slope = (P2(2) - P1(2))/(P2(1) - P1(1));
intercept = P1(2) - slope*P1(1);
drift = @(t) slope*t + intercept;
% Correct position
driftCorrection = drift(time);
posCorrected = pos - driftCorrection';
posCorrFiltered = bandpass(posCorrected,[1 filtThreshold],128);
% Differentiate
velCorrected = diff(posCorrFiltered)./diff(time');
accCorrected = diff(velCorrected)./diff(time(1:end-1)');
% Get HS
[~,timeHS] = findpeaks(accCorrected,time(3:end)','MinPeakHeight',0.25); % HS in seconds
[~,framHS] = findpeaks(accCorrected,'MinPeakHeight',0.25); % HS in frames
% Get TS
[~,timeTS] = findpeaks(-accCorrected,time(3:end)','MinPeakHeight',0.1); % TS in seconds
end
% Method 2: Following Digo et al.(2020) using CWT
function dataset = computeGaitEventsPerDataset(dataset, GD)
% Input:
% - dataset: pre, post, td, td1, td2
% - GD: struct of Gait Detection parameters:
%   + HSTS: array [cwtHS, intHS, cwtTS, intTS] (4x1)
%   + minHS: parameter minDistances
%   + rangeThres: parameter of thresolds of range
[dataset.W1.HS, dataset.W1.TS] = computeGaitEventsDetection(dataset.W1, GD.HSTS, GD.minHS,
GD.rangeThres);
[dataset.W2.HS, dataset.W2.TS] = computeGaitEventsDetection(dataset.W2, GD.HSTS, GD.minHS,
GD.rangeThres);
[dataset.W3.HS, dataset.W3.TS] = computeGaitEventsDetection(dataset.W3, GD.HSTS, GD.minHS,
GD.rangeThres);
```

```
[dataset.W4.HS, dataset.W4.TS] = computeGaitEventsDetection(dataset.W4, GD.HSTS, GD.minHS,
GD.rangeThres);
[dataset.W5.HS, dataset.W5.TS] = computeGaitEventsDetection(dataset.W5, GD.HSTS, GD.minHS,
GD.rangeThres);
[dataset.W6.HS, dataset.W6.TS] = computeGaitEventsDetection(dataset.W6, GD.HSTS, GD.minHS,
GD.rangeThres);
end
function [HS, TS] = computeGaitEventsDetection(W, CWT, minDist, threshold)
cwtHS = CWT(1); intHS = CWT(2); cwtTS = CWT(3); intTS = CWT(4);
freqLims = [0 20]; fs = 128;
detHS = getIC(W.time, W.lumbar.acc(:,1), fs, freqLims, cwtHS, intHS);
detTS = getFF(W.time, W.lumbar.acc(:,1), fs, freqLims, cwtTS, intTS);
% Initilalizing arrays
HS.Time = []; TS.Time = []; lastMaxTime = -Inf; minRange = 0.02;
maxRange = minRange + threshold;
% For every HS detected
for i = 1:length(detHS)
    % Identify if the variation between HS is between the range
    if detHS(i) - lastMaxTime >= minDist
        % Find the proximal TS
        nextMin = detTS > detHS(i);
        if any(nextMin)
            minIdx = find(nextMin, 1);
            diffHSTS = detTS(minIdx) - detHS(i);
            % If the difference between the HS and TS is in the range
            if (diffHSTS >= minRange) && (diffHSTS <= maxRange)
                HS.Time = [HS.Time, detHS(i)];
                TS.Time = [TS.Time, detTS(minIdx)];
                lastMaxTime = detHS(i);
            end
        end
    end
end
HS.Frames = ceil((HS.Time - W.time(1))*128);
TS.Frames = ceil((TS.Time - W.time(1))*128);
end
% Get Initial Contact (Heel-strike event)
function HS = getIC(t, sig, fs, lim, min_cwt, max_int)
% CWT calculation ("Morlet" wavelet) from the studied signal
[cfs, frq] = cwt(sig, fs, 'amor', 'FrequencyLimits', lim);
% Integration of CWT coefficients
int_cfs = cumtrapz(frq, real(cfs), 1);
% Getting the signal in time domain
intSig = icwt(int_cfs,'amor');
% Finding local maxima and minima
[~, max_cfs_locs] = findpeaks(sig, t, 'MinPeakHeight', min_cwt);
[~, min_int_locs] = findpeaks(-intSig, t, 'MinPeakHeight', max_int);
HS = [];
for j=1:length(max_cfs_locs)
    for k=1:length(min_int_locs)
        if abs(min_int_locs(k)-max_cfs_locs(j)) <= 0.05 % minimum time between CWT and
integral of CWT detection
            HS = [HS; max_cfs_locs(j)]; break;
        end
    end
end
HS = unique(HS);
end
% Get Footflat (Toe-strike event)
function TS = getFF(t, sig, fs, lim, min_cwt, max_int)
% CWT calculation ("Morlet" wavelet) from the studied signal
[cfs, frq] = cwt(sig, fs, 'amor', 'FrequencyLimits', lim);
% Integration of CWT coefficients
int_cfs = cumtrapz(frq, real(cfs), 1);
% Getting the signal in time domain
intSig = icwt(int_cfs,'amor');
% Finding local maxima and minima
[~, min_cfs_locs] = findpeaks(-sig, t, 'MinPeakHeight', min_cwt);
```

```
[~, max_int_locs] = findpeaks(intSig, t, 'MinPeakHeight', max_int);
TS = [];
for j=1:length(min_cfs_locs)
    for k=1:length(max_int_locs)
        if abs(max_int_locs(k)-min_cfs_locs(j))<= 0.05
            TS = [TS; min_cfs_locs(j)]; break;
        end
    end
end
TS = unique(TS);
end
%% 2. Cut GCs for all data in a dataset
function struct = cutGaitCycleDataset(struct)
struct.W1 = cutGaitCyclesPerWalk(struct.W1);
struct.W2 = cutGaitCyclesPerWalk(struct.W2);
struct.W3 = cutGaitCyclesPerWalk(struct.W3);
struct.W4 = cutGaitCyclesPerWalk(struct.W4);
struct.W5 = cutGaitCyclesPerWalk(struct.W5);
struct.W6 = cutGaitCyclesPerWalk(struct.W6);
end
function Walk = cutGaitCyclesPerWalk(struct)
preproc = libPreProcessing;
Walk = struct; HS = struct.HS.Frames;
nrGC = floor((length(HS)-1)/2);
Walk.GC1 = preproc.cutGCData(struct, [HS(1) HS(3)]);
Walk.GC2 = preproc.cutGCData(struct, [HS(3) HS(5)]);
Walk.GC1.name = "GC1"; Walk.GC2.name = "GC2";
switch nrGC
    case 3
        Walk.GC3 = preproc.cutGCData(struct, [HS(5) HS(7)]); Walk.GC3.name = "GC3";
    case 4
        Walk.GC3 = preproc.cutGCData(struct, [HS(5) HS(7)]);
        Walk.GC4 = preproc.cutGCData(struct, [HS(7) HS(9)]);
        Walk.GC3.name = "GC3"; Walk.GC4.name = "GC4";
    case 5
        Walk.GC3 = preproc.cutGCData(struct, [HS(5) HS(7)]);
        Walk.GC4 = preproc.cutGCData(struct, [HS(7) HS(9)]);
        Walk.GC5 = preproc.cutGCData(struct, [HS(9) HS(11)]);
        Walk.GC3.name = "GC3"; Walk.GC4.name = "GC4"; Walk.GC5.name = "GC5";
    case 6
        Walk.GC3 = preproc.cutGCData(struct, [HS(5) HS(7)]);
        Walk.GC4 = preproc.cutGCData(struct, [HS(7) HS(9)]);
        Walk.GC5 = preproc.cutGCData(struct, [HS(9) HS(11)]);
        Walk.GC6 = preproc.cutGCData(struct, [HS(11) HS(13)]);
        Walk.GC3.name = "GC3"; Walk.GC4.name = "GC4"; Walk.GC5.name = "GC5";
        Walk.GC6.name = "GC6";
    case 7
        Walk.GC3 = preproc.cutGCData(struct, [HS(5) HS(7)]);
        Walk.GC4 = preproc.cutGCData(struct, [HS(7) HS(9)]);
        Walk.GC5 = preproc.cutGCData(struct, [HS(9) HS(11)]);
        Walk.GC6 = preproc.cutGCData(struct, [HS(11) HS(13)]);
        Walk.GC7 = preproc.cutGCData(struct, [HS(13) HS(15)]);
        Walk.GC3.name = "GC3"; Walk.GC4.name = "GC4"; Walk.GC5.name = "GC5";
        Walk.GC6.name = "GC6"; Walk.GC7.name = "GC7";
    case 8
        Walk.GC3 = preproc.cutGCData(struct, [HS(5) HS(7)]);
        Walk.GC4 = preproc.cutGCData(struct, [HS(7) HS(9)]);
        Walk.GC5 = preproc.cutGCData(struct, [HS(9) HS(11)]);
        Walk.GC6 = preproc.cutGCData(struct, [HS(11) HS(13)]);
        Walk.GC7 = preproc.cutGCData(struct, [HS(13) HS(15)]);
        Walk.GC8 = preproc.cutGCData(struct, [HS(15) HS(17)]);
        Walk.GC3.name = "GC3"; Walk.GC4.name = "GC4"; Walk.GC5.name = "GC5";
        Walk.GC6.name = "GC6"; Walk.GC7.name = "GC7"; Walk.GC8.name = "GC8";
    case 9
        Walk.GC3 = preproc.cutGCData(struct, [HS(5) HS(7)]);
        Walk.GC4 = preproc.cutGCData(struct, [HS(7) HS(9)]);
        Walk.GC5 = preproc.cutGCData(struct, [HS(9) HS(11)]);
        Walk.GC6 = preproc.cutGCData(struct, [HS(11) HS(13)]);
```

```
        Walk.GC7 = preproc.cutGCData(struct, [HS(13) HS(15)]);
        Walk.GC8 = preproc.cutGCData(struct, [HS(15) HS(17)]);
        Walk.GC9 = preproc.cutGCData(struct, [HS(17) HS(19)]);
        Walk.GC3.name = "GC3"; Walk.GC4.name = "GC4"; Walk.GC5.name = "GC5"; Walk.GC6.name =
"GC6";
        Walk.GC7.name = "GC7"; Walk.GC8.name = "GC8"; Walk.GC9.name = "GC9";
    case 10
        Walk.GC3 = preproc.cutGCData(struct, [HS(5) HS(7)]);
        Walk.GC4 = preproc.cutGCData(struct, [HS(7) HS(9)]);
        Walk.GC5 = preproc.cutGCData(struct, [HS(9) HS(11)]);
        Walk.GC6 = preproc.cutGCData(struct, [HS(11) HS(13)]);
        Walk.GC7 = preproc.cutGCData(struct, [HS(13) HS(15)]);
        Walk.GC8 = preproc.cutGCData(struct, [HS(15) HS(17)]);
        Walk.GC9 = preproc.cutGCData(struct, [HS(17) HS(19)]);
        Walk.GC10 = preproc.cutGCData(struct, [HS(19) HS(21)]);
        Walk.GC3.name = "GC3"; Walk.GC4.name = "GC4"; Walk.GC5.name = "GC5";
        Walk.GC6.name = "GC6"; Walk.GC7.name = "GC7"; Walk.GC8.name = "GC8";
        Walk.GC9.name = "GC9"; Walk.GC10.name = "GC10";
end
Walk.nrGC = nrGC;
end
```

G.4 Parameters calculation library

```
%% Parameters calculation functions library
% Author: Maria Alejandra Guzman Alfaro
% Last updated: 01.11.2024
function param = libParameters
param.getSpatiotemporalParametersPerDataset=@getSpatiotemporalParametersPerDataset;
param.saveSpatiotemporalPerDataset=@saveSpatiotemporalPerDataset;
param.getRMSPerDataset=@getRMSPerDataset;
param.getHRPerDataset=@getHRPerDataset;
param.getACPerDataset=@getACPerDataset;
param.getAbsoluteAnglesPerDataset=@getAbsoluteAnglesPerDataset;
param.getRelativeAnglesPerDataset=@getRelativeAnglesPerDataset;
end

%% 1. Spatiotemporal parameters
function struct = getSpatiotemporalParametersPerDataset(struct)
struct.W1 = getSpatiotemporalPerWalk(struct.W1);
struct.W2 = getSpatiotemporalPerWalk(struct.W2);
struct.W3 = getSpatiotemporalPerWalk(struct.W3);
struct.W4 = getSpatiotemporalPerWalk(struct.W4);
struct.W5 = getSpatiotemporalPerWalk(struct.W5);
struct.W6 = getSpatiotemporalPerWalk(struct.W6);
% Mean values per walk
struct.speed = [struct.W1.speed; struct.W2.speed; struct.W3.speed; struct.W4.speed;
struct.W5.speed; struct.W6.speed];
struct.cad.stride = [struct.W1.cad.stride; struct.W2.cad.stride; struct.W3.cad.stride;
struct.W4.cad.stride; struct.W5.cad.stride; struct.W6.cad.stride];
struct.cad.step = [struct.W1.cad.step; struct.W2.cad.step; struct.W3.cad.step;
struct.W4.cad.step; struct.W5.cad.step; struct.W6.cad.step];
struct.length.stride = [struct.W1.length.stride; struct.W2.length.stride;
struct.W3.length.stride; struct.W4.length.stride; struct.W5.length.stride;
struct.W6.length.stride];
struct.length.step = [struct.W1.length.step; struct.W2.length.step; struct.W3.length.step;
struct.W4.length.step; struct.W5.length.step; struct.W6.length.step];
struct.nrStride = [struct.W1.nrStride,struct.W2.nrStride,struct.W3.nrStride,
struct.W4.nrStride,struct.W5.nrStride,struct.W6.nrStride];
struct.nrStep = [struct.W1.nrStep,struct.W2.nrStep,struct.W3.nrStep,
struct.W4.nrStep,struct.W5.nrStep,struct.W6.nrStep];
struct.time.stride = [struct.W1.tst.stride; struct.W2.tst.stride; struct.W3.tst.stride;
struct.W4.tst.stride; struct.W5.tst.stride; struct.W6.tst.stride];
struct.time.step = [struct.W1.tst.step; struct.W2.tst.step; struct.W3.tst.step;
struct.W4.tst.step; struct.W5.tst.step; struct.W6.tst.step];
struct.tstride = [mean(struct.W1.tst.stride); mean(struct.W2.tst.stride);
mean(struct.W3.tst.stride); mean(struct.W4.tst.stride); mean(struct.W5.tst.stride);
mean(struct.W6.tst.stride)];
struct.tstep = [mean(struct.W1.tst.step); mean(struct.W2.tst.step); mean(struct.W3.tst.step);
mean(struct.W4.tst.step); mean(struct.W5.tst.step); mean(struct.W6.tst.step)];
% Mean values per dataset
struct.avg.speed = mean(struct.speed);
struct.avg.cad.stride = mean(struct.cad.stride); struct.avg.cad.step = mean(struct.cad.step);
struct.avg.length.stride = mean(struct.length.stride); struct.avg.length.step =
mean(struct.length.step);
struct.avg.nrStride = mean(struct.nrStride); struct.avg.nrStep = mean(struct.nrStep);
struct.avg.time.stride = mean(struct.time.stride); struct.avg.time.step =
mean(struct.time.step);
struct.total.nrStep = sum(struct.nrStep); struct.total.nrStride = sum(struct.nrStride);
end
function walk = getSpatiotemporalPerWalk(walk)
% Get temporal parameters per Gait Cycle:
walk.GC1 = getTemporalParametersPerGC(walk.GC1);
walk.GC2 = getTemporalParametersPerGC(walk.GC2);
walk.tst.step = [walk.GC1.temp.step; walk.GC2.temp.step];
walk.tst.stride = [walk.GC1.temp.stride; walk.GC2.temp.stride];
switch walk.nrGC
    case 3
        walk.GC3 = getTemporalParametersPerGC(walk.GC3);
        walk.tst.step = [walk.tst.step; walk.GC3.temp.step];
```

```
    walk.tst.stride = [walk.tst.stride; walk.GC3.temp.stride];
case 4
    walk.GC3 = getTemporalParametersPerGC(walk.GC3);
    walk.GC4 = getTemporalParametersPerGC(walk.GC4);
    walk.tst.step = [walk.tst.step; walk.GC3.temp.step; walk.GC4.temp.step];
    walk.tst.stride = [walk.tst.stride; walk.GC3.temp.stride; walk.GC4.temp.stride];
case 5
    walk.GC3 = getTemporalParametersPerGC(walk.GC3);
    walk.GC4 = getTemporalParametersPerGC(walk.GC4);
    walk.GC5 = getTemporalParametersPerGC(walk.GC5);
    walk.tst.step = [walk.tst.step; walk.GC3.temp.step; walk.GC4.temp.step;
        walk.GC5.temp.step];
    walk.tst.stride = [walk.tst.stride; walk.GC3.temp.stride; walk.GC4.temp.stride;
        walk.GC5.temp.stride];
case 6
    walk.GC3 = getTemporalParametersPerGC(walk.GC3);
    walk.GC4 = getTemporalParametersPerGC(walk.GC4);
    walk.GC5 = getTemporalParametersPerGC(walk.GC5);
    walk.GC6 = getTemporalParametersPerGC(walk.GC6);
    walk.tst.step = [walk.tst.step; walk.GC3.temp.step; walk.GC4.temp.step;
        walk.GC5.temp.step; walk.GC6.temp.step];
    walk.tst.stride = [walk.tst.stride; walk.GC3.temp.stride; walk.GC4.temp.stride;
        walk.GC5.temp.stride; walk.GC6.temp.stride];
case 7
    walk.GC3 = getTemporalParametersPerGC(walk.GC3);
    walk.GC4 = getTemporalParametersPerGC(walk.GC4);
    walk.GC5 = getTemporalParametersPerGC(walk.GC5);
    walk.GC6 = getTemporalParametersPerGC(walk.GC6);
    walk.GC7 = getTemporalParametersPerGC(walk.GC7);
    walk.tst.step = [walk.tst.step; walk.GC3.temp.step; walk.GC4.temp.step;
        walk.GC5.temp.step; walk.GC6.temp.step; walk.GC7.temp.step];
    walk.tst.stride = [walk.tst.stride; walk.GC3.temp.stride; walk.GC4.temp.stride;
        walk.GC5.temp.stride; walk.GC6.temp.stride; walk.GC7.temp.stride];
case 8
    walk.GC3 = getTemporalParametersPerGC(walk.GC3);
    walk.GC4 = getTemporalParametersPerGC(walk.GC4);
    walk.GC5 = getTemporalParametersPerGC(walk.GC5);
    walk.GC6 = getTemporalParametersPerGC(walk.GC6);
    walk.GC7 = getTemporalParametersPerGC(walk.GC7);
    walk.GC8 = getTemporalParametersPerGC(walk.GC8);
    walk.tst.step = [walk.tst.step; walk.GC3.temp.step; walk.GC4.temp.step;
        walk.GC5.temp.step; walk.GC6.temp.step; walk.GC7.temp.step;
        walk.GC8.temp.step];
    walk.tst.stride = [walk.tst.stride; walk.GC3.temp.stride; walk.GC4.temp.stride;
        walk.GC5.temp.stride; walk.GC6.temp.stride; walk.GC7.temp.stride;
        walk.GC8.temp.stride];
case 9
    walk.GC3 = getTemporalParametersPerGC(walk.GC3);
    walk.GC4 = getTemporalParametersPerGC(walk.GC4);
    walk.GC5 = getTemporalParametersPerGC(walk.GC5);
    walk.GC6 = getTemporalParametersPerGC(walk.GC6);
    walk.GC7 = getTemporalParametersPerGC(walk.GC7);
    walk.GC8 = getTemporalParametersPerGC(walk.GC8);
    walk.GC9 = getTemporalParametersPerGC(walk.GC9);
    walk.tst.step = [walk.tst.step; walk.GC3.temp.step; walk.GC4.temp.step;
        walk.GC5.temp.step; walk.GC6.temp.step; walk.GC7.temp.step;
        walk.GC8.temp.step; walk.GC9.temp.step];
    walk.tst.stride = [walk.tst.stride; walk.GC3.temp.stride; walk.GC4.temp.stride;
        walk.GC5.temp.stride; walk.GC6.temp.stride; walk.GC7.temp.stride;
        walk.GC8.temp.stride; walk.GC9.temp.stride];
case 10
    walk.GC3 = getTemporalParametersPerGC(walk.GC3);
    walk.GC4 = getTemporalParametersPerGC(walk.GC4);
    walk.GC5 = getTemporalParametersPerGC(walk.GC5);
    walk.GC6 = getTemporalParametersPerGC(walk.GC6);
    walk.GC7 = getTemporalParametersPerGC(walk.GC7);
    walk.GC8 = getTemporalParametersPerGC(walk.GC8);
    walk.GC9 = getTemporalParametersPerGC(walk.GC9);
```

```
        walk.GC10 = getTemporalParametersPerGC(walk.GC10);
        walk.tst.step = [walk.tst.step; walk.GC3.temp.step; walk.GC4.temp.step;
            walk.GC5.temp.step; walk.GC6.temp.step; walk.GC7.temp.step;
            walk.GC8.temp.step; walk.GC9.temp.step; walk.GC10.temp.step];
        walk.tst.stride = [walk.tst.stride; walk.GC3.temp.stride; walk.GC4.temp.stride;
            walk.GC5.temp.stride; walk.GC6.temp.stride; walk.GC7.temp.stride;
            walk.GC8.temp.stride; walk.GC9.temp.stride; walk.GC10.temp.stride];
end

% Get spatial and spatiotemporal parameters:
d = 5; % 5mWT
walk.nrStride = walk.nrGC;
walk.nrStep = walk.nrGC*2; % change later
walk.speed = d/(walk.time(end)-walk.time(1));
walk.cad.stride = walk.nrStride/(walk.time(end)-walk.time(1));
walk.cad.step = walk.nrStep/(walk.time(end)-walk.time(1));
walk.length.stride = d/walk.nrStride;
walk.length.step = d/walk.nrStep;
end
function GC = getTemporalParametersPerGC(gc)
t = gc.time; GC = gc;
% Normalized time
gt = linspace(0,100,length(t));
% Temporal parameters
time.stride = t(end)-t(1); time.step = time.stride/2;
% Saving on GC struct
GC.gt = gt; GC.temp = time;
end
%% 2. Root Mean Square (RMS)
function struct = getRMSPerDataset(struct)
% get RMS per walk
struct.W1 = getRMSAllIMUs(struct.W1); struct.W2 = getRMSAllIMUs(struct.W2);
struct.W3 = getRMSAllIMUs(struct.W3); struct.W4 = getRMSAllIMUs(struct.W4);
struct.W5 = getRMSAllIMUs(struct.W5); struct.W6 = getRMSAllIMUs(struct.W6);
% get RMS per GC in Walk
struct.W1 = getRMSPerGC(struct.W1); struct.W2 = getRMSPerGC(struct.W2);
struct.W3 = getRMSPerGC(struct.W3); struct.W4 = getRMSPerGC(struct.W4);
struct.W5 = getRMSPerGC(struct.W5); struct.W6 = getRMSPerGC(struct.W6);
struct = getMeanRMSPerDataset(struct);
end
function struct = getMeanRMSPerDataset(struct)
% Sternum RMS
struct.RMS.sternum.acc.AP = [struct.W1.RMS.sternum.acc.AP; struct.W2.RMS.sternum.acc.AP;
struct.W3.RMS.sternum.acc.AP;
    struct.W4.RMS.sternum.acc.AP; struct.W5.RMS.sternum.acc.AP; struct.W6.RMS.sternum.acc.AP];
struct.RMS.sternum.acc.ML = [struct.W1.RMS.sternum.acc.ML; struct.W2.RMS.sternum.acc.ML;
struct.W3.RMS.sternum.acc.ML;
    struct.W4.RMS.sternum.acc.ML; struct.W5.RMS.sternum.acc.ML; struct.W6.RMS.sternum.acc.ML];
struct.RMS.sternum.acc.VT = [struct.W1.RMS.sternum.acc.VT; struct.W2.RMS.sternum.acc.VT;
struct.W3.RMS.sternum.acc.VT;
    struct.W4.RMS.sternum.acc.VT; struct.W5.RMS.sternum.acc.VT; struct.W6.RMS.sternum.acc.VT];
struct.RMS.sternum.acc.total = [struct.W1.RMS.sternum.acc.total;
struct.W2.RMS.sternum.acc.total; struct.W3.RMS.sternum.acc.total;
    struct.W4.RMS.sternum.acc.total; struct.W5.RMS.sternum.acc.total;
struct.W6.RMS.sternum.acc.total];
struct.RMS.sternum.acc.all.AP = [struct.W1.RMS.sternum.acc.all.AP',
struct.W2.RMS.sternum.acc.all.AP', ...
    struct.W3.RMS.sternum.acc.all.AP', struct.W4.RMS.sternum.acc.all.AP', ...
    struct.W5.RMS.sternum.acc.all.AP', struct.W6.RMS.sternum.acc.all.AP'];
struct.RMS.sternum.acc.all.ML = [struct.W1.RMS.sternum.acc.all.ML',
struct.W2.RMS.sternum.acc.all.ML', ...
    struct.W3.RMS.sternum.acc.all.ML', struct.W4.RMS.sternum.acc.all.ML', ...
    struct.W5.RMS.sternum.acc.all.ML', struct.W6.RMS.sternum.acc.all.ML'];
struct.RMS.sternum.acc.all.VT = [struct.W1.RMS.sternum.acc.all.VT',
struct.W2.RMS.sternum.acc.all.VT', ...
    struct.W3.RMS.sternum.acc.all.VT', struct.W4.RMS.sternum.acc.all.VT' ...
    struct.W5.RMS.sternum.acc.all.VT', struct.W6.RMS.sternum.acc.all.VT'];
```

```
struct.RMS.sternum.acc.all.total = [struct.W1.RMS.sternum.acc.all.total',
struct.W2.RMS.sternum.acc.all.total', ...
    struct.W3.RMS.sternum.acc.all.total', struct.W4.RMS.sternum.acc.all.total', ...
    struct.W5.RMS.sternum.acc.all.total', struct.W6.RMS.sternum.acc.all.total'];
% Mean RMS
struct.avg.RMS.sternum.acc.AP = mean(struct.RMS.sternum.acc.all.AP);
struct.avg.RMS.sternum.acc.ML = mean(struct.RMS.sternum.acc.all.ML);
struct.avg.RMS.sternum.acc.VT = mean(struct.RMS.sternum.acc.all.VT);
struct.avg.RMS.sternum.acc.total = mean(struct.RMS.sternum.acc.all.total);
% Sternum RMSR
struct.RMSR.sternum.acc.all.AP = [struct.W1.RMSR.sternum.acc.AP',
struct.W2.RMSR.sternum.acc.AP', ...
    struct.W3.RMSR.sternum.acc.AP', struct.W4.RMSR.sternum.acc.AP', ...
    struct.W5.RMSR.sternum.acc.AP', struct.W6.RMSR.sternum.acc.AP'];
struct.RMSR.sternum.acc.all.ML = [struct.W1.RMSR.sternum.acc.ML',
struct.W2.RMSR.sternum.acc.ML', ...
    struct.W3.RMSR.sternum.acc.ML', struct.W4.RMSR.sternum.acc.ML', ...
    struct.W5.RMSR.sternum.acc.ML', struct.W6.RMSR.sternum.acc.ML'];
struct.RMSR.sternum.acc.all.VT = [struct.W1.RMSR.sternum.acc.VT',
struct.W2.RMSR.sternum.acc.VT', ...
    struct.W3.RMSR.sternum.acc.VT', struct.W4.RMSR.sternum.acc.VT', ...
    struct.W5.RMSR.sternum.acc.VT', struct.W6.RMSR.sternum.acc.VT'];
% Mean RMSR
struct.avg.RMSR.sternum.acc.AP = mean(struct.RMSR.sternum.acc.all.AP);
struct.avg.RMSR.sternum.acc.ML = mean(struct.RMSR.sternum.acc.all.ML);
struct.avg.RMSR.sternum.acc.VT = mean(struct.RMSR.sternum.acc.all.VT);
% Lumbar RMS
struct.RMS.lumbar.acc.AP = [struct.W1.RMS.lumbar.acc.AP; struct.W2.RMS.lumbar.acc.AP;
struct.W3.RMS.lumbar.acc.AP;
    struct.W4.RMS.lumbar.acc.AP; struct.W5.RMS.lumbar.acc.AP; struct.W6.RMS.lumbar.acc.AP];
struct.RMS.lumbar.acc.ML = [struct.W1.RMS.lumbar.acc.ML; struct.W2.RMS.lumbar.acc.ML;
struct.W3.RMS.lumbar.acc.ML;
    struct.W4.RMS.lumbar.acc.ML; struct.W5.RMS.lumbar.acc.ML; struct.W6.RMS.lumbar.acc.ML];
struct.RMS.lumbar.acc.VT = [struct.W1.RMS.lumbar.acc.VT; struct.W2.RMS.lumbar.acc.VT;
struct.W3.RMS.lumbar.acc.VT;
    struct.W4.RMS.lumbar.acc.VT; struct.W5.RMS.lumbar.acc.VT; struct.W6.RMS.lumbar.acc.VT];
struct.RMS.lumbar.acc.total = [struct.W1.RMS.lumbar.acc.total; struct.W2.RMS.lumbar.acc.total;
struct.W3.RMS.lumbar.acc.total;
    struct.W4.RMS.lumbar.acc.total; struct.W5.RMS.lumbar.acc.total;
struct.W6.RMS.lumbar.acc.total];
struct.RMS.lumbar.acc.all.AP = [struct.W1.RMS.lumbar.acc.all.AP',
struct.W2.RMS.lumbar.acc.all.AP', ...
    struct.W3.RMS.lumbar.acc.all.AP', struct.W4.RMS.lumbar.acc.all.AP', ...
    struct.W5.RMS.lumbar.acc.all.AP', struct.W6.RMS.lumbar.acc.all.AP'];
struct.RMS.lumbar.acc.all.ML = [struct.W1.RMS.lumbar.acc.all.ML',
struct.W2.RMS.lumbar.acc.all.ML', ...
    struct.W3.RMS.lumbar.acc.all.ML', struct.W4.RMS.lumbar.acc.all.ML', ...
    struct.W5.RMS.lumbar.acc.all.ML', struct.W6.RMS.lumbar.acc.all.ML'];
struct.RMS.lumbar.acc.all.VT = [struct.W1.RMS.lumbar.acc.all.VT',
struct.W2.RMS.lumbar.acc.all.VT', ...
    struct.W3.RMS.lumbar.acc.all.VT', struct.W4.RMS.lumbar.acc.all.VT', ...
    struct.W5.RMS.lumbar.acc.all.VT', struct.W6.RMS.lumbar.acc.all.VT'];
struct.RMS.lumbar.acc.all.total = [struct.W1.RMS.lumbar.acc.all.total',
struct.W2.RMS.lumbar.acc.all.total', ...
    struct.W3.RMS.lumbar.acc.all.total', struct.W4.RMS.lumbar.acc.all.total', ...
    struct.W5.RMS.lumbar.acc.all.total', struct.W6.RMS.lumbar.acc.all.total'];
% Mean RMS
struct.avg.RMS.lumbar.acc.AP = mean(struct.RMS.lumbar.acc.all.AP);
struct.avg.RMS.lumbar.acc.ML = mean(struct.RMS.lumbar.acc.all.ML);
struct.avg.RMS.lumbar.acc.VT = mean(struct.RMS.lumbar.acc.all.VT);
struct.avg.RMS.lumbar.acc.total = mean(struct.RMS.lumbar.acc.all.total);
% Sternum RMSR
struct.RMSR.lumbar.acc.all.AP = [struct.W1.RMSR.lumbar.acc.AP', struct.W2.RMSR.lumbar.acc.AP',
...
    struct.W3.RMSR.lumbar.acc.AP', struct.W4.RMSR.lumbar.acc.AP' ...
    struct.W5.RMSR.lumbar.acc.AP', struct.W6.RMSR.lumbar.acc.AP'];
struct.RMSR.lumbar.acc.all.ML = [struct.W1.RMSR.lumbar.acc.ML', struct.W2.RMSR.lumbar.acc.ML',
...
```

```
    struct.W3.RMSR.lumbar.acc.ML', struct.W4.RMSR.lumbar.acc.ML' ...
    struct.W5.RMSR.lumbar.acc.ML', struct.W6.RMSR.lumbar.acc.ML'];
struct.RMSR.lumbar.acc.all.VT = [struct.W1.RMSR.lumbar.acc.VT', struct.W2.RMSR.lumbar.acc.VT',
...
    struct.W3.RMSR.lumbar.acc.VT', struct.W4.RMSR.lumbar.acc.VT' ...
    struct.W5.RMSR.lumbar.acc.VT', struct.W6.RMSR.lumbar.acc.VT'];
% Mean RMSR
struct.avg.RMSR.lumbar.acc.AP = mean(struct.RMSR.lumbar.acc.all.AP);
struct.avg.RMSR.lumbar.acc.ML = mean(struct.RMSR.lumbar.acc.all.ML);
struct.avg.RMSR.lumbar.acc.VT = mean(struct.RMSR.lumbar.acc.all.VT);
end
function struct = getRMSAllIMUs(struct)
% Get RMS per Walk
[struct.RMS.sternum.acc, struct.RMSR.sternum.acc] = getRMS(struct.sternum.acc);
[struct.RMS.sternum.gyr, struct.RMSR.sternum.gyr] = getRMS(struct.sternum.gyr);
[struct.RMS.lumbar.acc, struct.RMSR.lumbar.acc] = getRMS(struct.lumbar.acc);
[struct.RMS.lumbar.gyr, struct.RMSR.lumbar.gyr] = getRMS(struct.lumbar.gyr);
[struct.RMS.rfoot.acc, struct.RMSR.rfoot.acc] = getRMS(struct.rfoot.acc);
[struct.RMS.rfoot.gyr, struct.RMSR.rfoot.gyr] = getRMS(struct.rfoot.gyr);
[struct.RMS.lfoot.acc, struct.RMSR.lfoot.acc] = getRMS(struct.lfoot.acc);
[struct.RMS.lfoot.gyr, struct.RMSR.lfoot.gyr] = getRMS(struct.lfoot.gyr);
end
function walk = getRMSPerGC(walk)
% Get RMS per Gait Cycle:
walk.GC1 = getRMSAllIMUs(walk.GC1);
walk.GC2 = getRMSAllIMUs(walk.GC2);
walk.RMS.sternum.acc.all.AP = [walk.GC1.RMS.sternum.acc.AP; walk.GC2.RMS.sternum.acc.AP];
walk.RMS.sternum.acc.all.ML = [walk.GC1.RMS.sternum.acc.ML; walk.GC2.RMS.sternum.acc.ML];
walk.RMS.sternum.acc.all.VT = [walk.GC1.RMS.sternum.acc.VT; walk.GC2.RMS.sternum.acc.VT];
walk.RMS.sternum.acc.all.total = [walk.GC1.RMS.sternum.acc.total;
walk.GC2.RMS.sternum.acc.total];
walk.RMS.lumbar.acc.all.AP = [walk.GC1.RMS.lumbar.acc.AP; walk.GC2.RMS.lumbar.acc.AP];
walk.RMS.lumbar.acc.all.ML = [walk.GC1.RMS.lumbar.acc.ML; walk.GC2.RMS.lumbar.acc.ML];
walk.RMS.lumbar.acc.all.VT = [walk.GC1.RMS.lumbar.acc.VT; walk.GC2.RMS.lumbar.acc.VT];
walk.RMS.lumbar.acc.all.total = [walk.GC1.RMS.lumbar.acc.total;
walk.GC2.RMS.lumbar.acc.total];
switch walk.nrGC
    case 3
        walk.GC3 = getRMSAllIMUs(walk.GC3);
        walk.RMS.sternum.acc.all.AP = [walk.RMS.sternum.acc.all.AP;
walk.GC3.RMS.sternum.acc.AP];
        walk.RMS.sternum.acc.all.ML = [walk.RMS.sternum.acc.all.ML;
walk.GC3.RMS.sternum.acc.ML];
        walk.RMS.sternum.acc.all.VT = [walk.RMS.sternum.acc.all.VT;
walk.GC3.RMS.sternum.acc.VT];
        walk.RMS.sternum.acc.all.total = [walk.RMS.sternum.acc.all.total;
walk.GC3.RMS.sternum.acc.total];
        walk.RMS.lumbar.acc.all.AP = [walk.RMS.sternum.acc.all.AP;
walk.GC3.RMS.lumbar.acc.AP];
        walk.RMS.lumbar.acc.all.ML = [walk.RMS.sternum.acc.all.ML;
walk.GC3.RMS.lumbar.acc.ML];
        walk.RMS.lumbar.acc.all.VT = [walk.RMS.sternum.acc.all.VT;
walk.GC3.RMS.lumbar.acc.VT];
        walk.RMS.lumbar.acc.all.total = [walk.RMS.sternum.acc.all.total;
walk.GC3.RMS.lumbar.acc.total];
    case 4
        walk.GC3 = getRMSAllIMUs(walk.GC3); walk.GC4 = getRMSAllIMUs(walk.GC4);
        walk.RMS.sternum.acc.all.AP = [walk.RMS.sternum.acc.all.AP;
walk.GC3.RMS.sternum.acc.AP;
            walk.GC4.RMS.sternum.acc.AP];
        walk.RMS.sternum.acc.all.ML = [walk.RMS.sternum.acc.all.ML;
walk.GC3.RMS.sternum.acc.ML;
            walk.GC4.RMS.sternum.acc.ML];
        walk.RMS.sternum.acc.all.VT = [walk.RMS.sternum.acc.all.VT;
walk.GC3.RMS.sternum.acc.VT;
            walk.GC4.RMS.sternum.acc.VT];
        walk.RMS.sternum.acc.all.total = [walk.RMS.sternum.acc.all.total;
walk.GC3.RMS.sternum.acc.total;
```

```
            walk.GC4.RMS.sternum.acc.total];
        walk.RMS.lumbar.acc.all.AP = [walk.RMS.lumbar.acc.all.AP; walk.GC3.RMS.lumbar.acc.AP;
            walk.GC4.RMS.lumbar.acc.AP];
        walk.RMS.lumbar.acc.all.ML = [walk.RMS.lumbar.acc.all.ML; walk.GC3.RMS.lumbar.acc.ML;
            walk.GC4.RMS.lumbar.acc.ML];
        walk.RMS.lumbar.acc.all.VT = [walk.RMS.lumbar.acc.all.VT; walk.GC3.RMS.lumbar.acc.VT;
            walk.GC4.RMS.lumbar.acc.VT];
        walk.RMS.lumbar.acc.all.total = [walk.RMS.lumbar.acc.all.total;
walk.GC3.RMS.lumbar.acc.total;
            walk.GC4.RMS.lumbar.acc.total];
    case 5
        walk.GC3 = getRMSAllIMUs(walk.GC3); walk.GC4 = getRMSAllIMUs(walk.GC4);
        walk.GC5 = getRMSAllIMUs(walk.GC5);
        walk.RMS.sternum.acc.all.AP = [walk.RMS.sternum.acc.all.AP;
walk.GC3.RMS.sternum.acc.AP;
            walk.GC4.RMS.sternum.acc.AP; walk.GC5.RMS.sternum.acc.AP];
        walk.RMS.sternum.acc.all.ML = [walk.RMS.sternum.acc.all.ML;
walk.GC3.RMS.sternum.acc.ML;
            walk.GC4.RMS.sternum.acc.ML; walk.GC5.RMS.sternum.acc.ML];
        walk.RMS.sternum.acc.all.VT = [walk.RMS.sternum.acc.all.VT;
walk.GC3.RMS.sternum.acc.VT;
            walk.GC4.RMS.sternum.acc.VT; walk.GC5.RMS.sternum.acc.VT];
        walk.RMS.sternum.acc.all.total = [walk.RMS.sternum.acc.all.total;
walk.GC3.RMS.sternum.acc.total;
            walk.GC4.RMS.sternum.acc.total; walk.GC5.RMS.sternum.acc.total];
        walk.RMS.lumbar.acc.all.AP = [walk.RMS.lumbar.acc.all.AP; walk.GC3.RMS.lumbar.acc.AP;
            walk.GC4.RMS.lumbar.acc.AP; walk.GC5.RMS.lumbar.acc.AP];
        walk.RMS.lumbar.acc.all.ML = [walk.RMS.lumbar.acc.all.ML; walk.GC3.RMS.lumbar.acc.ML;
            walk.GC4.RMS.lumbar.acc.ML; walk.GC5.RMS.lumbar.acc.ML];
        walk.RMS.lumbar.acc.all.VT = [walk.RMS.lumbar.acc.all.VT; walk.GC3.RMS.lumbar.acc.VT;
            walk.GC4.RMS.lumbar.acc.VT; walk.GC5.RMS.lumbar.acc.VT];
        walk.RMS.lumbar.acc.all.total = [walk.RMS.lumbar.acc.all.total;
walk.GC3.RMS.lumbar.acc.total;
            walk.GC4.RMS.lumbar.acc.total; walk.GC5.RMS.lumbar.acc.total];
    case 6
        walk.GC3 = getRMSAllIMUs(walk.GC3); walk.GC4 = getRMSAllIMUs(walk.GC4);
        walk.GC5 = getRMSAllIMUs(walk.GC5); walk.GC6 = getRMSAllIMUs(walk.GC6);
        walk.RMS.sternum.acc.all.AP = [walk.RMS.sternum.acc.all.AP;
walk.GC3.RMS.sternum.acc.AP;
            walk.GC4.RMS.sternum.acc.AP; walk.GC5.RMS.sternum.acc.AP;
            walk.GC6.RMS.sternum.acc.AP];
        walk.RMS.sternum.acc.all.ML = [walk.RMS.sternum.acc.all.ML;
walk.GC3.RMS.sternum.acc.ML;
            walk.GC4.RMS.sternum.acc.ML; walk.GC5.RMS.sternum.acc.ML;
            walk.GC6.RMS.sternum.acc.ML];
        walk.RMS.sternum.acc.all.VT = [walk.RMS.sternum.acc.all.VT;
walk.GC3.RMS.sternum.acc.VT;
            walk.GC4.RMS.sternum.acc.VT; walk.GC5.RMS.sternum.acc.VT;
            walk.GC6.RMS.sternum.acc.VT];
        walk.RMS.sternum.acc.all.total = [walk.RMS.sternum.acc.all.total;
walk.GC3.RMS.sternum.acc.total;
            walk.GC4.RMS.sternum.acc.total; walk.GC5.RMS.sternum.acc.total;
            walk.GC6.RMS.sternum.acc.total];
        walk.RMS.lumbar.acc.all.AP = [walk.RMS.lumbar.acc.all.AP; walk.GC3.RMS.lumbar.acc.AP;
            walk.GC4.RMS.lumbar.acc.AP; walk.GC5.RMS.lumbar.acc.AP;
            walk.GC6.RMS.lumbar.acc.AP];
        walk.RMS.lumbar.acc.all.ML = [walk.RMS.lumbar.acc.all.ML; walk.GC3.RMS.lumbar.acc.ML;
            walk.GC4.RMS.lumbar.acc.ML; walk.GC5.RMS.lumbar.acc.ML;
            walk.GC6.RMS.lumbar.acc.ML];
        walk.RMS.lumbar.acc.all.VT = [walk.RMS.lumbar.acc.all.VT; walk.GC3.RMS.lumbar.acc.VT;
            walk.GC4.RMS.lumbar.acc.VT; walk.GC5.RMS.lumbar.acc.VT;
            walk.GC6.RMS.lumbar.acc.VT];
        walk.RMS.lumbar.acc.all.total = [walk.RMS.lumbar.acc.all.total;
walk.GC3.RMS.lumbar.acc.total;
            walk.GC4.RMS.lumbar.acc.total; walk.GC5.RMS.lumbar.acc.total;
            walk.GC6.RMS.lumbar.acc.total];
    case 7
        walk.GC3 = getRMSAllIMUs(walk.GC3); walk.GC4 = getRMSAllIMUs(walk.GC4);
```

```
        walk.GC5 = getRMSAllIMUs(walk.GC5); walk.GC6 = getRMSAllIMUs(walk.GC6);
        walk.GC7 = getRMSAllIMUs(walk.GC7);
        walk.RMS.sternum.acc.all.AP = [walk.RMS.sternum.acc.all.AP;
walk.GC3.RMS.sternum.acc.AP;
            walk.GC4.RMS.sternum.acc.AP; walk.GC5.RMS.sternum.acc.AP;
            walk.GC6.RMS.sternum.acc.AP; walk.GC7.RMS.sternum.acc.AP];
        walk.RMS.sternum.acc.all.ML = [walk.RMS.sternum.acc.all.ML;
walk.GC3.RMS.sternum.acc.ML;
            walk.GC4.RMS.sternum.acc.ML; walk.GC5.RMS.sternum.acc.ML;
            walk.GC6.RMS.sternum.acc.ML; walk.GC7.RMS.sternum.acc.ML];
        walk.RMS.sternum.acc.all.VT = [walk.RMS.sternum.acc.all.VT;
walk.GC3.RMS.sternum.acc.VT;
            walk.GC4.RMS.sternum.acc.VT; walk.GC5.RMS.sternum.acc.VT;
            walk.GC6.RMS.sternum.acc.VT; walk.GC7.RMS.sternum.acc.VT];
        walk.RMS.sternum.acc.all.total = [walk.RMS.sternum.acc.all.total;
walk.GC3.RMS.sternum.acc.total;
            walk.GC4.RMS.sternum.acc.total; walk.GC5.RMS.sternum.acc.total;
            walk.GC6.RMS.sternum.acc.total; walk.GC7.RMS.sternum.acc.total];
        walk.RMS.lumbar.acc.all.AP = [walk.RMS.lumbar.acc.all.AP; walk.GC3.RMS.lumbar.acc.AP;
            walk.GC4.RMS.lumbar.acc.AP; walk.GC5.RMS.lumbar.acc.AP;
            walk.GC6.RMS.lumbar.acc.AP; walk.GC7.RMS.lumbar.acc.AP];
        walk.RMS.lumbar.acc.all.ML = [walk.RMS.lumbar.acc.all.ML; walk.GC3.RMS.lumbar.acc.ML;
            walk.GC4.RMS.lumbar.acc.ML; walk.GC5.RMS.lumbar.acc.ML;
            walk.GC6.RMS.lumbar.acc.ML; walk.GC7.RMS.lumbar.acc.ML];
        walk.RMS.lumbar.acc.all.VT = [walk.RMS.lumbar.acc.all.VT; walk.GC3.RMS.lumbar.acc.VT;
            walk.GC4.RMS.lumbar.acc.VT; walk.GC5.RMS.lumbar.acc.VT;
            walk.GC6.RMS.lumbar.acc.VT; walk.GC7.RMS.lumbar.acc.VT];
        walk.RMS.lumbar.acc.all.total = [walk.RMS.lumbar.acc.all.total;
walk.GC3.RMS.lumbar.acc.total;
            walk.GC4.RMS.lumbar.acc.total; walk.GC5.RMS.lumbar.acc.total;
            walk.GC6.RMS.lumbar.acc.total; walk.GC7.RMS.lumbar.acc.total];
    case 8
        walk.GC3 = getRMSAllIMUs(walk.GC3); walk.GC4 = getRMSAllIMUs(walk.GC4);
        walk.GC5 = getRMSAllIMUs(walk.GC5); walk.GC6 = getRMSAllIMUs(walk.GC6);
        walk.GC7 = getRMSAllIMUs(walk.GC7); walk.GC8 = getRMSAllIMUs(walk.GC8);
        walk.RMS.sternum.acc.all.AP = [walk.RMS.sternum.acc.all.AP;
walk.GC3.RMS.sternum.acc.AP;
            walk.GC4.RMS.sternum.acc.AP; walk.GC5.RMS.sternum.acc.AP;
            walk.GC6.RMS.sternum.acc.AP; walk.GC7.RMS.sternum.acc.AP;
            walk.GC8.RMS.sternum.acc.AP];
        walk.RMS.sternum.acc.all.ML = [walk.RMS.sternum.acc.all.ML;
walk.GC3.RMS.sternum.acc.ML;
            walk.GC4.RMS.sternum.acc.ML; walk.GC5.RMS.sternum.acc.ML;
            walk.GC6.RMS.sternum.acc.ML; walk.GC7.RMS.sternum.acc.ML;
            walk.GC8.RMS.sternum.acc.ML];
        walk.RMS.sternum.acc.all.VT = [walk.RMS.sternum.acc.all.VT;
walk.GC3.RMS.sternum.acc.VT;
            walk.GC4.RMS.sternum.acc.VT; walk.GC5.RMS.sternum.acc.VT;
            walk.GC6.RMS.sternum.acc.VT; walk.GC7.RMS.sternum.acc.VT;
            walk.GC8.RMS.sternum.acc.VT];
        walk.RMS.sternum.acc.all.total = [walk.RMS.sternum.acc.all.total;
walk.GC3.RMS.sternum.acc.total;
            walk.GC4.RMS.sternum.acc.total; walk.GC5.RMS.sternum.acc.total;
            walk.GC6.RMS.sternum.acc.total; walk.GC7.RMS.sternum.acc.total;
            walk.GC8.RMS.sternum.acc.total];
        walk.RMS.lumbar.acc.all.AP = [walk.RMS.lumbar.acc.all.AP; walk.GC3.RMS.lumbar.acc.AP;
            walk.GC4.RMS.lumbar.acc.AP; walk.GC5.RMS.lumbar.acc.AP;
            walk.GC6.RMS.lumbar.acc.AP; walk.GC7.RMS.lumbar.acc.AP;
            walk.GC8.RMS.lumbar.acc.AP];
        walk.RMS.lumbar.acc.all.ML = [walk.RMS.lumbar.acc.all.ML; walk.GC3.RMS.lumbar.acc.ML;
            walk.GC4.RMS.lumbar.acc.ML; walk.GC5.RMS.lumbar.acc.ML;
            walk.GC6.RMS.lumbar.acc.ML; walk.GC7.RMS.lumbar.acc.ML;
            walk.GC8.RMS.lumbar.acc.ML];
        walk.RMS.lumbar.acc.all.VT = [walk.RMS.lumbar.acc.all.VT; walk.GC3.RMS.lumbar.acc.VT;
            walk.GC4.RMS.lumbar.acc.VT; walk.GC5.RMS.lumbar.acc.VT;
            walk.GC6.RMS.lumbar.acc.VT; walk.GC7.RMS.lumbar.acc.VT;
            walk.GC8.RMS.lumbar.acc.VT];
```

```
        walk.RMS.lumbar.acc.all.total = [walk.RMS.lumbar.acc.all.total;
walk.GC3.RMS.lumbar.acc.total;
            walk.GC4.RMS.lumbar.acc.total; walk.GC5.RMS.lumbar.acc.total;
            walk.GC6.RMS.lumbar.acc.total; walk.GC7.RMS.lumbar.acc.total;
            walk.GC8.RMS.lumbar.acc.total];
    case 9
        walk.GC3 = getRMSAllIMUs(walk.GC3); walk.GC4 = getRMSAllIMUs(walk.GC4);
        walk.GC5 = getRMSAllIMUs(walk.GC5); walk.GC6 = getRMSAllIMUs(walk.GC6);
        walk.GC7 = getRMSAllIMUs(walk.GC7); walk.GC8 = getRMSAllIMUs(walk.GC8);
        walk.GC9 = getRMSAllIMUs(walk.GC9);
        walk.RMS.sternum.acc.all.AP = [walk.RMS.sternum.acc.all.AP;
walk.GC3.RMS.sternum.acc.AP;
            walk.GC4.RMS.sternum.acc.AP; walk.GC5.RMS.sternum.acc.AP;
            walk.GC6.RMS.sternum.acc.AP; walk.GC7.RMS.sternum.acc.AP;
            walk.GC8.RMS.sternum.acc.AP; walk.GC9.RMS.sternum.acc.AP];
        walk.RMS.sternum.acc.all.ML = [walk.RMS.sternum.acc.all.ML;
walk.GC3.RMS.sternum.acc.ML;
            walk.GC4.RMS.sternum.acc.ML; walk.GC5.RMS.sternum.acc.ML;
            walk.GC6.RMS.sternum.acc.ML; walk.GC7.RMS.sternum.acc.ML;
            walk.GC8.RMS.sternum.acc.ML; walk.GC9.RMS.sternum.acc.ML];
        walk.RMS.sternum.acc.all.VT = [walk.RMS.sternum.acc.all.VT;
walk.GC3.RMS.sternum.acc.VT;
            walk.GC4.RMS.sternum.acc.VT; walk.GC5.RMS.sternum.acc.VT;
            walk.GC6.RMS.sternum.acc.VT; walk.GC7.RMS.sternum.acc.VT;
            walk.GC8.RMS.sternum.acc.VT; walk.GC9.RMS.sternum.acc.VT];
        walk.RMS.sternum.acc.all.total = [walk.RMS.sternum.acc.all.total;
walk.GC3.RMS.sternum.acc.total;
            walk.GC4.RMS.sternum.acc.total; walk.GC5.RMS.sternum.acc.total;
            walk.GC6.RMS.sternum.acc.total; walk.GC7.RMS.sternum.acc.total;
            walk.GC8.RMS.sternum.acc.total; walk.GC9.RMS.sternum.acc.total];

        walk.RMS.lumbar.acc.all.AP = [walk.RMS.lumbar.acc.all.AP; walk.GC3.RMS.lumbar.acc.AP;
            walk.GC4.RMS.lumbar.acc.AP; walk.GC5.RMS.lumbar.acc.AP;
            walk.GC6.RMS.lumbar.acc.AP; walk.GC7.RMS.lumbar.acc.AP;
            walk.GC8.RMS.lumbar.acc.AP; walk.GC9.RMS.lumbar.acc.AP];
        walk.RMS.lumbar.acc.all.ML = [walk.RMS.lumbar.acc.all.ML; walk.GC3.RMS.lumbar.acc.ML;
            walk.GC4.RMS.lumbar.acc.ML; walk.GC5.RMS.lumbar.acc.ML;
            walk.GC6.RMS.lumbar.acc.ML; walk.GC7.RMS.lumbar.acc.ML;
            walk.GC8.RMS.lumbar.acc.ML; walk.GC9.RMS.lumbar.acc.ML];
        walk.RMS.lumbar.acc.all.VT = [walk.RMS.lumbar.acc.all.VT; walk.GC3.RMS.lumbar.acc.VT;
            walk.GC4.RMS.lumbar.acc.VT; walk.GC5.RMS.lumbar.acc.VT;
            walk.GC6.RMS.lumbar.acc.VT; walk.GC7.RMS.lumbar.acc.VT;
            walk.GC8.RMS.lumbar.acc.VT; walk.GC9.RMS.lumbar.acc.VT];
        walk.RMS.lumbar.acc.all.total = [walk.RMS.lumbar.acc.all.total;
walk.GC3.RMS.lumbar.acc.total;
            walk.GC4.RMS.lumbar.acc.total; walk.GC5.RMS.lumbar.acc.total;
            walk.GC6.RMS.lumbar.acc.total; walk.GC7.RMS.lumbar.acc.total;
            walk.GC8.RMS.lumbar.acc.total; walk.GC9.RMS.lumbar.acc.total];
    case 10
        walk.GC3 = getRMSAllIMUs(walk.GC3); walk.GC4 = getRMSAllIMUs(walk.GC4);
        walk.GC5 = getRMSAllIMUs(walk.GC5); walk.GC6 = getRMSAllIMUs(walk.GC6);
        walk.GC7 = getRMSAllIMUs(walk.GC7); walk.GC8 = getRMSAllIMUs(walk.GC8);
        walk.GC9 = getRMSAllIMUs(walk.GC9); walk.GC10 = getRMSAllIMUs(walk.GC10);

        walk.RMS.sternum.acc.all.AP = [walk.RMS.sternum.acc.all.AP;
walk.GC3.RMS.sternum.acc.AP;
            walk.GC4.RMS.sternum.acc.AP; walk.GC5.RMS.sternum.acc.AP;
            walk.GC6.RMS.sternum.acc.AP; walk.GC7.RMS.sternum.acc.AP;
            walk.GC8.RMS.sternum.acc.AP; walk.GC9.RMS.sternum.acc.AP;
            walk.GC10.RMS.sternum.acc.AP];
        walk.RMS.sternum.acc.all.ML = [walk.RMS.sternum.acc.all.ML;
walk.GC3.RMS.sternum.acc.ML;
            walk.GC4.RMS.sternum.acc.ML; walk.GC5.RMS.sternum.acc.ML;
            walk.GC6.RMS.sternum.acc.ML; walk.GC7.RMS.sternum.acc.ML;
            walk.GC8.RMS.sternum.acc.ML; walk.GC9.RMS.sternum.acc.ML;
            walk.GC10.RMS.sternum.acc.ML];
        walk.RMS.sternum.acc.all.VT = [walk.RMS.sternum.acc.all.VT;
walk.GC3.RMS.sternum.acc.VT;
```

```
            walk.GC4.RMS.sternum.acc.VT; walk.GC5.RMS.sternum.acc.VT;
            walk.GC6.RMS.sternum.acc.VT; walk.GC7.RMS.sternum.acc.VT;
            walk.GC8.RMS.sternum.acc.VT; walk.GC9.RMS.sternum.acc.VT;
            walk.GC10.RMS.sternum.acc.VT];
        walk.RMS.sternum.acc.all.total = [walk.RMS.sternum.acc.all.total;
walk.GC3.RMS.sternum.acc.total;
            walk.GC4.RMS.sternum.acc.total; walk.GC5.RMS.sternum.acc.total;
            walk.GC6.RMS.sternum.acc.total; walk.GC7.RMS.sternum.acc.total;
            walk.GC8.RMS.sternum.acc.total; walk.GC9.RMS.sternum.acc.total;
            walk.GC10.RMS.sternum.acc.total];

        walk.RMS.lumbar.acc.all.AP = [walk.RMS.lumbar.acc.all.AP; walk.GC3.RMS.lumbar.acc.AP;
            walk.GC4.RMS.lumbar.acc.AP; walk.GC5.RMS.lumbar.acc.AP;
            walk.GC6.RMS.lumbar.acc.AP; walk.GC7.RMS.lumbar.acc.AP;
            walk.GC8.RMS.lumbar.acc.AP; walk.GC9.RMS.lumbar.acc.AP;
            walk.GC10.RMS.lumbar.acc.AP];
        walk.RMS.lumbar.acc.all.ML = [walk.RMS.lumbar.acc.all.ML; walk.GC3.RMS.lumbar.acc.ML;
            walk.GC4.RMS.lumbar.acc.ML; walk.GC5.RMS.lumbar.acc.ML;
            walk.GC6.RMS.lumbar.acc.ML; walk.GC7.RMS.lumbar.acc.ML;
            walk.GC8.RMS.lumbar.acc.ML; walk.GC9.RMS.lumbar.acc.ML;
            walk.GC10.RMS.lumbar.acc.ML];
        walk.RMS.lumbar.acc.all.VT = [walk.RMS.lumbar.acc.all.VT; walk.GC3.RMS.lumbar.acc.VT;
            walk.GC4.RMS.lumbar.acc.VT; walk.GC5.RMS.lumbar.acc.VT;
            walk.GC6.RMS.lumbar.acc.VT; walk.GC7.RMS.lumbar.acc.VT;
            walk.GC8.RMS.lumbar.acc.VT; walk.GC9.RMS.lumbar.acc.VT;
            walk.GC10.RMS.lumbar.acc.VT];
        walk.RMS.lumbar.acc.all.total = [walk.RMS.lumbar.acc.all.total;
walk.GC3.RMS.lumbar.acc.total;
            walk.GC4.RMS.lumbar.acc.total; walk.GC5.RMS.lumbar.acc.total;
            walk.GC6.RMS.lumbar.acc.total; walk.GC7.RMS.lumbar.acc.total;
            walk.GC8.RMS.lumbar.acc.total; walk.GC9.RMS.lumbar.acc.total;
            walk.GC10.RMS.lumbar.acc.total];
end
end
function [RMS, RMSR] = getRMS(data)
RMS.AP = rms(data(:,1)); RMS.ML = rms(data(:,2)); RMS.VT = rms(data(:,3));
RMS.total = sqrt(RMS.AP^2 + RMS.ML^2 + RMS.VT^2);
% RMS Ratio (Sekine et al.,2013)
RMSR.AP = RMS.AP/RMS.total; RMSR.ML = RMS.ML/RMS.total; RMSR.VT = RMS.VT/RMS.total;
end
%% 3. Attenuation coefficient (AC)
function AC = getAC(gc)
% Attenuation coefficient from sternum to lumbar sensor (per axis)
AC.AP = (1 - gc.RMS.sternum.acc.AP/gc.RMS.lumbar.acc.AP)*100;
AC.ML = (1 - gc.RMS.sternum.acc.ML/gc.RMS.lumbar.acc.ML)*100;
AC.VT = (1 - gc.RMS.sternum.acc.VT/gc.RMS.lumbar.acc.VT)*100;
% Attenuation coefficient from sternum to lumbar sensor for all axes
AC.total = (1 - gc.RMS.sternum.acc.total/gc.RMS.lumbar.acc.total)*100;
end
function walk = getACPerWalk(walk)
% Get AC per all Walk
walk.AC = getAC(walk);
% Get AC per GC
walk.GC1.AC = getAC(walk.GC1);
walk.GC2.AC = getAC(walk.GC2);
% Gathering GC's AC per axis
walk.AC.all.AP = [walk.GC1.AC.AP; walk.GC2.AC.AP];
walk.AC.all.ML = [walk.GC1.AC.ML; walk.GC2.AC.ML];
walk.AC.all.VT = [walk.GC1.AC.VT; walk.GC2.AC.VT];
walk.AC.all.total = [walk.GC1.AC.total; walk.GC2.AC.total];
switch walk.nrGC
    case 3
        walk.GC3.AC = getAC(walk.GC3);
        % Gathering GC's AC per axis
        walk.AC.all.AP = [walk.AC.all.AP; walk.GC3.AC.AP];
        walk.AC.all.ML = [walk.AC.all.ML; walk.GC3.AC.ML];
        walk.AC.all.VT = [walk.AC.all.VT; walk.GC3.AC.VT];
        walk.AC.all.total = [walk.AC.all.total; walk.GC3.AC.total];
```

```
case 4
    walk.GC3.AC = getAC(walk.GC3); walk.GC4.AC = getAC(walk.GC4);
    % Gathering GC's AC per axis
    walk.AC.all.AP = [walk.AC.all.AP; walk.GC3.AC.AP; walk.GC4.AC.AP];
    walk.AC.all.ML = [walk.AC.all.ML; walk.GC3.AC.ML; walk.GC4.AC.ML];
    walk.AC.all.VT = [walk.AC.all.VT; walk.GC3.AC.VT; walk.GC4.AC.VT];
    walk.AC.all.total = [walk.AC.all.total; walk.GC3.AC.total; walk.GC4.AC.total];
case 5
    walk.GC3.AC = getAC(walk.GC3); walk.GC4.AC = getAC(walk.GC4);
    walk.GC5.AC = getAC(walk.GC5);
    % Gathering GC's AC per axis
    walk.AC.all.AP = [walk.AC.all.AP; walk.GC3.AC.AP; walk.GC4.AC.AP; walk.GC5.AC.AP];
    walk.AC.all.ML = [walk.AC.all.ML; walk.GC3.AC.ML; walk.GC4.AC.ML; walk.GC5.AC.ML];
    walk.AC.all.VT = [walk.AC.all.VT; walk.GC3.AC.VT; walk.GC4.AC.VT; walk.GC5.AC.VT];
    walk.AC.all.total = [walk.AC.all.total; walk.GC3.AC.total; walk.GC4.AC.total;
        walk.GC5.AC.total];
case 6
    walk.GC3.AC = getAC(walk.GC3); walk.GC4.AC = getAC(walk.GC4);
    walk.GC5.AC = getAC(walk.GC5); walk.GC6.AC = getAC(walk.GC6);
    % Gathering GC's AC per axis
    walk.AC.all.AP = [walk.AC.all.AP; walk.GC3.AC.AP; walk.GC4.AC.AP;
        walk.GC5.AC.AP; walk.GC6.AC.AP];
    walk.AC.all.ML = [walk.AC.all.ML; walk.GC3.AC.ML; walk.GC4.AC.ML;
        walk.GC5.AC.ML; walk.GC6.AC.ML];
    walk.AC.all.VT = [walk.AC.all.VT; walk.GC3.AC.VT; walk.GC4.AC.VT;
        walk.GC5.AC.VT; walk.GC6.AC.VT];
    walk.AC.all.total = [walk.AC.all.total; walk.GC3.AC.total; walk.GC4.AC.total;
        walk.GC5.AC.total; walk.GC6.AC.total];
case 7
    walk.GC3.AC = getAC(walk.GC3); walk.GC4.AC = getAC(walk.GC4);
    walk.GC5.AC = getAC(walk.GC5); walk.GC6.AC = getAC(walk.GC6);
    walk.GC7.AC = getAC(walk.GC7);
    % Gathering GC's AC per axis
    walk.AC.all.AP = [walk.AC.all.AP; walk.GC3.AC.AP; walk.GC4.AC.AP;
        walk.GC5.AC.AP; walk.GC6.AC.AP; walk.GC7.AC.AP];
    walk.AC.all.ML = [walk.AC.all.ML; walk.GC3.AC.ML; walk.GC4.AC.ML;
        walk.GC5.AC.ML; walk.GC6.AC.ML; walk.GC7.AC.ML];
    walk.AC.all.VT = [walk.AC.all.VT; walk.GC3.AC.VT; walk.GC4.AC.VT;
        walk.GC5.AC.VT; walk.GC6.AC.VT; walk.GC7.AC.VT];
    walk.AC.all.total = [walk.AC.all.total; walk.GC3.AC.total; walk.GC4.AC.total;
        walk.GC5.AC.total; walk.GC6.AC.total; walk.GC7.AC.total];
case 8
    walk.GC3.AC = getAC(walk.GC3); walk.GC4.AC = getAC(walk.GC4);
    walk.GC5.AC = getAC(walk.GC5); walk.GC6.AC = getAC(walk.GC6);
    walk.GC7.AC = getAC(walk.GC7); walk.GC8.AC = getAC(walk.GC8);
    % Gathering GC's AC per axis
    walk.AC.all.AP = [walk.AC.all.AP; walk.GC3.AC.AP; walk.GC4.AC.AP;
        walk.GC5.AC.AP; walk.GC6.AC.AP; walk.GC7.AC.AP;
        walk.GC8.AC.AP];
    walk.AC.all.ML = [walk.AC.all.ML; walk.GC3.AC.ML; walk.GC4.AC.ML;
        walk.GC5.AC.ML; walk.GC6.AC.ML; walk.GC7.AC.ML;
        walk.GC8.AC.ML];
    walk.AC.all.VT = [walk.AC.all.VT; walk.GC3.AC.VT; walk.GC4.AC.VT;
        walk.GC5.AC.VT; walk.GC6.AC.VT; walk.GC7.AC.VT;
        walk.GC8.AC.VT];
    walk.AC.all.total = [walk.AC.all.total; walk.GC3.AC.total; walk.GC4.AC.total;
        walk.GC5.AC.total; walk.GC6.AC.total; walk.GC7.AC.total;
        walk.GC8.AC.total];
case 9
    walk.GC3.AC = getAC(walk.GC3); walk.GC4.AC = getAC(walk.GC4);
    walk.GC5.AC = getAC(walk.GC5); walk.GC6.AC = getAC(walk.GC6);
    walk.GC7.AC = getAC(walk.GC7); walk.GC8.AC = getAC(walk.GC8);
    walk.GC9.AC = getAC(walk.GC9);
    % Gathering GC's AC per axis
    walk.AC.all.AP = [walk.AC.all.AP; walk.GC3.AC.AP; walk.GC4.AC.AP;
        walk.GC5.AC.AP; walk.GC6.AC.AP; walk.GC7.AC.AP;
        walk.GC8.AC.AP; walk.GC9.AC.AP];
    walk.AC.all.ML = [walk.AC.all.ML; walk.GC3.AC.ML; walk.GC4.AC.ML;
```

```
            walk.GC5.AC.ML; walk.GC6.AC.ML; walk.GC7.AC.ML;
            walk.GC8.AC.ML; walk.GC9.AC.ML];
        walk.AC.all.VT = [walk.AC.all.VT; walk.GC3.AC.VT; walk.GC4.AC.VT;
            walk.GC5.AC.VT; walk.GC6.AC.VT; walk.GC7.AC.VT;
            walk.GC8.AC.VT; walk.GC9.AC.VT];
        walk.AC.all.total = [walk.AC.all.total; walk.GC3.AC.total; walk.GC4.AC.total;
            walk.GC5.AC.total; walk.GC6.AC.total; walk.GC7.AC.total;
            walk.GC8.AC.total; walk.GC9.AC.total];
    case 10
        walk.GC3.AC = getAC(walk.GC3); walk.GC4.AC = getAC(walk.GC4);
        walk.GC5.AC = getAC(walk.GC5); walk.GC6.AC = getAC(walk.GC6);
        walk.GC7.AC = getAC(walk.GC7); walk.GC8.AC = getAC(walk.GC8);
        walk.GC9.AC = getAC(walk.GC9); walk.GC10.AC = getAC(walk.GC10);
        % Gathering GC's AC per axis
        walk.AC.all.AP = [walk.AC.all.AP; walk.GC3.AC.AP; walk.GC4.AC.AP;
            walk.GC5.AC.AP; walk.GC6.AC.AP; walk.GC7.AC.AP;
            walk.GC8.AC.AP; walk.GC9.AC.AP; walk.GC10.AC.AP];
        walk.AC.all.ML = [walk.AC.all.ML; walk.GC3.AC.ML; walk.GC4.AC.ML;
            walk.GC5.AC.ML; walk.GC6.AC.ML; walk.GC7.AC.ML;
            walk.GC8.AC.ML; walk.GC9.AC.ML; walk.GC10.AC.ML];
        walk.AC.all.VT = [walk.AC.all.VT; walk.GC3.AC.VT; walk.GC4.AC.VT;
            walk.GC5.AC.VT; walk.GC6.AC.VT; walk.GC7.AC.VT;
            walk.GC8.AC.VT; walk.GC9.AC.VT; walk.GC10.AC.VT];
        walk.AC.all.total = [walk.AC.all.total; walk.GC3.AC.total; walk.GC4.AC.total;
            walk.GC5.AC.total; walk.GC6.AC.total; walk.GC7.AC.total;
            walk.GC8.AC.total; walk.GC9.AC.total; walk.GC10.AC.total];
end
end
function struct = getACPerDataset(struct)
% Get AC values per walk
struct.W1 = getACPerWalk(struct.W1);
struct.W2 = getACPerWalk(struct.W2);
struct.W3 = getACPerWalk(struct.W3);
struct.W4 = getACPerWalk(struct.W4);
struct.W5 = getACPerWalk(struct.W5);
struct.W6 = getACPerWalk(struct.W6);
% Get mean AC values per axis
struct.AC.AP = [struct.W1.AC.AP, struct.W2.AC.AP, struct.W3.AC.AP, struct.W4.AC.AP,
struct.W5.AC.AP, struct.W6.AC.AP];
struct.AC.ML = [struct.W1.AC.ML, struct.W2.AC.ML, struct.W3.AC.ML, struct.W4.AC.ML,
struct.W5.AC.ML, struct.W6.AC.ML];
struct.AC.VT = [struct.W1.AC.VT, struct.W2.AC.VT, struct.W3.AC.VT, struct.W4.AC.VT,
struct.W5.AC.VT, struct.W6.AC.VT];
struct.AC.total = [struct.W1.AC.total, struct.W2.AC.total, struct.W3.AC.total,
struct.W4.AC.total, struct.W5.AC.total, struct.W6.AC.total];
struct.avg.AC.AP = mean(struct.AC.AP);
struct.avg.AC.ML = mean(struct.AC.ML);
struct.avg.AC.VT = mean(struct.AC.VT);
struct.avg.AC.total = mean(struct.AC.total);
struct.AC.all.ML = [struct.W1.AC.all.ML', struct.W2.AC.all.ML', struct.W3.AC.all.ML',
struct.W4.AC.all.ML', struct.W5.AC.all.ML', struct.W6.AC.all.ML'];
end
%% 4. Harmonic Ratio (HR)
function HR = getHR(data)
% Number of harmonics
harm = 20; i = 1;
% FFT
X = fft(data); XM = abs(X);
% Extracting amplitude of even and odd harmonics
oddHarm.AP = abs(XM(i:2:harm,1));      % A_i*2-1, i=1,3,5,...
evenHarm.AP = abs(XM(i+1:2:harm+1,1)); % A_i*2  , i=2,4,6,...
oddHarm.ML = abs(XM(i:2:harm,2));      % A_i*2-1, i=1,3,5,...
evenHarm.ML = abs(XM(i+1:2:harm+1,2)); % A_i*2  , i=2,4,6,...
oddHarm.VT = abs(XM(i:2:harm,3));      % A_i*2-1, i=1,3,5,...
evenHarm.VT = abs(XM(i+1:2:harm+1,3)); % A_i*2  , i=2,4,6,...
% Calculate harmonic ratio
HR.AP = sum(evenHarm.AP)/ sum(oddHarm.AP);
HR.ML = sum(oddHarm.ML) / sum(evenHarm.ML);
```

```
HR.VT = sum(evenHarm.VT)/ sum(oddHarm.VT);
end
% Compare HR from a Gait segment
function walk = getHRPerWalk(walk)
% --- Input ---
% Walk: Struct data from a Walk segment
switch walk.nrGC
    case 2
        % Sternum HR
        walk.GC1.HR.sternum = getHR(walk.GC1.sternum.acc);
        walk.GC2.HR.sternum = getHR(walk.GC2.sternum.acc);
        % Lumbar HR
        walk.GC1.HR.lumbar  = getHR(walk.GC1.lumbar.acc);
        walk.GC2.HR.lumbar  = getHR(walk.GC2.lumbar.acc);
        % Every axis Sternum's HR
        walk.HR.sternum.AP = [walk.GC1.HR.sternum.AP, walk.GC2.HR.sternum.AP];
        walk.HR.sternum.ML = [walk.GC1.HR.sternum.ML, walk.GC2.HR.sternum.ML];
        walk.HR.sternum.VT = [walk.GC1.HR.sternum.VT, walk.GC2.HR.sternum.VT];
        % Every axis Lumbar's HR
        walk.HR.lumbar.AP = [walk.GC1.HR.lumbar.AP, walk.GC2.HR.lumbar.AP];
        walk.HR.lumbar.ML = [walk.GC1.HR.lumbar.ML, walk.GC2.HR.lumbar.ML];
        walk.HR.lumbar.VT = [walk.GC1.HR.lumbar.VT, walk.GC2.HR.lumbar.VT];
    case 3
        % Sternum HR
        walk.GC1.HR.sternum = getHR(walk.GC1.sternum.acc);
        walk.GC2.HR.sternum = getHR(walk.GC2.sternum.acc);
        walk.GC3.HR.sternum = getHR(walk.GC3.sternum.acc);
        % Lumbar HR
        walk.GC1.HR.lumbar  = getHR(walk.GC1.lumbar.acc);
        walk.GC2.HR.lumbar  = getHR(walk.GC2.lumbar.acc);
        walk.GC3.HR.lumbar  = getHR(walk.GC3.lumbar.acc);
        % Every axis Sternum's HR
        walk.HR.sternum.AP = [walk.GC1.HR.sternum.AP, walk.GC2.HR.sternum.AP,
walk.GC3.HR.sternum.AP];
        walk.HR.sternum.ML = [walk.GC1.HR.sternum.ML, walk.GC2.HR.sternum.ML,
walk.GC3.HR.sternum.ML];
        walk.HR.sternum.VT = [walk.GC1.HR.sternum.VT, walk.GC2.HR.sternum.VT,
walk.GC3.HR.sternum.VT];
        % Every axis Lumbar's HR
        walk.HR.lumbar.AP = [walk.GC1.HR.lumbar.AP, walk.GC2.HR.lumbar.AP,
walk.GC3.HR.lumbar.AP];
        walk.HR.lumbar.ML = [walk.GC1.HR.lumbar.ML, walk.GC2.HR.lumbar.ML,
walk.GC3.HR.lumbar.ML];
        walk.HR.lumbar.VT = [walk.GC1.HR.lumbar.VT, walk.GC2.HR.lumbar.VT,
walk.GC3.HR.lumbar.VT];
    case 4
        % Sternum HR
        walk.GC1.HR.sternum  = getHR(walk.GC1.sternum.acc);
        walk.GC2.HR.sternum  = getHR(walk.GC2.sternum.acc);
        walk.GC3.HR.sternum  = getHR(walk.GC3.sternum.acc);
        walk.GC4.HR.sternum  = getHR(walk.GC4.sternum.acc);
        % Lumbar HR
        walk.GC1.HR.lumbar  = getHR(walk.GC1.lumbar.acc);
        walk.GC2.HR.lumbar  = getHR(walk.GC2.lumbar.acc);
        walk.GC3.HR.lumbar  = getHR(walk.GC3.lumbar.acc);
        walk.GC4.HR.lumbar  = getHR(walk.GC4.lumbar.acc);
        % Every axis Sternum's HR
        walk.HR.sternum.AP = [walk.GC1.HR.sternum.AP, walk.GC2.HR.sternum.AP,...
            walk.GC3.HR.sternum.AP, walk.GC4.HR.sternum.AP];
        walk.HR.sternum.ML = [walk.GC1.HR.sternum.ML, walk.GC2.HR.sternum.ML,...
            walk.GC3.HR.sternum.ML, walk.GC4.HR.sternum.ML];
        walk.HR.sternum.VT = [walk.GC1.HR.sternum.VT, walk.GC2.HR.sternum.VT,...
            walk.GC3.HR.sternum.VT, walk.GC4.HR.sternum.VT];
        % Every axis Lumbar's HR
        walk.HR.lumbar.AP = [walk.GC1.HR.lumbar.AP, walk.GC2.HR.lumbar.AP,...
            walk.GC3.HR.lumbar.AP, walk.GC4.HR.lumbar.AP];
        walk.HR.lumbar.ML = [walk.GC1.HR.lumbar.ML, walk.GC2.HR.lumbar.ML,...
            walk.GC3.HR.lumbar.ML, walk.GC4.HR.lumbar.ML];
```

```
        walk.HR.lumbar.VT = [walk.GC1.HR.lumbar.VT, walk.GC2.HR.lumbar.VT,...
            walk.GC3.HR.lumbar.VT, walk.GC4.HR.lumbar.VT];
    case 5
        % Sternum HR
        walk.GC1.HR.sternum  = getHR(walk.GC1.sternum.acc);
        walk.GC2.HR.sternum  = getHR(walk.GC2.sternum.acc);
        walk.GC3.HR.sternum  = getHR(walk.GC3.sternum.acc);
        walk.GC4.HR.sternum  = getHR(walk.GC4.sternum.acc);
        walk.GC5.HR.sternum  = getHR(walk.GC5.sternum.acc);
        % Lumbar HR
        walk.GC1.HR.lumbar  = getHR(walk.GC1.lumbar.acc);
        walk.GC2.HR.lumbar  = getHR(walk.GC2.lumbar.acc);
        walk.GC3.HR.lumbar  = getHR(walk.GC3.lumbar.acc);
        walk.GC4.HR.lumbar  = getHR(walk.GC4.lumbar.acc);
        walk.GC5.HR.lumbar  = getHR(walk.GC5.lumbar.acc);
        % Every axis Sternum's HR
        walk.HR.sternum.AP = [walk.GC1.HR.sternum.AP, walk.GC2.HR.sternum.AP,...
            walk.GC3.HR.sternum.AP, walk.GC4.HR.sternum.AP, walk.GC5.HR.sternum.AP];
        walk.HR.sternum.ML = [walk.GC1.HR.sternum.ML, walk.GC2.HR.sternum.ML,...
            walk.GC3.HR.sternum.ML, walk.GC4.HR.sternum.ML, walk.GC5.HR.sternum.ML];
        walk.HR.sternum.VT = [walk.GC1.HR.sternum.VT, walk.GC2.HR.sternum.VT,...
            walk.GC3.HR.sternum.VT, walk.GC4.HR.sternum.VT, walk.GC5.HR.sternum.VT];
        % Every axis Lumbar's HR
        walk.HR.lumbar.AP = [walk.GC1.HR.lumbar.AP, walk.GC2.HR.lumbar.AP,...
            walk.GC3.HR.lumbar.AP, walk.GC4.HR.lumbar.AP, walk.GC5.HR.lumbar.AP];
        walk.HR.lumbar.ML = [walk.GC1.HR.lumbar.ML, walk.GC2.HR.lumbar.ML,...
            walk.GC3.HR.lumbar.ML, walk.GC4.HR.lumbar.ML, walk.GC5.HR.lumbar.ML];
        walk.HR.lumbar.VT = [walk.GC1.HR.lumbar.VT, walk.GC2.HR.lumbar.VT,...
            walk.GC3.HR.lumbar.VT, walk.GC4.HR.lumbar.VT, walk.GC5.HR.lumbar.VT];
    case 6
        % Sternum HR
        walk.GC1.HR.sternum  = getHR(walk.GC1.sternum.acc);
        walk.GC2.HR.sternum  = getHR(walk.GC2.sternum.acc);
        walk.GC3.HR.sternum  = getHR(walk.GC3.sternum.acc);
        walk.GC4.HR.sternum  = getHR(walk.GC4.sternum.acc);
        walk.GC5.HR.sternum  = getHR(walk.GC5.sternum.acc);
        walk.GC6.HR.sternum  = getHR(walk.GC6.sternum.acc);
        % Lumbar HR
        walk.GC1.HR.lumbar  = getHR(walk.GC1.lumbar.acc);
        walk.GC2.HR.lumbar  = getHR(walk.GC2.lumbar.acc);
        walk.GC3.HR.lumbar  = getHR(walk.GC3.lumbar.acc);
        walk.GC4.HR.lumbar  = getHR(walk.GC4.lumbar.acc);
        walk.GC5.HR.lumbar  = getHR(walk.GC5.lumbar.acc);
        walk.GC6.HR.lumbar  = getHR(walk.GC6.lumbar.acc);
        % Every axis Sternum's HR
        walk.HR.sternum.AP = [walk.GC1.HR.sternum.AP, walk.GC2.HR.sternum.AP,
walk.GC3.HR.sternum.AP,...
            walk.GC4.HR.sternum.AP, walk.GC5.HR.sternum.AP, walk.GC6.HR.sternum.AP];
        walk.HR.sternum.ML = [walk.GC1.HR.sternum.ML, walk.GC2.HR.sternum.ML,
walk.GC3.HR.sternum.ML,...
            walk.GC4.HR.sternum.ML, walk.GC5.HR.sternum.ML, walk.GC6.HR.sternum.ML];
        walk.HR.sternum.VT = [walk.GC1.HR.sternum.VT, walk.GC2.HR.sternum.VT,
walk.GC3.HR.sternum.VT,...
            walk.GC4.HR.sternum.VT, walk.GC5.HR.sternum.VT, walk.GC6.HR.sternum.VT];
        % Every axis Lumbar's HR
        walk.HR.lumbar.AP = [walk.GC1.HR.lumbar.AP, walk.GC2.HR.lumbar.AP,
walk.GC3.HR.lumbar.AP,...
            walk.GC4.HR.lumbar.AP, walk.GC5.HR.lumbar.AP, walk.GC6.HR.lumbar.AP];
        walk.HR.lumbar.ML = [walk.GC1.HR.lumbar.ML, walk.GC2.HR.lumbar.ML,
walk.GC3.HR.lumbar.ML,...
            walk.GC4.HR.lumbar.ML, walk.GC5.HR.lumbar.ML, walk.GC6.HR.lumbar.ML];
        walk.HR.lumbar.VT = [walk.GC1.HR.lumbar.VT, walk.GC2.HR.lumbar.VT,
walk.GC3.HR.lumbar.VT,...
            walk.GC4.HR.lumbar.VT, walk.GC5.HR.lumbar.VT, walk.GC6.HR.lumbar.VT];
    case 7
        % Sternum HR
        walk.GC1.HR.sternum  = getHR(walk.GC1.sternum.acc);
        walk.GC2.HR.sternum  = getHR(walk.GC2.sternum.acc);
```

```
        walk.GC3.HR.sternum  = getHR(walk.GC3.sternum.acc);
        walk.GC4.HR.sternum  = getHR(walk.GC4.sternum.acc);
        walk.GC5.HR.sternum  = getHR(walk.GC5.sternum.acc);
        walk.GC6.HR.sternum  = getHR(walk.GC6.sternum.acc);
        walk.GC7.HR.sternum  = getHR(walk.GC7.sternum.acc);
        % Lumbar HR
        walk.GC1.HR.lumbar  = getHR(walk.GC1.lumbar.acc);
        walk.GC2.HR.lumbar  = getHR(walk.GC2.lumbar.acc);
        walk.GC3.HR.lumbar  = getHR(walk.GC3.lumbar.acc);
        walk.GC4.HR.lumbar  = getHR(walk.GC4.lumbar.acc);
        walk.GC5.HR.lumbar  = getHR(walk.GC5.lumbar.acc);
        walk.GC6.HR.lumbar  = getHR(walk.GC6.lumbar.acc);
        walk.GC7.HR.lumbar  = getHR(walk.GC7.lumbar.acc);
        % Every axis Sternum's HR
        walk.HR.sternum.AP = [walk.GC1.HR.sternum.AP, walk.GC2.HR.sternum.AP,
walk.GC3.HR.sternum.AP,...
            walk.GC4.HR.sternum.AP, walk.GC5.HR.sternum.AP, walk.GC6.HR.sternum.AP,
walk.GC7.HR.sternum.AP];
        walk.HR.sternum.ML = [walk.GC1.HR.sternum.ML, walk.GC2.HR.sternum.ML,
walk.GC3.HR.sternum.ML,...
            walk.GC4.HR.sternum.ML, walk.GC5.HR.sternum.ML, walk.GC6.HR.sternum.ML,
walk.GC7.HR.sternum.ML];
        walk.HR.sternum.VT = [walk.GC1.HR.sternum.VT, walk.GC2.HR.sternum.VT,
walk.GC3.HR.sternum.VT,...
            walk.GC4.HR.sternum.VT, walk.GC5.HR.sternum.VT, walk.GC6.HR.sternum.VT,
walk.GC7.HR.sternum.VT];
        % Every axis Lumbar's HR
        walk.HR.lumbar.AP = [walk.GC1.HR.lumbar.AP, walk.GC2.HR.lumbar.AP,
walk.GC3.HR.lumbar.AP,...
            walk.GC4.HR.lumbar.AP, walk.GC5.HR.lumbar.AP, walk.GC6.HR.lumbar.AP,
walk.GC7.HR.lumbar.AP];
        walk.HR.lumbar.ML = [walk.GC1.HR.lumbar.ML, walk.GC2.HR.lumbar.ML,
walk.GC3.HR.lumbar.ML,...
            walk.GC4.HR.lumbar.ML, walk.GC5.HR.lumbar.ML, walk.GC6.HR.lumbar.ML,
walk.GC7.HR.lumbar.ML];
        walk.HR.lumbar.VT = [walk.GC1.HR.lumbar.VT, walk.GC2.HR.lumbar.VT,
walk.GC3.HR.lumbar.VT,...
            walk.GC4.HR.lumbar.VT, walk.GC5.HR.lumbar.VT, walk.GC6.HR.lumbar.VT,
walk.GC7.HR.lumbar.VT];
    case 8
        % Sternum HR
        walk.GC1.HR.sternum  = getHR(walk.GC1.sternum.acc);
        walk.GC2.HR.sternum  = getHR(walk.GC2.sternum.acc);
        walk.GC3.HR.sternum  = getHR(walk.GC3.sternum.acc);
        walk.GC4.HR.sternum  = getHR(walk.GC4.sternum.acc);
        walk.GC5.HR.sternum  = getHR(walk.GC5.sternum.acc);
        walk.GC6.HR.sternum  = getHR(walk.GC6.sternum.acc);
        walk.GC7.HR.sternum  = getHR(walk.GC7.sternum.acc);
        walk.GC8.HR.sternum  = getHR(walk.GC8.sternum.acc);
        % Lumbar HR
        walk.GC1.HR.lumbar  = getHR(walk.GC1.lumbar.acc);
        walk.GC2.HR.lumbar  = getHR(walk.GC2.lumbar.acc);
        walk.GC3.HR.lumbar  = getHR(walk.GC3.lumbar.acc);
        walk.GC4.HR.lumbar  = getHR(walk.GC4.lumbar.acc);
        walk.GC5.HR.lumbar  = getHR(walk.GC5.lumbar.acc);
        walk.GC6.HR.lumbar  = getHR(walk.GC6.lumbar.acc);
        walk.GC7.HR.lumbar  = getHR(walk.GC7.lumbar.acc);
        walk.GC8.HR.lumbar  = getHR(walk.GC8.lumbar.acc);
        % Every axis Sternum's HR
        walk.HR.sternum.AP = [walk.GC1.HR.sternum.AP, walk.GC2.HR.sternum.AP,
walk.GC3.HR.sternum.AP,...
            walk.GC4.HR.sternum.AP, walk.GC5.HR.sternum.AP, walk.GC6.HR.sternum.AP,...
            walk.GC7.HR.sternum.AP, walk.GC8.HR.sternum.AP];
        walk.HR.sternum.ML = [walk.GC1.HR.sternum.ML, walk.GC2.HR.sternum.ML,
walk.GC3.HR.sternum.ML,...
            walk.GC4.HR.sternum.ML, walk.GC5.HR.sternum.ML, walk.GC6.HR.sternum.ML,...
            walk.GC7.HR.sternum.ML, walk.GC8.HR.sternum.ML];
```

```
        walk.HR.sternum.VT = [walk.GC1.HR.sternum.VT, walk.GC2.HR.sternum.VT,
walk.GC3.HR.sternum.VT,...
            walk.GC4.HR.sternum.VT, walk.GC5.HR.sternum.VT, walk.GC6.HR.sternum.VT,...
            walk.GC7.HR.sternum.VT, walk.GC8.HR.sternum.VT];
        % Every axis Lumbar's HR
        walk.HR.lumbar.AP = [walk.GC1.HR.lumbar.AP, walk.GC2.HR.lumbar.AP,
walk.GC3.HR.lumbar.AP,...
            walk.GC4.HR.lumbar.AP, walk.GC5.HR.lumbar.AP, walk.GC6.HR.lumbar.AP,...
            walk.GC7.HR.lumbar.AP, walk.GC8.HR.lumbar.AP];
        walk.HR.lumbar.ML = [walk.GC1.HR.lumbar.ML, walk.GC2.HR.lumbar.ML,
walk.GC3.HR.lumbar.ML,...
            walk.GC4.HR.lumbar.ML, walk.GC5.HR.lumbar.ML, walk.GC6.HR.lumbar.ML,...
            walk.GC7.HR.lumbar.ML, walk.GC8.HR.lumbar.ML];
        walk.HR.lumbar.VT = [walk.GC1.HR.lumbar.VT, walk.GC2.HR.lumbar.VT,
walk.GC3.HR.lumbar.VT,...
            walk.GC4.HR.lumbar.VT, walk.GC5.HR.lumbar.VT, walk.GC6.HR.lumbar.VT,...
            walk.GC7.HR.lumbar.VT, walk.GC8.HR.lumbar.VT];
    case 9
        % Sternum HR
        walk.GC1.HR.sternum  = getHR(walk.GC1.sternum.acc);
        walk.GC2.HR.sternum  = getHR(walk.GC2.sternum.acc);
        walk.GC3.HR.sternum  = getHR(walk.GC3.sternum.acc);
        walk.GC4.HR.sternum  = getHR(walk.GC4.sternum.acc);
        walk.GC5.HR.sternum  = getHR(walk.GC5.sternum.acc);
        walk.GC6.HR.sternum  = getHR(walk.GC6.sternum.acc);
        walk.GC7.HR.sternum  = getHR(walk.GC7.sternum.acc);
        walk.GC8.HR.sternum  = getHR(walk.GC8.sternum.acc);
        walk.GC9.HR.sternum  = getHR(walk.GC9.sternum.acc);
        % Lumbar HR
        walk.GC1.HR.lumbar  = getHR(walk.GC1.lumbar.acc);
        walk.GC2.HR.lumbar  = getHR(walk.GC2.lumbar.acc);
        walk.GC3.HR.lumbar  = getHR(walk.GC3.lumbar.acc);
        walk.GC4.HR.lumbar  = getHR(walk.GC4.lumbar.acc);
        walk.GC5.HR.lumbar  = getHR(walk.GC5.lumbar.acc);
        walk.GC6.HR.lumbar  = getHR(walk.GC6.lumbar.acc);
        walk.GC7.HR.lumbar  = getHR(walk.GC7.lumbar.acc);
        walk.GC8.HR.lumbar  = getHR(walk.GC8.lumbar.acc);
        walk.GC9.HR.lumbar  = getHR(walk.GC9.lumbar.acc);
        % Every axis Sternum's HR
        walk.HR.sternum.AP = [walk.GC1.HR.sternum.AP, walk.GC2.HR.sternum.AP,
walk.GC3.HR.sternum.AP,...
            walk.GC4.HR.sternum.AP, walk.GC5.HR.sternum.AP, walk.GC6.HR.sternum.AP,...
            walk.GC7.HR.sternum.AP, walk.GC8.HR.sternum.AP, walk.GC9.HR.sternum.AP];
        walk.HR.sternum.ML = [walk.GC1.HR.sternum.ML, walk.GC2.HR.sternum.ML,
walk.GC3.HR.sternum.ML,...
            walk.GC4.HR.sternum.ML, walk.GC5.HR.sternum.ML, walk.GC6.HR.sternum.ML,...
            walk.GC7.HR.sternum.ML, walk.GC8.HR.sternum.ML, walk.GC9.HR.sternum.ML];
        walk.HR.sternum.VT = [walk.GC1.HR.sternum.VT, walk.GC2.HR.sternum.VT,
walk.GC3.HR.sternum.VT,...
            walk.GC4.HR.sternum.VT, walk.GC5.HR.sternum.VT, walk.GC6.HR.sternum.VT,...
            walk.GC7.HR.sternum.VT, walk.GC8.HR.sternum.VT, walk.GC9.HR.sternum.VT];
        % Every axis Lumbar's HR
        walk.HR.lumbar.AP = [walk.GC1.HR.lumbar.AP, walk.GC2.HR.lumbar.AP,
walk.GC3.HR.lumbar.AP,...
            walk.GC4.HR.lumbar.AP, walk.GC5.HR.lumbar.AP, walk.GC6.HR.lumbar.AP,...
            walk.GC7.HR.lumbar.AP, walk.GC8.HR.lumbar.AP, walk.GC9.HR.lumbar.AP];
        walk.HR.lumbar.ML = [walk.GC1.HR.lumbar.ML, walk.GC2.HR.lumbar.ML,
walk.GC3.HR.lumbar.ML,...
            walk.GC4.HR.lumbar.ML, walk.GC5.HR.lumbar.ML, walk.GC6.HR.lumbar.ML,...
            walk.GC7.HR.lumbar.ML, walk.GC8.HR.lumbar.ML, walk.GC9.HR.lumbar.ML];
        walk.HR.lumbar.VT = [walk.GC1.HR.lumbar.VT, walk.GC2.HR.lumbar.VT,
walk.GC3.HR.lumbar.VT,...
            walk.GC4.HR.lumbar.VT, walk.GC5.HR.lumbar.VT, walk.GC6.HR.lumbar.VT,...
            walk.GC7.HR.lumbar.VT, walk.GC8.HR.lumbar.VT, walk.GC9.HR.lumbar.VT];
    case 10
        % Sternum HR
        walk.GC1.HR.sternum  = getHR(walk.GC1.sternum.acc);
        walk.GC2.HR.sternum  = getHR(walk.GC2.sternum.acc);
```

```
        walk.GC3.HR.sternum  = getHR(walk.GC3.sternum.acc);
        walk.GC4.HR.sternum  = getHR(walk.GC4.sternum.acc);
        walk.GC5.HR.sternum  = getHR(walk.GC5.sternum.acc);
        walk.GC6.HR.sternum  = getHR(walk.GC6.sternum.acc);
        walk.GC7.HR.sternum  = getHR(walk.GC7.sternum.acc);
        walk.GC8.HR.sternum  = getHR(walk.GC8.sternum.acc);
        walk.GC9.HR.sternum  = getHR(walk.GC9.sternum.acc);
        walk.GC10.HR.sternum  = getHR(walk.GC10.sternum.acc);
        % Lumbar HR
        walk.GC1.HR.lumbar  = getHR(walk.GC1.lumbar.acc);
        walk.GC2.HR.lumbar  = getHR(walk.GC2.lumbar.acc);
        walk.GC3.HR.lumbar  = getHR(walk.GC3.lumbar.acc);
        walk.GC4.HR.lumbar  = getHR(walk.GC4.lumbar.acc);
        walk.GC5.HR.lumbar  = getHR(walk.GC5.lumbar.acc);
        walk.GC6.HR.lumbar  = getHR(walk.GC6.lumbar.acc);
        walk.GC7.HR.lumbar  = getHR(walk.GC7.lumbar.acc);
        walk.GC8.HR.lumbar  = getHR(walk.GC8.lumbar.acc);
        walk.GC9.HR.lumbar  = getHR(walk.GC9.lumbar.acc);
        walk.GC10.HR.lumbar  = getHR(walk.GC10.lumbar.acc);
        % Every axis Sternum's HR
        walk.HR.sternum.AP = [walk.GC1.HR.sternum.AP, walk.GC2.HR.sternum.AP,
walk.GC3.HR.sternum.AP,...
            walk.GC4.HR.sternum.AP, walk.GC5.HR.sternum.AP, walk.GC6.HR.sternum.AP, ...
            walk.GC7.HR.sternum.AP, walk.GC8.HR.sternum.AP, walk.GC9.HR.sternum.AP, ...
            walk.GC10.HR.sternum.AP];
        walk.HR.sternum.ML = [walk.GC1.HR.sternum.ML, walk.GC2.HR.sternum.ML,
walk.GC3.HR.sternum.ML,...
            walk.GC4.HR.sternum.ML, walk.GC5.HR.sternum.ML, walk.GC6.HR.sternum.ML,...
            walk.GC7.HR.sternum.ML, walk.GC8.HR.sternum.ML, walk.GC9.HR.sternum.ML, ...
            walk.GC10.HR.sternum.ML];
        walk.HR.sternum.VT = [walk.GC1.HR.sternum.VT, walk.GC2.HR.sternum.VT,
walk.GC3.HR.sternum.VT,...
            walk.GC4.HR.sternum.VT, walk.GC5.HR.sternum.VT, walk.GC6.HR.sternum.VT,...
            walk.GC7.HR.sternum.VT, walk.GC8.HR.sternum.VT, walk.GC9.HR.sternum.VT, ...
            walk.GC10.HR.sternum.VT];
        % Every axis Lumbar's HR
        walk.HR.lumbar.AP = [walk.GC1.HR.lumbar.AP, walk.GC2.HR.lumbar.AP,
walk.GC3.HR.lumbar.AP,...
            walk.GC4.HR.lumbar.AP, walk.GC5.HR.lumbar.AP, walk.GC6.HR.lumbar.AP,...
            walk.GC7.HR.lumbar.AP, walk.GC8.HR.lumbar.AP, walk.GC9.HR.lumbar.AP, ...
            walk.GC10.HR.lumbar.AP];
        walk.HR.lumbar.ML = [walk.GC1.HR.lumbar.ML, walk.GC2.HR.lumbar.ML,
walk.GC3.HR.lumbar.ML,...
            walk.GC4.HR.lumbar.ML, walk.GC5.HR.lumbar.ML, walk.GC6.HR.lumbar.ML,...
            walk.GC7.HR.lumbar.ML, walk.GC8.HR.lumbar.ML, walk.GC9.HR.lumbar.ML, ...
            walk.GC10.HR.lumbar.ML];
        walk.HR.lumbar.VT = [walk.GC1.HR.lumbar.VT, walk.GC2.HR.lumbar.VT,
walk.GC3.HR.lumbar.VT,...
            walk.GC4.HR.lumbar.VT, walk.GC5.HR.lumbar.VT, walk.GC6.HR.lumbar.VT,...
            walk.GC7.HR.lumbar.VT, walk.GC8.HR.lumbar.VT, walk.GC9.HR.lumbar.VT, ...
            walk.GC10.HR.lumbar.VT];
end
% Getting avg per dataset
walk.avg.HR.sternum.AP = mean(walk.HR.sternum.AP);
walk.avg.HR.sternum.ML = mean(walk.HR.sternum.ML);
walk.avg.HR.sternum.VT = mean(walk.HR.sternum.VT);
walk.avg.HR.lumbar.AP = mean(walk.HR.sternum.AP);
walk.avg.HR.lumbar.ML = mean(walk.HR.sternum.ML);
walk.avg.HR.lumbar.VT = mean(walk.HR.sternum.VT);
end
% Compare HR from the whole Gait
function struct = getHRPerDataset(struct)
% --- Input ---
% Wi: Struct data from the i-th Walk

% HR data
struct.W1 = getHRPerWalk(struct.W1);
struct.W2 = getHRPerWalk(struct.W2);
```

```
struct.W3 = getHRPerWalk(struct.W3);
struct.W4 = getHRPerWalk(struct.W4);
struct.W5 = getHRPerWalk(struct.W5);
struct.W6 = getHRPerWalk(struct.W6);

% Mean values per dataset
struct.HR.sternum.AP = [struct.W1.avg.HR.sternum.AP, struct.W2.avg.HR.sternum.AP,
struct.W3.avg.HR.sternum.AP, ...
    struct.W4.avg.HR.sternum.AP, struct.W5.avg.HR.sternum.AP, struct.W6.avg.HR.sternum.AP];
struct.HR.sternum.ML = [struct.W1.avg.HR.sternum.ML, struct.W2.avg.HR.sternum.ML,
struct.W3.avg.HR.sternum.ML, ...
    struct.W4.avg.HR.sternum.ML, struct.W5.avg.HR.sternum.ML, struct.W6.avg.HR.sternum.ML];
struct.HR.sternum.VT = [struct.W1.avg.HR.sternum.VT, struct.W2.avg.HR.sternum.VT,
struct.W3.avg.HR.sternum.VT, ...
    struct.W4.avg.HR.sternum.VT, struct.W5.avg.HR.sternum.VT, struct.W6.avg.HR.sternum.VT];
struct.avg.HR.sternum.AP = mean(struct.HR.sternum.AP);
struct.avg.HR.sternum.ML = mean(struct.HR.sternum.ML);
struct.avg.HR.sternum.VT = mean(struct.HR.sternum.VT);
struct.HR.lumbar.AP = [struct.W1.avg.HR.lumbar.AP, struct.W2.avg.HR.lumbar.AP,
struct.W3.avg.HR.lumbar.AP, ...
    struct.W4.avg.HR.lumbar.AP, struct.W5.avg.HR.lumbar.AP, struct.W6.avg.HR.lumbar.AP];
struct.HR.lumbar.ML = [struct.W1.avg.HR.lumbar.ML, struct.W2.avg.HR.lumbar.ML,
struct.W3.avg.HR.lumbar.ML, ...
    struct.W4.avg.HR.lumbar.ML, struct.W5.avg.HR.lumbar.ML, struct.W6.avg.HR.lumbar.ML];
struct.HR.lumbar.VT = [struct.W1.avg.HR.lumbar.VT, struct.W2.avg.HR.lumbar.VT,
struct.W3.avg.HR.lumbar.VT, ...
    struct.W4.avg.HR.lumbar.VT, struct.W5.avg.HR.lumbar.VT, struct.W6.avg.HR.lumbar.VT];
struct.avg.HR.lumbar.AP = mean(struct.HR.lumbar.AP);
struct.avg.HR.lumbar.ML = mean(struct.HR.lumbar.ML);
struct.avg.HR.lumbar.VT = mean(struct.HR.lumbar.VT);
end
%% 5. Absolute angles (AA)
function dataset = getAbsoluteAnglesPerDataset(dataset)
dataset.W1 = getAbsoluteAnglesPerWalk(dataset, 1);
dataset.W2 = getAbsoluteAnglesPerWalk(dataset, 2);
dataset.W3 = getAbsoluteAnglesPerWalk(dataset, 3);
dataset.W4 = getAbsoluteAnglesPerWalk(dataset, 4);
dataset.W5 = getAbsoluteAnglesPerWalk(dataset, 5);
dataset.W6 = getAbsoluteAnglesPerWalk(dataset, 6);
% All dataset avgs
% sternum data
dataset.AA.sternum.roll = [dataset.W1.sternum.roll,dataset.W2.sternum.roll, ...
    dataset.W3.sternum.roll,dataset.W4.sternum.roll, ...
    dataset.W5.sternum.roll,dataset.W6.sternum.roll]';
dataset.AA.sternum.pitch = [dataset.W1.sternum.pitch,dataset.W2.sternum.pitch, ...
    dataset.W3.sternum.pitch,dataset.W4.sternum.pitch, ...
    dataset.W5.sternum.pitch,dataset.W6.sternum.pitch]';
dataset.AA.sternum.yaw = [dataset.W1.sternum.yaw, dataset.W2.sternum.yaw, ...
    dataset.W3.sternum.yaw, dataset.W4.sternum.yaw, ...
    dataset.W5.sternum.yaw, dataset.W6.sternum.yaw]';
dataset.AA.sternum.avg.roll = [dataset.W1.avg.sternum.roll; dataset.W2.avg.sternum.roll;
    dataset.W3.avg.sternum.roll; dataset.W4.avg.sternum.roll;
    dataset.W5.avg.sternum.roll; dataset.W6.avg.sternum.roll];
dataset.AA.sternum.avg.pitch = [dataset.W1.avg.sternum.pitch; dataset.W2.avg.sternum.pitch;
    dataset.W3.avg.sternum.pitch; dataset.W4.avg.sternum.pitch;
    dataset.W5.avg.sternum.pitch; dataset.W6.avg.sternum.pitch];
dataset.AA.sternum.avg.yaw = [dataset.W1.avg.sternum.yaw; dataset.W2.avg.sternum.yaw;
    dataset.W3.avg.sternum.yaw; dataset.W4.avg.sternum.yaw;
    dataset.W5.avg.sternum.yaw; dataset.W6.avg.sternum.yaw];
dataset.avg.sternum.roll = mean(dataset.AA.sternum.roll);
dataset.avg.sternum.pitch = mean(dataset.AA.sternum.pitch);
dataset.avg.sternum.yaw = mean (dataset.AA.sternum.yaw);
dataset.max.sternum.roll = max(dataset.AA.sternum.roll);
dataset.max.sternum.pitch = max(dataset.AA.sternum.pitch);
dataset.max.sternum.yaw = max(dataset.AA.sternum.yaw);
dataset.min.sternum.roll = min(dataset.AA.sternum.roll);
dataset.min.sternum.pitch = min(dataset.AA.sternum.pitch);
dataset.min.sternum.yaw = min(dataset.AA.sternum.yaw);
```

```
% lumbar data
dataset.AA.lumbar.roll = [dataset.W1.lumbar.roll,dataset.W2.lumbar.roll, ...
    dataset.W3.lumbar.roll,dataset.W4.lumbar.roll, ...
    dataset.W5.lumbar.roll,dataset.W6.lumbar.roll]';
dataset.AA.lumbar.pitch = [dataset.W1.lumbar.pitch,dataset.W2.lumbar.pitch, ...
    dataset.W3.lumbar.pitch,dataset.W4.lumbar.pitch, ...
    dataset.W5.lumbar.pitch,dataset.W6.lumbar.pitch]';
dataset.AA.lumbar.yaw = [dataset.W1.lumbar.yaw,dataset.W2.lumbar.yaw, ...
    dataset.W3.lumbar.yaw,dataset.W4.lumbar.yaw, ...
    dataset.W5.lumbar.yaw,dataset.W6.lumbar.yaw]';
dataset.AA.lumbar.avg.roll = [dataset.W1.avg.sternum.roll; dataset.W2.avg.sternum.roll;
    dataset.W3.avg.sternum.roll; dataset.W4.avg.sternum.roll;
    dataset.W5.avg.sternum.roll; dataset.W6.avg.sternum.roll];
dataset.AA.lumbar.avg.pitch = [dataset.W1.avg.sternum.pitch; dataset.W2.avg.sternum.pitch;
    dataset.W3.avg.sternum.pitch; dataset.W4.avg.sternum.pitch;
    dataset.W5.avg.sternum.pitch; dataset.W6.avg.sternum.pitch];
dataset.AA.lumbar.avg.yaw = [dataset.W1.avg.sternum.yaw; dataset.W2.avg.sternum.yaw;
    dataset.W3.avg.sternum.yaw; dataset.W4.avg.sternum.yaw;
    dataset.W5.avg.sternum.yaw; dataset.W6.avg.sternum.yaw];
dataset.avg.lumbar.roll = mean(dataset.AA.lumbar.roll);
dataset.avg.lumbar.pitch = mean(dataset.AA.lumbar.pitch);
dataset.avg.lumbar.yaw = mean(dataset.AA.lumbar.yaw);
dataset.max.lumbar.roll = max(dataset.AA.lumbar.roll);
dataset.max.lumbar.pitch = max(dataset.AA.lumbar.pitch);
dataset.max.lumbar.yaw = max(dataset.AA.lumbar.yaw);
dataset.min.lumbar.roll = min(dataset.AA.lumbar.roll);
dataset.min.lumbar.pitch = min(dataset.AA.lumbar.pitch);
dataset.min.lumbar.yaw = min(dataset.AA.lumbar.yaw);
end
function W = getAbsoluteAnglesPerWalk(struct, Walk)
preproc = libPreProcessing;
% Choosing walks
switch Walk
    case 1; W = struct.W1; case 2; W = struct.W2; case 3; W = struct.W3;
    case 4; W = struct.W4; case 5; W = struct.W5; case 6; W = struct.W6;
end
ini = 1; % Cut values (from first HS to the last)
if length(W.sternum.roll) > W.HS.Frames(end)
    range = [W.HS.Frames(ini) W.HS.Frames(end-1)];
    W = preproc.cutHSData(W, range);
end
% Calculation: mean, max, min, initial, final
% Roll values
W.avg.sternum.roll = mean(W.sternum.roll);
W.max.sternum.roll = max(W.sternum.roll);
W.min.sternum.roll = min(W.sternum.roll);
W.avg.lumbar.roll = mean(W.lumbar.roll);
W.max.lumbar.roll = max(W.lumbar.roll);
W.min.lumbar.roll = min(W.lumbar.roll);
% Pitch values
W.avg.sternum.pitch = mean(W.sternum.pitch);
W.max.sternum.pitch = max(W.sternum.pitch);
W.min.sternum.pitch = min(W.sternum.pitch);
W.avg.lumbar.pitch = mean(W.lumbar.pitch);
W.max.lumbar.pitch = max(W.lumbar.pitch);
W.min.lumbar.pitch = min(W.lumbar.pitch);
% Yaw values
W.avg.sternum.yaw = mean(W.sternum.yaw);
W.max.sternum.yaw = max(W.sternum.yaw);
W.min.sternum.yaw = min(W.sternum.yaw);
W.avg.lumbar.yaw = mean(W.lumbar.yaw);
W.max.lumbar.yaw = max(W.lumbar.yaw);
W.min.lumbar.yaw = min(W.lumbar.yaw);
end
%% 6. Relative angles (RA)
function rel = getRelativeAnglesTrunk(struct, reference)
qfun = libQuaternions; % Calling quaternions library
qLumbar = struct.lumbar.q; qSternum = struct.sternum.q; % Quaternion arrays
```

```
qRelAngle = zeros(size(qSternum)); % Output
for j=1:size(qRelAngle,1)
    if reference == "lumbar"
        qConj = qfun.quaternionConj(qLumbar(j,:));
        qRelAngle(j,:) = qfun.quaternionProduct(qSternum(j,:), qConj);
    elseif reference == "sternum"
        qConj = qfun.quaternionConj(qSternum(j,:));
        qRelAngle(j,:) = qfun.quaternionProduct(qLumbar(j,:), qConj);
    end
end
% Converting quaternion to RPY euler angles
rpy = rad2deg(quat2eul(qRelAngle,'XYZ'));
rel.roll = rpy(:,1); rel.pitch = rpy(:,2); rel.yaw = rpy(:,3);
rel.avg.roll = mean(rel.roll); rel.avg.pitch = mean(rel.pitch); rel.avg.yaw = mean(rel.yaw);
end
function struct = getRelativeAnglesPerDataset(struct)
reference = "lumbar";
struct.W1.relAngles = getRelativeAnglesTrunk(struct.W1, reference);
struct.W2.relAngles = getRelativeAnglesTrunk(struct.W2, reference);
struct.W3.relAngles = getRelativeAnglesTrunk(struct.W3, reference);
struct.W4.relAngles = getRelativeAnglesTrunk(struct.W4, reference);
struct.W5.relAngles = getRelativeAnglesTrunk(struct.W5, reference);
struct.W6.relAngles = getRelativeAnglesTrunk(struct.W6, reference);
struct.relAngles.roll = [struct.W1.relAngles.roll; struct.W2.relAngles.roll;
    struct.W3.relAngles.roll; struct.W5.relAngles.roll;
    struct.W5.relAngles.roll; struct.W6.relAngles.roll];
struct.relAngles.avg.roll = [struct.W1.relAngles.avg.roll; struct.W2.relAngles.avg.roll;
    struct.W3.relAngles.avg.roll; struct.W5.relAngles.avg.roll;
    struct.W5.relAngles.avg.roll; struct.W6.relAngles.avg.roll];
struct.relAngles.pitch = [struct.W1.relAngles.pitch; struct.W2.relAngles.pitch;
    struct.W3.relAngles.pitch; struct.W5.relAngles.pitch;
    struct.W5.relAngles.pitch; struct.W6.relAngles.pitch];
struct.relAngles.avg.pitch = [struct.W1.relAngles.avg.pitch; struct.W2.relAngles.avg.pitch;
    struct.W3.relAngles.avg.pitch; struct.W5.relAngles.avg.pitch;
    struct.W5.relAngles.avg.pitch; struct.W6.relAngles.avg.pitch];
struct.relAngles.yaw = [struct.W1.relAngles.yaw; struct.W2.relAngles.yaw;
    struct.W3.relAngles.yaw; struct.W5.relAngles.yaw;
    struct.W5.relAngles.yaw; struct.W6.relAngles.yaw];
struct.relAngles.avg.yaw = [struct.W1.relAngles.avg.yaw; struct.W2.relAngles.avg.yaw;
    struct.W3.relAngles.avg.yaw; struct.W5.relAngles.avg.yaw;
    struct.W5.relAngles.avg.yaw; struct.W6.relAngles.avg.yaw];
struct.avg.roll = mean(struct.relAngles.roll);
struct.avg.pitch = mean(struct.relAngles.pitch);
struct.avg.yaw = mean(struct.relAngles.yaw);
struct.max.roll = max(struct.relAngles.roll);
struct.max.pitch = max(struct.relAngles.pitch);
struct.max.yaw = max(struct.relAngles.yaw);
struct.min.roll = min(struct.relAngles.roll);
struct.min.pitch = min(struct.relAngles.pitch);
struct.min.yaw = min(struct.relAngles.yaw);
end
```

G.5 Quaternions library

```
%% Quaternions functions library
% Author: Maria Alejandra Guzman Alfaro
% Last updated: 01.11.2024
function funs = libQuaternions
funs.create_quaternion=@create_quaternion;
funs.quaternionNorm=@quaternionNorm;
funs.quaternionInv=@quaternionInv;
funs.quaternionConj=@quaternionConj;
funs.quaternionProduct=@quaternionProduct;
funs.imuDataRotation=@imuDataRotation;
end
%% Creation of quaterion given an angle and axis
function q = create_quaternion(ang,v)
% Input:
% - ang: angle of rotation in degrees
% - v: axis of rotation
q = [cosd(ang/2), sind(ang/2)*v];
end
%% Norm of quaternions
function norm = quaternionNorm(q)
norm = q/sqrt(q(1).^2 + q(2).^2 + q(3).^2 + q(4).^2);
end
%% Invertion of quaternions
function q_inv = quaternionInv(q)
w = q(1); x = q(2); y = q(3); z = q(4);
norm_quat = sqrt(w^2 + x^2 + y^2 + z^2);
q_inv = [w, -x, -y, -z] / norm_quat^2;
end
%% Conjugation of quaternions
function norm = quaternionConj(q)
norm = [q(1) -q(2) -q(3) -q(4)];
end
%% Product of quaternions
function product = quaternionProduct(q1, q2)
w1 = q1(1); x1 = q1(2); y1 = q1(3); z1 = q1(4);
w2 = q2(1); x2 = q2(2); y2 = q2(3); z2 = q2(4);
% Getting the product of both quaternions
product = [w1*w2 - x1*x2 - y1*y2 - z1*z2, ...
           w1*x2 + x1*w2 + y1*z2 - z1*y2, ...
           w1*y2 - x1*z2 + y1*w2 + z1*x2, ...
           w1*z2 + x1*y2 - y1*x2 + z1*w2];
end
%% Rotation of data by means of quaternions
function [rot_acc, rot_gyr, rot_mag] = imuDataRotation(acc, gyr, mag, qIni)
% Inputs:
% - initial_q: Initial quaternion (condition given by IMU frame)
% - acc: Acceleration struct
% - gyr: Angular velocity struct
% - mag: Magnetic field struct
% IMU frame (v_a)
angVel = [zeros(size(gyr, 1), 1), gyr]; accel = [zeros(size(acc, 1), 1), acc];
magne = [zeros(size(mag, 1), 1), mag];
% Initial quaternion rotation
q = qIni';
q_conj = quaternionConj(q);
% Rotation through a global frame (v_b)
rot_gyr = zeros(size(angVel)); rot_acc = zeros(size(accel)); rot_mag = zeros(size(magne));
for i = 1:size(angVel, 1)
    rot_gyr(i, :) = quaternionProduct(quaternionProduct(q, angVel(i,:)), q_conj);
    rot_acc(i, :) = quaternionProduct(quaternionProduct(q, accel(i,:)), q_conj);
    rot_mag(i, :) = quaternionProduct(quaternionProduct(q, magne(i,:)), q_conj);
end
% Remove the first column from the new data
rot_gyr = rot_gyr(:, 2:4); rot_acc = rot_acc(:, 2:4); rot_mag = rot_mag(:, 2:4);
end
```