HuMoD Database

A versatile and open database for the investigation, modeling and simulation of human motion dynamics

Documentation

Janis Wojtusch

TECHNISCHE UNIVERSITÄT DARMSTADT

Documentation 1.4

December 22, 2017

Janis Wojtusch

wojtusch@sim.tu-darsmtadt.de

Technische Universität Darmstadt
Department of Computer Science
Simulation, Systems Optimization
and Robotics Group
Hochschulstraße 10
64289 Darmstadt
Germany

Contents

1 Introduction

The HuMoD Database, derived from **Hu**man **Mo**tion **D**ynamics, is a versatile and open database for the investigation, modeling and simulation of human motion dynamics with a focus on lower limbs. The database contains raw and processed biomechanical measurement data from a three-dimensional motion capture system, an instrumented treadmill and an electromyographical measurement system for eight different motion tasks performed by a female and male subject as well as anthropometric parameters for both subjects. The quite unique combination of biomechanical measurement data with anthropometric parameters allows to create biomechanical models of the human locomotor system and to investigate and simulate human motion dynamics including muscle driven actuation. Besides investigations in biomechanics, the database can be of value especially for the design and development of musculoskeletal humanoid robots and for better understanding and benchmarking human-like robot locomotion. The biomechanical measurement data and the source code of the applied computational scripts is open and can be obtained free of charge from the HuMoD Database website:

`http://www.sim.informatik.tu-darmstadt.de/humod/`

 Please cite the following publication, if you are using processed or raw data or computational scripts provided in the context of the HuMoD Database in your research:

> *J. Wojtusch and O. von Stryk (2015).* HuMoD - A Versatile and Open Database for the Investigation, Modeling and Simulation of Human Motion Dynamics on Actuation Level. *In Proceedings of the IEEE-RAS International Conference on Humanoid Robots (pp. 74 – 79).*

Some texts, tables and figures in this documentation are taken from or are based on this publication.

2 Subjects

A healthy female and male subject performed eight different motion tasks without shoes dressed in underwear. The subjects were given time to become familiar with the measurement setup and equipment before the measurements and to rest between the trials. The measurement procedure was reviewed and approved by the ethical review committee of Friedrich-Schiller-Universität Jena, Germany. Both subjects provided informed consent in accordance with the policies of the ethical review committee. Table 1 lists some details of the subjects.

Table 1: Details of the female and male subject.

	Subject A	**Subject B**
Gender	female	male
Age	27 yrs	32 yrs
Height	161 cm	179 cm
Weight	57.3 kg	84.8 kg
Origin	Central Europe	Central Europe
Clothing	underpants, sports bra	underpants
Date	April 2014	November 2014

3 Motion Protocol

The subjects performed eight motion tasks, partially at different speeds or under changed conditions resulting in thirteen trials. The motion tasks cover locomotion, interaction with an object and physical activity representing a sample of typical repetitive tasks and goal-oriented tasks useful for biomechanics and humanoid robotics research. These include walking, running, squatting and jumping as well as avoiding obstacles and kicking a ball. During the first and last 10 s of each trial the force plates of the instrumented treadmill remained unloaded. Before and after performing the particular motion task, the subject stood still on the treadmill for at least 10 s. This idle time was increased to 20 s after fast motion tasks. Details of the single trials are summarized in Table 2.

Table 2: Details of the motion tasks.

#	Description	Initial idle time	Task duration	Final idle time
1.1	Straight walking at 1.0 $\frac{m}{s}$	10 s + 10 s	60 s	10 s + 10 s
1.2	Straight walking at 1.5 $\frac{m}{s}$	10 s + 10 s	60 s	10 s + 10 s
1.3	Straight walking at 2.0 $\frac{m}{s}$	10 s + 10 s	60 s	10 s + 10 s
2.1	Straight running at 2.0 $\frac{m}{s}$	10 s + 10 s	60 s	20 s + 10 s
2.2	Straight running at 3.0 $\frac{m}{s}$	10 s + 10 s	60 s	20 s + 10 s
2.3	Straight running at 4.0 $\frac{m}{s}$	10 s + 10 s	60 s	20 s + 10 s
3	Sideways walking at 0.5 $\frac{m}{s}$	10 s + 10 s	60 s	10 s + 10 s
4	Transition between standing and straight running at 4.0 $\frac{m}{s}$*	10 s + 10 s	112 s	10 s + 10 s
5.1	Avoiding a long box obstacle (41 × 20 × 15 cm) at 1.0 $\frac{m}{s}$	10 s + 10 s	120 s	10 s + 10 s
5.2	Avoiding a wide box obstacle (20 × 41 × 15 cm) at 1.0 $\frac{m}{s}$	10 s + 10 s	120 s	10 s + 10 s
6	Continuous squats with arms akimbo and stopped treadmill	10 s + 10 s	40 s	10 s + 10 s
7	Kicking a soft football (20 cm, 160 g) with stopped treadmill	10 s + 10 s	100 s	10 s + 10 s
8	Continuous jumps with arms akimbo and stopped treadmill	10 s + 10 s	20 s	10 s + 10 s

* Transition between standing and straight running comprised accelerating from 0.0 $\frac{m}{s}$ to 4.0 $\frac{m}{s}$ at 0.1 $\frac{m}{s^2}$, holding 4.0 $\frac{m}{s}$ for 20 s and decelerating from 4.0 $\frac{m}{s}$ to 0.0 $\frac{m}{s}$ at -0.1 $\frac{m}{s^2}$.

4 Measurement Setup

The measurements were collected at the Locomotion Lab of André Seyfarth at Technische Universität Darmstadt, Germany. All trials were performed on the instrumented treadmill ADAL3D-WR (Tecmachine,

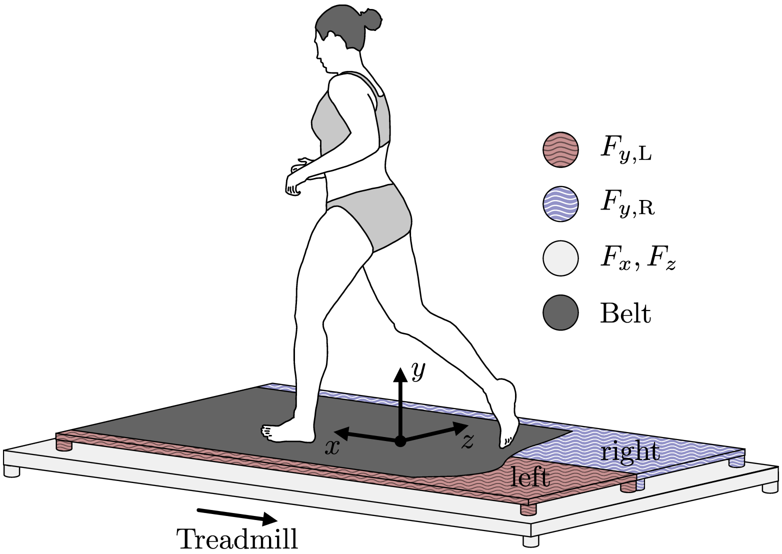


Figure 1: Schematic diagram of the instrumented treadmill.

France). The belt of the treadmill runs over two force plates with four single-axis force sensors (Kistler, Switzerland) that were used to measure the vertical ground reaction forces F_y of the left and right foot. The two force plates are mounted on top of four multi-axis force sensors (Kistler, Switzerland) that measured lateral forces F_x and F_z. All forces were recorded at 1000 Hz. The software ADIMIX Walking 2.0 was used to control the treadmill and store the recorded force data. Figure 1 shows a schematic diagram of the instrumented treadmill and the single- and multi-axis force sensors. Detailed dimensions of the instrumented treadmill are provided in Appendix A.

The motion of upper and lower limbs was recorded at 500 Hz with a three-dimensional motion capture system consisting of four Oqus 310+ cameras and six Oqus 300+ cameras (Qualisys, Sweden). A set of thirty-five reflective markers with a diameter of 19 mm mounted on thin cardboard was placed on the skin at anatomical landmarks by an experienced examiner. One additional reflective marker was placed on top of the underpants above pubic symphysis [Reed1999]. Figure 2a illustrates the locations of the thirty-six reflective markers. A description of the associated landmarks and used abbreviations is given in Appendix B. For calibrating and controlling the motion capture system, storing the recorded motion data as well as assigning the recorded motion data to the individual markers, the software TrackManager 2.7 was applied.

The electrical activity of fourteen selected skeletal muscles in the legs was recorded at 2000 Hz with the electromyographical measurement system Bagnoli-16 Desktop (Delsys, USA). The measured signals were internally filtered to a bandwidth between 20 Hz and 450 Hz. The set of fourteen surface electrodes was placed according to SENIAM guidelines [Hermens2000] by an experienced examiner. Figure 2b shows the locations of the fourteen surface electrodes for electromyographical measurement. The associated muscles and used abbreviations are listed in Appendix C. The software EMGworks Acquisition 3.6 was used to control the electromyographical measurement system and store the recorded activity data.

5 Data Processing

The raw data measured with the motion capture system, instrumented treadmill and electromyographical measurement system was exported into the MAT file format and processed with the numerical computing software MATLAB (MathWorks, USA) in order to provide additional information for the investigation, modeling and simulation of human motion dynamics.

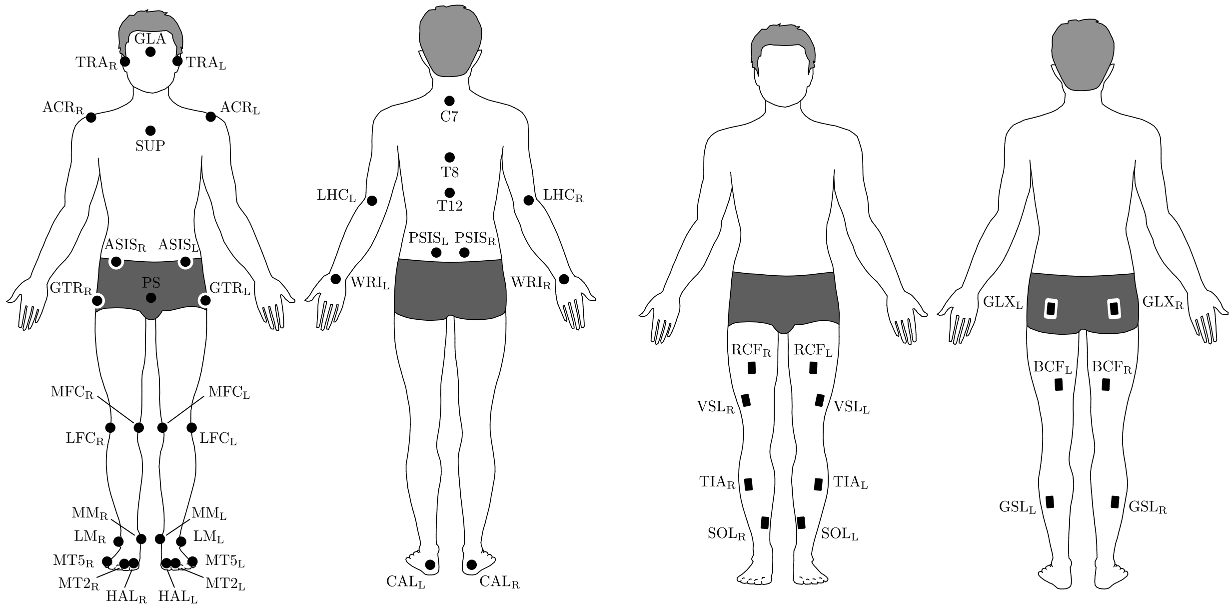


Figure 2: Locations of the thirty-six reflective markers for motion capture in (a) and the fourteen electrodes for electromyographical measurement in (b).

Raw motion and force data was synchronized by compensating temporal offset and drift as well as transforming the global reference frame of the motion capture system into the global reference frame of the instrumented treadmill considering the ISB recommendations for reference frame notation [Wu1995]. Figure 1 illustrates the applied global reference frame where the origin is located at the center of the rectangle spanned by the left and right force plates projected to the top of the belt surface.

Infrequent gaps in the raw kinematic motion data of up to 300 ms resulting from temporarily covered reflective markers were filled by applying polynomial approximations. The measured spatial positions of the reflective markers were then shifted to the approximated skin surface. This was achieved by approximating a normal vector perpendicular to the skin surface pointing towards the considered reflective marker from adjacent reflective markers and estimated joint centers. The normalized normal vector was multiplied with the reflective marker radius and additional support material thickness and subtracted from the measured spatial position.

GLA	The normal vector is parallel to the line connecting the midpoint between the TRA markers with the GLA marker.
TRA	The normal vector is parallel to the line connecting the TRA markers.
SUP, C7	The normal vector is parallel to the line connecting the C7 and SUP markers.
T8	The normal vector is parallel to vector sum of the normal vectors specified for the SUP, C7 and T12 markers.
T12	The normal vector is parallel to the line connecting the T8 and T12 markers rotated by $\frac{\pi}{2}$ rad about the line connecting the ACR markers.
ACR	The normal vector is perpendicular to the normal vector specified for the SUP and C7 markers and the line connecting the ACR markers.
LHC	The normal vector is perpendicular to the lines connecting the WRI and LHC markers as well as the estimated shoulder joint center and LHC marker.

ASIS, PSIS	The normal vector is parallel to the line connecting the midpoint between the ASIS markers with the midpoint between the PSIS markers.
PS	The normal vector is parallel to the line connecting the midpoint between the PSIS markers with the PS marker.
GTR	The normal vector is parallel to the line connecting the GTR markers.
LFC, MFC	The normal vector is parallel to the line connecting the LFC and MFC markers.
LM, MM	The normal vector is parallel to the line connecting the LM and MM markers.
CAL	The normal vector is parallel to the line connecting the CAL and MT2 markers
MT2, MT5, HAL	The normal vector is perpendicular to the lines connecting the CAL and MT5 markers as well as the MT2 and MT5 markers.

The shifted spatial positions of the reflective markers were then used to estimate the joint centers of fifteen joints in arms, trunk, pelvis and legs by applying established regressing equations. A description of the used abbreviations is given in Appendix D.

LNJ	The lower neck joint center was estimated from the C7, SUP and ACR markers according to Reed et al. [Reed1999].
SJ_L, SJ_R	The shoulder joint centers were estimated from the C7, SUP and ACR markers according to Reed et al. [Reed1999].
EJ_L, EJ_R	The elbow joint centers were estimated from the WRI and LHC markers as well as the estimated shoulder joint centers according to Reed et al. [Reed1999].
ULJ	The upper lumbar joint center was estimated from the C7, T8, T12, SUP and ACR markers according to Reed et al. [Reed1999] and Dumas et al. [Dumas2015].
LLJ	The lower lumbar joint center was estimated from the ASIS, PSIS and PS markers according to Reed et al. [Reed1999].
HJ_L, HJ_R	The hip joint centers were estimated from the ASIS, PSIS and PS markers according to Harrington et al. [Harrington2007].
KJ_L, KJ_R	The knee joint centers were estimated from the LFC and MFC markers according to Dumas et al. [Dumas2007a].
AJ_L, AJ_R	The ankle joint centers were estimated from the LM and MM markers according to Dumas et al. [Dumas2007a].
TJ_L, TJ_R	The toe joint centers were estimated from the CAL, MT2 and MT5 markers based on definitions by Zatsiorsky [Zatsiorsky1998].

Additional regression equations from literature for hip, knee and ankle joints were implemented and can be used alternatively by applying the provided computational scripts [Reed1999; Leardini1999; Seidel1995; Davis1991; Bell1990; Dempster1955; Hicks1953].

For the estimation of the joint trajectories including joint positions, velocities and accelerations, a Kalman smoother in combination with a subject-specific forward kinematics model with thirty degrees of freedom and fourteen body segments was applied [DeGroote2008; Yu2004]. This approach allows to reduce the influences of instrumental errors and soft tissue artifacts. The forward kinematics model consists of a head, thorax and abdomen segment, two upper and lower arm segments, a pelvis segment and two thigh, shank and foot segments. The model structure is shown in Figure 3. The joint trajectories are given as Tait–Bryan angles in x-y'-z'' convention.

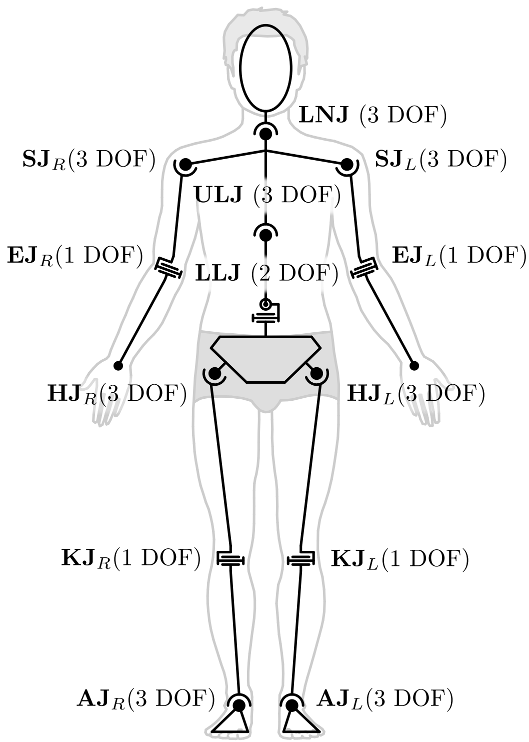


Figure 3: Forward kinematics model with thirty degrees of freedom.

The raw ground reaction force data was filtered using a sixth order zero-lag low-pass filter with a cut-off frequency of 50 Hz. In order to decompose the measured lateral ground reaction forces F_x and F_z and the measured vertical ground reaction force F_y in the event of mixed force plate contact during double support phase for the locomotion trials, parametrized transition functions determined using a multiple regression analysis were applied [Villeger2014]. The transition functions approximate the ground reaction force decrease of the foot leaving the ground during double support phase. The ground reaction force data was then used to estimate the center of pressure and detect individual events like left and right steps, squats or kicks.

The raw muscle activity data was rectified and filtered using a root-mean square filter with a window size of 300 ms [Konrad2005]. In addition, the filtered muscle activity data was normalized to the maximum activity level over all trials of the subject. Each dataset provides filtered and non-normalized as well as filtered and normalized muscle activity data.

Subject-specific anthropometric parameters including body segment masses, centers of mass as well as moments and products of inertia were estimated with linear regression equations [Dumas2007a; Dumas2007b; Dumas2015]. Raphaël Dumas kindly provided an updated version of the applied regression tables with some corrections in the foot parameters. The required body segment lengths were obtained from averaged kinematic motion data taken at the beginning of the trials with stopped treadmill. The applied joint axes match the axes of the estimated body segment inertial parameters and comply with the ISB recommendations [Wu2002; Wu2005].

6 Data Files

The HuMoD Database website provides a number of data files that contain the raw and processed biomechanical measurement data for the different motion tasks, the anthropometric parameters for the subjects and supplemental data. Raw and processed biomechanical measurement data as well as anthropometric parameters are stored in the MAT file format of the numerical computing software MATLAB (MathWorks, USA). Supplemental data is provided as PNG image files or WEBM video files. Links to the individual data files are organized in separate tables for each subject. The following graph gives a brief overview of the data file structure and content.

Data files

- **Subject parameters** (`Parameters.mat`)
  The `Parameters.mat` data file provides anthropometric parameters and meta data for the subject.
- **Processed data**
  - Dataset (`#.mat`)
    The `#.mat` data files, where # stands for the number of the motion task as given in Table 2, provide the processed biomechanical measurement data from the three-dimensional motion capture system, instrumented treadmill and electromyographical measurement system.
- **Raw data**
  - Motion (`#-RawMotion.mat`)
    The `#-RawMotion.mat` data files, where # stands for the number of the motion task as given in Table 2, contain the raw marker coordinates measured with the three-dimensional motion capture system.
  - Muscle (`#-RawMuscle.mat`)
    The `#-RawMuscle.mat` data files, where # stands for the number of the motion task as given in Table 2, contain the raw muscle activity data measured with the electromyographical measurement system.
  - Force (`#-RawForce.mat`)
    The `#-RawForce.mat` data files, where # stands for the number of the motion task as given in Table 2, contain the raw ground reaction forces measured with the instrumented treadmill.
- **Ground reference** (`GroundReference.mat`)
  The `GroundReference.mat` data file contains reference coordinates of the instrumented treadmill that are used to match the global reference frames of the motion capture system and the instrumented treadmill.
- **Diagrams**
  - Muscle (`#.png`)
    The supplemental `#.png` image files, where # stands for the number of the motion task as given in Table 2, visualize the processed muscle activities.
  - Force (`#.png`)
    The supplemental `#.png` image files, where # stands for the number of the motion task as given in Table 2, visualize the processed ground reaction forces.
- **Videos**
  - Motion (`#.webm`)
    The supplemental `#.webm` video files, where # stands for the number of the motion task as given in Table 2, play an animated visualization of the processed marker and joint center coordinates.

For most applications dealing with modeling and simulation of human motion dynamics, it is sufficient to employ the subject parameter file (`Parameters.mat`) and the processed data files (`#.mat`). The subject parameters can be used to create a subject-specific biomechanical kinematics and dynamics model, while the processed data files provide task-specific joint trajectories, ground reaction forces and muscle activities. For further investigations, the raw data files (`#-RawMotion.mat`, `#-RawMuscle.mat`, `#-RawForce.mat`) and the ground reference file (`GroundReference.mat`) allow to validate or modify the applied data processing and to derive additional information. A detailed description of the data file structure and content is given in Appendix E. All abbreviations used in the data files are listed in Appendices B, C and D.

7 Computational Scripts

The source code of the applied computational scripts is available in an online repository with distributed revision control. Additional helper scripts are located in the `Scripts` subdirectory. When starting with the raw biomechanical measurement data, some of the scripts require data extracted or generated by a different script. The following graph gives a brief overview of the computational scripts and provides a suggested execution sequence.

Computational scripts

- **Directory structure generation** (`DirectoryStructureGeneration.m`)
  This script generates a directory structure that is used by the computational scripts. Please modify the global path in `Scripts/getPath.m` and the local paths in `Scripts/getFile.m` if required.

- **Ground reference estimation** (`GroundReferenceEstimation.m`)
  This script estimates the rotation and translation parameters to transform points from the reference frame of the motion capture system into the the reference frame of the instrumented treadmill. The reference frame of the instrumented treadmill is the global reference frame for all datasets. This script creates the `ground` structure in the processed data files `#.mat`.

- **Motion gap filling** (`MotionGapFilling.m`)
  This tool processes the raw marker trajectories of the motion capture system and can be used to fill small gaps. It has a graphical user interface and provides different methods for gap filling. It transforms the reference frame according to ISB recommendations [Wu1995] and creates the initial `motion` structure in the processed data files `#.mat`. Figure 4 shows the graphical user interface and exemplary settings for filling a gap with the constrained fit method.

- **Motion transformation** (`MotionTransformation.m`)
  This script transforms the marker coordinates in the `motion` variable in the processed data files `#.mat` into the the reference frame of the instrumented treadmill. The reference frame of the instrumented treadmill is the global reference frame for all datasets.

- **Joint center estimation** (`JointCenterEstimation.m`)
  This script estimates the marker coordinates shifted to skin surface and the joint center positions from measured and estimated marker coordinates according to predictive methods given in different references.

- **Motion visualization** (`MotionVisualization.m`)
  This script creates an animated visualization of the processed marker and estimated joint center positions.

- **Subject parameter estimation** (`SubjectParameterEstimation.m`)
  This script estimates subject parameters based on segment lengths and on regression equations [Dumas2007a; Dumas2007b; Dumas2015] and with local reference frames according to ISB recommendations [Wu2002; Wu2005].

- **Joint trajectory estimation** (`JointTrajectoryEstimation.m`)
  This script estimates the joint trajectories including joint positions, velocities and accelerations and smoothes the estimated joint center positions by applying a Kalman smoother [DeGroote2008; Yu2004] and a subject-specific forward kinematics model.

- **Trajectory visualization** (`TrajectoryVisualization.m`)
  This script creates an animated visualization of the processed joint trajectories and smoothed joint center positions.

Force filtering (`ForceFilter.m`)
This script processes the raw ground reaction forces of the instrumented treadmill and transforms the reference frame according to ISB recommendations [Wu1995]. It synchronizes motion and force data by compensating the time delay between the motion capture system and the instrumented treadmill. This script creates the initial `force` structure in the processed data files `#.mat`.

Force separation (`ForceSeparation.m`)
This script smooths the measured ground reaction forces and separates the forces for left and right side by applying parametrized transition functions [Villeger2014].

Force matching (`ForceMatching.m`)
This script matches the ground reaction forces for left and right side by shifting residual ground reaction forces during single support.

Force visualization (`ForceVisualization.m`)
This script creates a visualization of the total and separate processed ground reaction forces.

Event detection (`EventDetection.m`)
This tool applies an event detection algorithm to find the start and end of events like steps or jumps. A graphical user interface allows to check and correct the automatically detected events. For some motion tasks, the events have to be defined manually within the graphical user interface. The tool creates the initial `events` structure in the processed data files `#.mat`. Figure 5 shows the graphical user interface with a detected event for slow straight walking.

Event visualization (`EventVisualization.m`)
This script creates a visualization of the processed ground reaction forces with an overlay of the detected events.

Center of pressure estimation (`CenterOfPressureEstimation.m`)
This script estimates the center of pressure positions from the processed ground reaction forces, measured force sensor data and given force sensor positions. The estimated center of pressure positions are limited to the foot dimensions in order to compensate high error amplification at low force sensor values.

Muscle filtering (`MuscleFilter.m`)
This script processes the raw muscle activities of the electromyographical measurement system. It rectifies the signals and applies a zero-phase low-pass, moving-average or root-mean-squares filter with adjustable parameters [Konrad2005]. The script creates the initial `muscle` structure in the processed data files `#.mat`.

Muscle normalization (`MuscleNormalization.m`)
This script normalizes the filtered muscle activities by finding the global maximum values in all datasets and correcting scattered outliers.

Muscle visualization (`MuscleVisualization.m`)
This script creates a visualization of the filtered or normalized muscle activities.

Meta data generation (`MetaDataGeneration.m`)
This scripts adds some meta data to the datasets. It creates the `meta` structure in the processed data files `#.mat`.

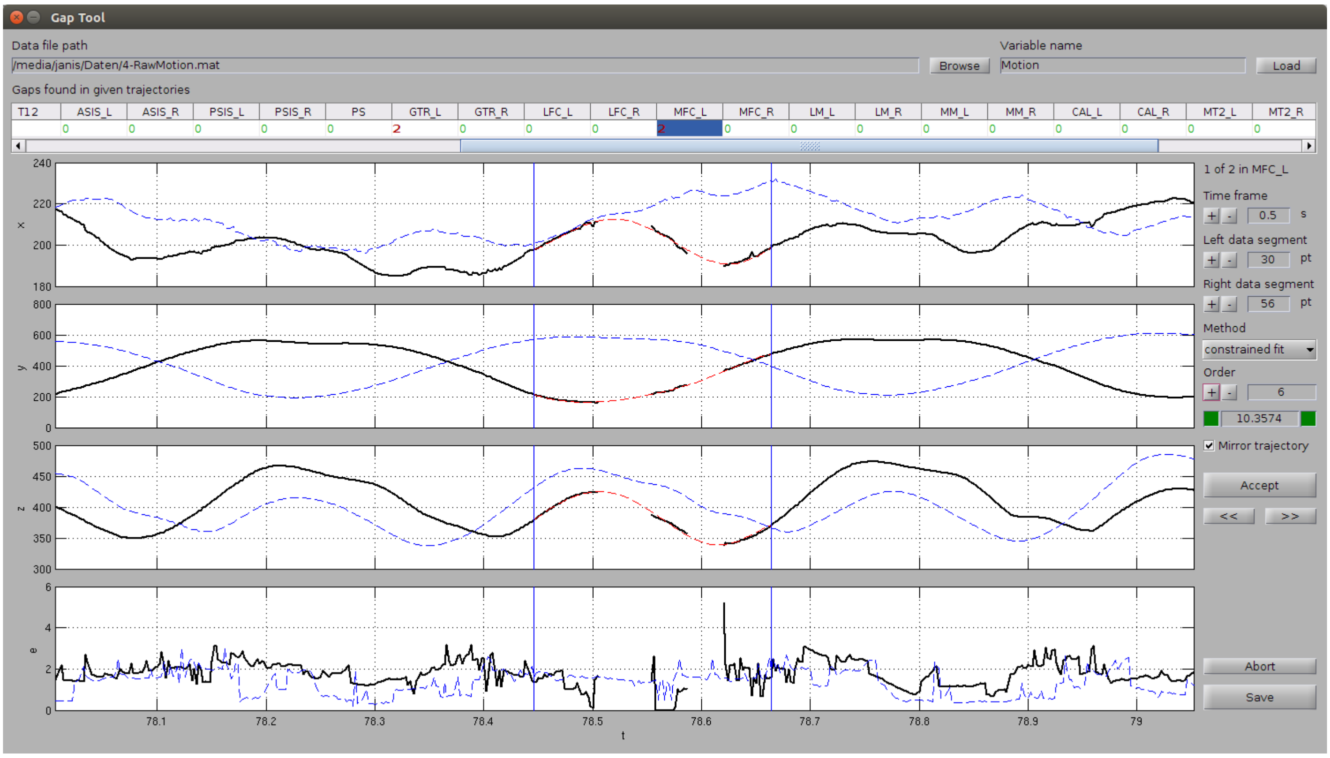


Figure 4: Graphical user interface of `MotionGapFilling.m`.

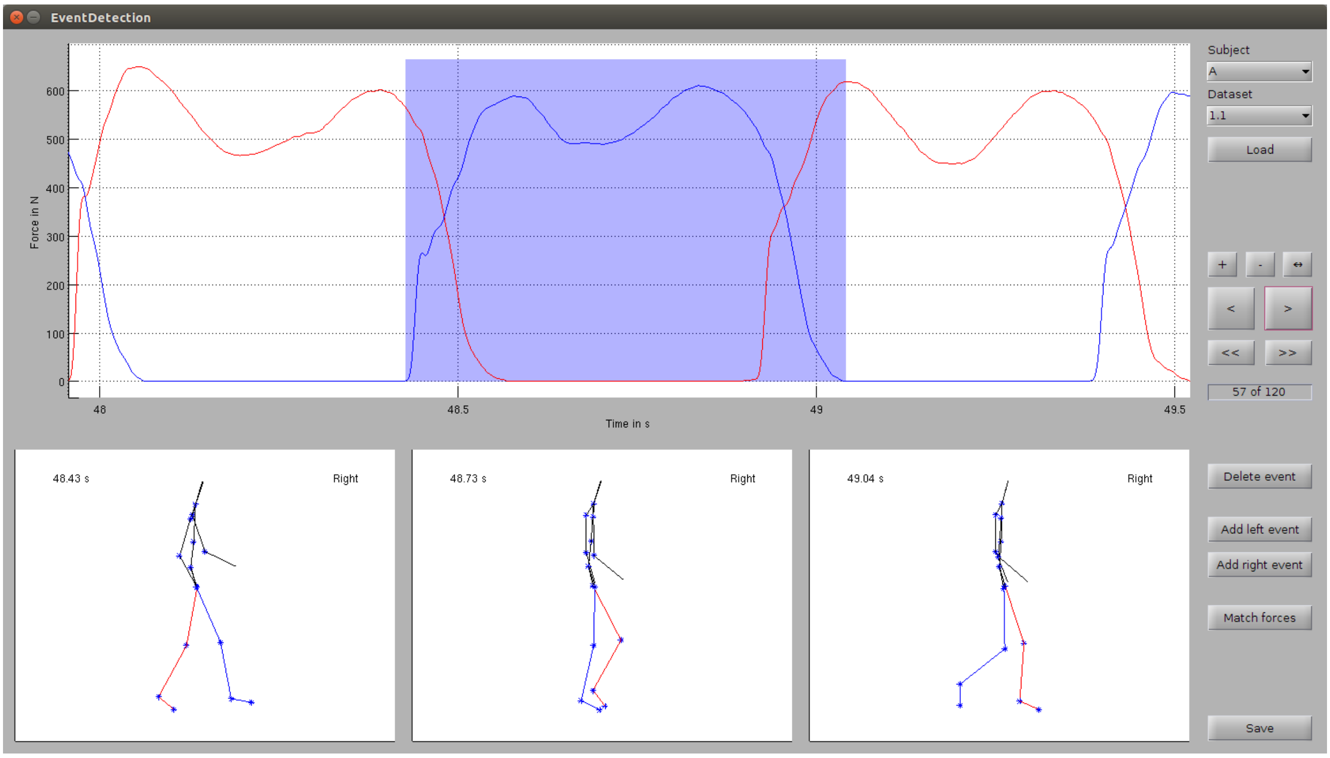
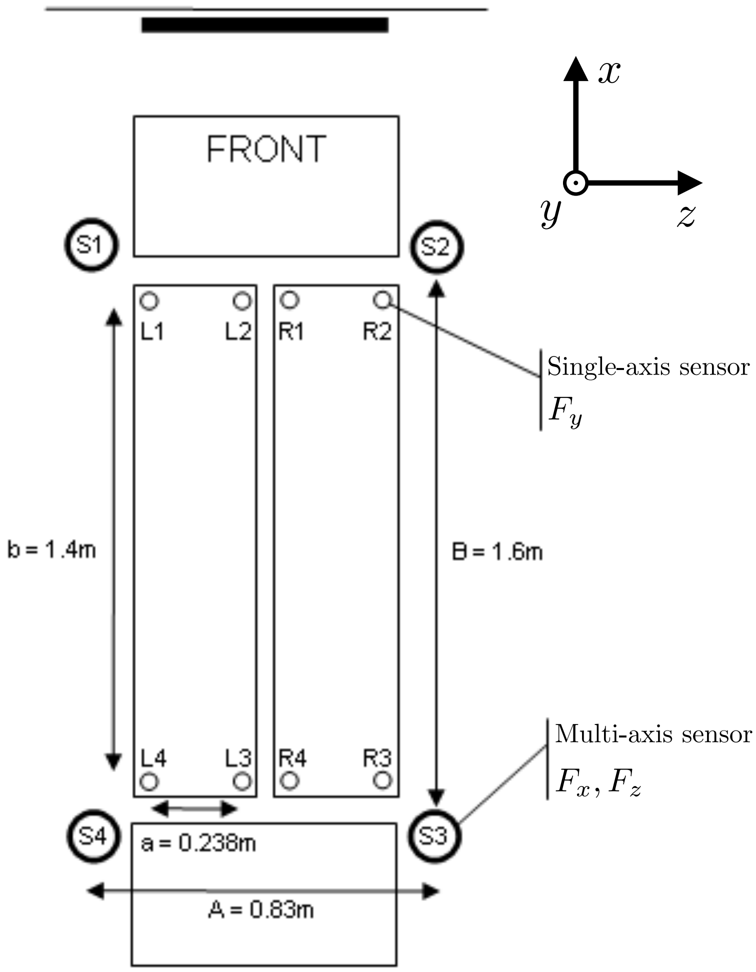


Figure 5: Graphical user interface of `EventDetection.m`.

A Treadmill Dimensions



Modified illustration from Tecmachine, France

The eight single-axis force sensors `L1` to `L4` and `R1` to `R4` were used to measure the vertical ground reaction forces F_y of the left and right foot. The four multi-axis force sensors `S1` to `S4` measured the lateral forces F_x and F_z.

B Landmark Abbreviations

GLA — **Glabella**: Undepressed skin surface point obtained by palpating the most forward projection of the forehead in the midline at the level of the brow ridges [Reed1999].

TRA_L, TRA_R — **Left and right tragion**: Undepressed skin surface point obtained by palpating the most anterior margin of the cartilaginous notch just superior to the tragus of the ear located at the upper edge of the external auditory meatus [Reed1999].

SUP — **Suprasternale**: Undepressed skin surface point at the superior margin of the jugular notch of the manubrium on the midline of the sternum [Reed1999].

C7 — **7th cervical vertebra**: Depressed skin surface point at the most posterior aspect of the spinous process of the 7th cervical vertebra [Reed1999].

T8 — **8th thoracic vertebra**: Depressed skin surface point at the most posterior aspect of the spinous process of the 8th thoracic vertebra [Reed1999].

T12 — **12th thoracic vertebra**: Depressed skin surface point at the most posterior aspect of the spinous process of the 12th thoracic vertebra [Reed1999].

ACR_L, ACR_R — **Left and right acromion**: Undepressed skin surface point obtained by palpating the most anterior portion of the lateral margin of the acromial process of the scapula [Reed1999].

LHC_L, LHC_R — **Left and right lateral humeral epicondyle**: Undepressed skin surface point at the most lateral aspect of the humeral epicondyle [Reed1999].

WRI_L, WRI_R — **Left and right wrist**: Undepressed skin surface point on the dorsal surface of the wrist midway between the radial and ulnar styloid processes [Reed1999].

$ASIS_L$, $ASIS_R$ — **Left and right anterior-superior iliac spine**: Depressed skin surface point at the anterior-superior iliac spine. Located by palpating proximally on the midline of the anterior thigh surface until the anterior prominence of the iliac spine is reached [Reed1999].

$PSIS_L$, $PSIS_R$ — **Left and right posterior-superior iliac spine**: Depressed skin surface point at the posterior-superior iliac spine. This landmark is located by palpating posteriorly along the margin of the iliac spine until the most posterior prominence is located, adjacent to the sacrum [Reed1999].

PS — **Pubic symphysis**: Depressed skin surface point at the anterior margin of pubic symphysis, located by the subject by palpating inferiorly on the midline of the abdomen until reaching the pubis. The subject is instructed to rock his or her fingers around the lower margin of the symphysis to locate the most anterior point [Reed1999].

GTR_L, GTR_R — **Left and right greater trochanter**: Undepressed skin surface point at the most lateral prominent of the upper femur.

LFC_L, LFC_R — **Left and right lateral femoral epicondyle**: Undepressed skin surface point at the most lateral aspect of the lateral femoral epicondyle [Reed1999].

MFC_L, MFC_R — **Left and right medial femoral epicondyle**: Undepressed skin surface point at the most medial aspect of the medial femoral epicondyle.

LM_L, LM_R — **Left and right lateral malleoius**: Undepressed skin surface point at the most lateral aspect of the malleolus of the fibula [Reed1999].

MM_L, MM_R — **Left and right medial malleoius**: Undepressed skin surface point at the most medial aspect of the malleolus of the tibia.

CAL_L, CAL_R — **Left and right calcaneus**: Undepressed skin surface point at the most posterior prominent of the calcaneus.

$MT2_L$, $MT2_R$ — **Left and right 2nd metatarsal head**: Undepressed skin surface point above the distal head of the 2nd metatarsal.

$MT5_L$, $MT5_R$ — **Left and right 5th metatarsal head**: Undepressed skin surface point above the distal head of the 5th metatarsal.

HAL_L, HAL_R — **Left and right hallux**: The anterior point of the 1st digit of each foot.

In the provided data files, all landmarks are identified by labels of the form `[xxx][_R/_L]`. The first part `xxx` is the two-, three- or four-letter landmark abbreviation as listed above. The last part `_L` or `_R` indicates the left or right body side if applicable.

C Muscle Abbreviations

SOL_L, SOL_R	**Left and right soleus muscle**: Plantar flexion of the ankle joint [Hermens2000].
TIA_L, TIA_R	**Left and right tibialis anterior muscle**: Dorsiflexion of the ankle joint and assistance in inversion of the foot [Hermens2000].
GLS_L, GLS_R	**Left and right gastrocnemius lateralis muscle**: Flexion of the ankle joint and assist in flexion of the knee joint [Hermens2000].
VSL_L, VSL_R	**Left and right vastus lateralis muscle**: Extension of the knee joint [Hermens2000].
RCF_L, RCF_R	**Left and right rectus femoris muscle**: Extension of the knee joint and flexion of the hip joint [Hermens2000].
BCF_L, BCF_R	**Left and right biceps femoris muscle**: Flexion and lateral rotation of the knee joint. The long head also extends and assists in lateral rotation of the hip joint [Hermens2000].
GLX_L, GLX_R	**Left and right gluteus maximus muscle**: Extends, laterally rotates and lower fibres assist in adduction of the hip joint. The upper fibres assist in adduction. Through its insertion into the iliotibial tract, helps to stabilise the knee in extension [Hermens2000].

In the provided data files, all muscles are identified by labels of the form `[xxx][_R/_L]`. The first part `xxx` is the three-letter muscle abbreviation as listed above. The last part `_L` or `_R` indicates the left or right body side if applicable.

D Joint Abbreviations

BJ	Base joint that connects the human model at the lower lumbar joint with the environment and provides three prismatic and three revolute degrees of freedom.
LNJ	Lower neck joint (C7/T1) [Reed1999].
SJ_L, SJ_R	Left and right shoulder joint.
EJ_L, EJ_R	Left and right elbow joint.
ULJ	Upper lumbar joint (T12/L1) [Reed1999; Dumas2015].
LLJ	Lower lumbar joint (L5/S1) [Reed1999].
HJ_L, HJ_R	Left and right hip joint.
KJ_L, KJ_R	Left and right knee joint.
AJ_L, AJ_R	Left and right ankle joint.
TJ_L, TJ_R	Left and right toe joint.

In the provided data files, all joints are identified by labels of the form `[p/r][xxx][y][_R/_L]`. The small letters `p` or `r` in the first part indicate, if the joint is a prismatic joint given in millimeters or a revolute joint given in radians. The second part `xxx` is the two- or three-letter joint abbreviation as listed above. The single letter `y` in the third part gives the joint axis `X`, `Y` or `Z` in the local reference frame. The last part `_L` or `_R` indicates the left or right body side if applicable.

E Data Structure

Subject parameters (`Parameters.mat`)

- `subject` [*string*] .... Subject identifier string
- `age` [*double scalar*] .... Age in years
- `gender` [*string*] .... Gender string
- `bodyHeight` [*double scalar*] .... Body height in mm
- `bodyMass` [*double scalar*] .... Body mass in kg
- `equipmentMass` [*double scalar*] .... Measurement equipment mass* in kg
- `origin` [*string*] .... Origin string
- `head` [*struct*]
  - `segmentLengthY` [*double scalar*] .... Head length along local y axis in mm
  - `mass` [*double scalar*] .... Head mass in kg
  - `comX` [*double scalar*] .... Head center of mass position in local x direction in mm
  - `comY` [*double scalar*] .... Head center of mass position in local y direction in mm
  - `comZ` [*double scalar*] .... Head center of mass position in local z direction in mm
  - `moiXX` [*double scalar*] .... Head moment of inertia about local x axis in kg m^2
  - `moiYY` [*double scalar*] .... Head moment of inertia about local y axis in kg m^2
  - `moiZZ` [*double scalar*] .... Head moment of inertia about local z axis in kg m^2
  - `poiXY` [*double scalar*] .... Head product of inertia about local x and y axes in kg m^2
  - `poiXZ` [*double scalar*] .... Head product of inertia about local x and z axes in kg m^2
  - `poiYZ` [*double scalar*] .... Head product of inertia about local y and z axes in kg m^2
  - `origin` [*struct*]
    - `point` [*string*] .... Local origin identifier string
    - `type` [*string*] .... Local origin type string
  - `relativePosition` [*struct*]
    - `xxx_x` [*double vector*] .... xxx_x marker or joint positions in local coordinates in mm
- `thorax` [*struct*]
  - `segmentLengthY` [*double scalar*] .... Thorax length along local y axis in mm
  - `mass` [*double scalar*] .... Thorax mass in kg
  - `comX` [*double scalar*] .... Thorax center of mass position in local x direction in mm
  - `comY` [*double scalar*] .... Thorax center of mass position in local y direction in mm
  - `comZ` [*double scalar*] .... Thorax center of mass position in local z direction in mm
  - `moiXX` [*double scalar*] .... Thorax moment of inertia about local x axis in kg m^2
  - `moiYY` [*double scalar*] .... Thorax moment of inertia about local y axis in kg m^2
  - `moiZZ` [*double scalar*] .... Thorax moment of inertia about local z axis in kg m^2

- poiXY [*double scalar*] .... Thorax product of inertia about local x and y axes in kg m^2
- poiXZ [*double scalar*] .... Thorax product of inertia about local x and z axes in kg m^2
- poiYZ [*double scalar*] .... Thorax product of inertia about local y and z axes in kg m^2
- origin [*struct*]
  - point [*string*] .... Local origin identifier string
  - type [*string*] .... Local origin type string
- relativePosition [*struct*]
  - xxx_x [*double vector*] .... xxx_x marker or joint positions in local coordinates in mm

abdomen [*struct*]
- segmentLengthY [*double scalar*] .... Abdomen length along local y axis in mm
- mass [*double scalar*] .... Abdomen mass in kg
- comX [*double scalar*] .... Abdomen center of mass position in local x direction in mm
- comY [*double scalar*] .... Abdomen center of mass position in local y direction in mm
- comZ [*double scalar*] .... Abdomen center of mass position in local z direction in mm
- moiXX [*double scalar*] .... Abdomen moment of inertia about local x axis in kg m^2
- moiYY [*double scalar*] .... Abdomen moment of inertia about local y axis in kg m^2
- moiZZ [*double scalar*] .... Abdomen moment of inertia about local z axis in kg m^2
- poiXY [*double scalar*] .... Abdomen product of inertia about local x and y axes in kg m^2
- poiXZ [*double scalar*] .... Abdomen product of inertia about local x and z axes in kg m^2
- poiYZ [*double scalar*] .... Abdomen product of inertia about local y and z axes in kg m^2
- origin [*struct*]
  - point [*string*] .... Local origin identifier string
  - type [*string*] .... Local origin type string
- relativePosition [*struct*]
  - xxx_x [*double vector*] .... xxx_x marker or joint positions in local coordinates in mm

upperArm_x [*struct*]
- segmentLengthY [*double scalar*] .... Torso length along local y axis in mm
- mass [*double scalar*] .... Upper arm mass in kg
- comX [*double scalar*] .... Upper arm center of mass position in local x direction in mm
- comY [*double scalar*] .... Upper arm center of mass position in local y direction in mm
- comZ [*double scalar*] .... Upper arm center of mass position in local z direction in mm
- moiXX [*double scalar*] .... Upper arm moment of inertia about local x axis in kg m^2
- moiYY [*double scalar*] .... Upper arm moment of inertia about local y axis in kg m^2
- moiZZ [*double scalar*] .... Upper arm moment of inertia about local z axis in kg m^2
- poiXY [*double scalar*] .... Upper arm product of inertia about local x and y axes in kg m^2
- poiXZ [*double scalar*] .... Upper arm product of inertia about local x and z axes in kg m^2

- poiYZ [*double scalar*] .... Upper arm product of inertia about local y and z axes in kg m^2
- origin [*struct*]
  - point [*string*] .... Local origin identifier string
  - type [*string*] .... Local origin type string
- relativePosition [*struct*]
  - xxx_x [*double vector*] .... xxx_x marker or joint positions in local coordinates in mm

lowerArm_x [*struct*]
- segmentLengthY [*double scalar*] .... Lower arm length along local y axis in mm
- mass [*double scalar*] .... Lower arm mass in kg
- comX [*double scalar*] .... Lower arm center of mass position in local x direction in mm
- comY [*double scalar*] .... Lower arm center of mass position in local y direction in mm
- comZ [*double scalar*] .... Lower arm center of mass position in local z direction in mm
- moiXX [*double scalar*] .... Lower arm moment of inertia about local x axis in kg m^2
- moiYY [*double scalar*] .... Lower arm moment of inertia about local y axis in kg m^2
- moiZZ [*double scalar*] .... Lower arm moment of inertia about local z axis in kg m^2
- poiXY [*double scalar*] .... Lower arm product of inertia about local x and y axes in kg m^2
- poiXZ [*double scalar*] .... Lower arm product of inertia about local x and z axes in kg m^2
- poiYZ [*double scalar*] .... Lower arm product of inertia about local y and z axes in kg m^2
- origin [*struct*]
  - point [*string*] .... Local origin identifier string
  - type [*string*] .... Local origin type string
- relativePosition [*struct*]
  - xxx_x [*double vector*] .... xxx_x marker or joint positions in local coordinates in mm

pelvis [*struct*]
- segmentLengthY [*double scalar*] .... Pelvis length along local y axis in mm
- segmentLengthZ [*double scalar*] .... Pelvis length along local z axis in mm
- mass [*double scalar*] .... Pelvis mass in kg
- comX [*double scalar*] .... Pelvis center of mass position in local x direction in mm
- comY [*double scalar*] .... Pelvis center of mass position in local y direction in mm
- comZ [*double scalar*] .... Pelvis center of mass position in local z direction in mm
- moiXX [*double scalar*] .... Pelvis moment of inertia about local x axis in kg m^2
- moiYY [*double scalar*] .... Pelvis moment of inertia about local y axis in kg m^2
- moiZZ [*double scalar*] .... Pelvis moment of inertia about local z axis in kg m^2
- poiXY [*double scalar*] .... Pelvis product of inertia about local x and y axes in kg m^2
- poiXZ [*double scalar*] .... Pelvis product of inertia about local x and z axes in kg m^2
- poiYZ [*double scalar*] .... Pelvis product of inertia about local y and z axes in kg m^2

- origin [*struct*]
  - point [*string*] .... Local origin identifier string
  - type [*string*] .... Local origin type string
- relativePosition [*struct*]
  - xxx_x [*double vector*] .... xxx_x marker or joint positions in local coordinates in mm

thigh_x [*struct*]
- segmentLengthY [*double scalar*] .... Thigh length along local y axis in mm
- mass [*double scalar*] .... Thigh mass in kg
- comX [*double scalar*] .... Thigh center of mass position in local x direction in mm
- comY [*double scalar*] .... Thigh center of mass position in local y direction in mm
- comZ [*double scalar*] .... Thigh center of mass position in local z direction in mm
- moiXX [*double scalar*] .... Thigh moment of inertia about local x axis in kg m^2
- moiYY [*double scalar*] .... Thigh moment of inertia about local y axis in kg m^2
- moiZZ [*double scalar*] .... Thigh moment of inertia about local z axis in kg m^2
- poiXY [*double scalar*] .... Thigh product of inertia about local x and y axes in kg m^2
- poiXZ [*double scalar*] .... Thigh product of inertia about local x and z axes in kg m^2
- poiYZ [*double scalar*] .... Thigh product of inertia about local y and z axes in kg m^2
- origin [*struct*]
  - point [*string*] .... Local origin identifier string
  - type [*string*] .... Local origin type string
- relativePosition [*struct*]
  - xxx_x [*double vector*] .... xxx_x marker or joint positions in local coordinates in mm

shank_x [*struct*]
- segmentLengthY [*double scalar*] .... Shank length along local y axis in mm
- mass [*double scalar*] .... Shank mass in kg
- comX [*double scalar*] .... Shank center of mass position in local x direction in mm
- comY [*double scalar*] .... Shank center of mass position in local y direction in mm
- comZ [*double scalar*] .... Shank center of mass position in local z direction in mm
- moiXX [*double scalar*] .... Shank moment of inertia about local x axis in kg m^2
- moiYY [*double scalar*] .... Shank moment of inertia about local y axis in kg m^2
- moiZZ [*double scalar*] .... Shank moment of inertia about local z axis in kg m^2
- poiXY [*double scalar*] .... Shank product of inertia about local x and y axes in kg m^2
- poiXZ [*double scalar*] .... Shank product of inertia about local x and z axes in kg m^2
- poiYZ [*double scalar*] .... Shank product of inertia about local y and z axes in kg m^2
- origin [*struct*]
  - point [*string*] .... Local origin identifier string

- type [*string*] .... Local origin type string
- relativePosition [*struct*]
  - xxx_x [*double vector*] .... xxx_x marker or joint positions in local coordinates in mm
- foot_x [*struct*]
  - segmentLengthX [*double scalar*] .... Foot length along local x axis in mm
  - segmentLengthY [*double scalar*] .... Foot length along local y axis in mm
  - segmentLengthZ [*double scalar*] .... Foot length along local z axis in mm
  - referenceLengthX [*double scalar*] .... Reference length for scaling along local x axis in mm
  - referenceLengthY [*double scalar*] .... Reference length for scaling along local y axis in mm
  - referenceLengthZ [*double scalar*] .... Reference length for scaling along local z axis in mm
  - mass [*double scalar*] .... Foot mass in kg
  - comX [*double scalar*] .... Foot center of mass position in local x direction in mm
  - comY [*double scalar*] .... Foot center of mass position in local y direction in mm
  - comZ [*double scalar*] .... Foot center of mass position in local z direction in mm
  - moiXX [*double scalar*] .... Foot moment of inertia about local x axis in kg m^2
  - moiYY [*double scalar*] .... Foot moment of inertia about local y axis in kg m^2
  - moiZZ [*double scalar*] .... Foot moment of inertia about local z axis in kg m^2
  - poiXY [*double scalar*] .... Foot product of inertia about local x and y axes in kg m^2
  - poiXZ [*double scalar*] .... Foot product of inertia about local x and z axes in kg m^2
  - poiYZ [*double scalar*] .... Foot product of inertia about local y and z axes in kg m^2
  - origin [*struct*]
    - point [*string*] .... Local origin identifier string
    - type [*string*] .... Local origin type string
  - relativePosition [*struct*]
    - xxx_x [*double vector*] .... xxx_x marker or joint positions in local coordinates in mm
- joints [*struct*]
  - absolutePosition [*struct*]
    - xxx_x [*double vector*] .... xxx_x joint positions in global coordinates in mm

* The measurement equipment consisted of reflective markers of the motion capture system and electrodes, wires and two switching boxes of the electromyographical measurement system. The mass of the reflective markers is neglectable small in relation to the components of the electromyographical measurement system. The switching boxes and wire connectors were placed about 50 mm inferior and 50 mm lateral of the respective $PSIS_x$ markers.

Processed data (#.mat)

- meta [*struct*]
  - subject [*string*] .... Subject identifier string
  - experiment [*string*] .... Measurement identifier string

- `date` [*string*] ..... Measurement date string
- `duration` [*double scalar*] ..... Total duration in s
- `startTime` [*double scalar*] ..... Motion task start time in s
- `endTime` [*double scalar*] ..... Motion task end time in Hz
- `author` [*string*] ..... Author string
- `license` [*string*] ..... License string
- `version` [*string*] ..... Version number string

`motion` [*struct*]

- `frameRate` [*integer scalar*] ..... Frame rate in Hz
- `frames` [*integer scalar*] ..... Total frame number f_{mo}
- `markerSize` [*double scalar*] ..... Marker diameter in mm
- `markerX` [*double matrix*] ..... $r \times f_{mo}$ matrix of reflective marker positions in global x coordinates in mm
- `markerY` [*double matrix*] ..... $r \times f_{mo}$ matrix of reflective marker positions in global y coordinates in mm
- `markerZ` [*double matrix*] ..... $r \times f_{mo}$ matrix of reflective marker positions in global z coordinates in mm
- `markerE` [*double matrix*] ..... $r \times f_{mo}$ matrix of reflective marker residuals in mm
- `markerLabels` [*cell array*] ..... $1 \times r$ array of reflective marker labels
- `jointX` [*struct*]
  - `estimated` [*double matrix*] . $j_e \times f_{mo}$ matrix of estimated joint positions in global x coordinates in mm
  - `smoothed` [*double matrix*] .. $j_s \times f_{mo}$ matrix of smoothed joint positions in global x coordinates in mm
- `jointY` [*struct*]
  - `estimated` [*double matrix*] . $j_e \times f_{mo}$ matrix of estimated joint positions in global y coordinates in mm
  - `smoothed` [*double matrix*] .. $j_s \times f_{mo}$ matrix of smoothed joint positions in global y coordinates in mm
- `jointZ` [*struct*]
  - `estimated` [*double matrix*] . $j_e \times f_{mo}$ matrix of estimated joint positions in global z coordinates in mm
  - `smoothed` [*double matrix*] .. $j_s \times f_{mo}$ matrix of smoothed joint positions in global z coordinates in mm
- `jointLabels` [*struct*]
  - `estimated` [*cell array*] ..... $1 \times j_e$ array of joint labels for estimated joint positions
  - `smoothed` [*cell array*] ..... $1 \times j_s$ array of joint labels for smoothed joint positions
- `surfaceX` [*double matrix*] ..... $s \times f_{mo}$ matrix of surface marker positions in global x coordinates in mm
- `surfaceY` [*double matrix*] ..... $s \times f_{mo}$ matrix of surface marker positions in global y coordinates in mm
- `surfaceZ` [*double matrix*] ..... $s \times f_{mo}$ matrix of surface marker positions in global z coordinates in mm
- `surfaceLabels` [*cell array*] ..... $1 \times s$ array of surface marker labels
- `trajectory` [*struct*]
  - `q` [*double matrix*] ..... $j_t \times f_{mo}$ matrix of smoothed joint positions in mm and rad
  - `dqdt` [*double matrix*] ..... $j_t \times f_{mo}$ matrix of smoothed joint velocities in mm s^{-1} and rad s^{-1}
  - `ddqddt` [*double matrix*] ..... $j_t \times f_{mo}$ matrix of smoothed joint accelerations in mm s^{-2} and rad s^{-2}

- `trajectoryLabels` [*cell array*] .... $1\times j_t$ array of joint labels
- `subjectVelocity` [*double vector*] .... Subject velocity at the midpoint of the $ASIS_x$ markers in $\frac{m}{s}$
- `clothThickness` [*double scalar*] .... Cloth thickness in mm
- `supportThickness` [*double scalar*] .... Cardboard thickness in mm

`force` [*struct*]

- `frameRate` [*integer scalar*] .... Frame rate in Hz
- `frames` [*integer scalar*] .... Total frame number f_{fo}
- `grfX` [*double vector*] .... $1\times f_{fo}$ vector of total ground reaction forces in global x direction in N
- `grfX_x` [*double vector*] .... $1\times f_{fo}$ vector of separated ground reaction forces in global x direction in N
- `grfY` [*double vector*] .... $1\times f_{fo}$ vector of total ground reaction forces in global y direction in N
- `grfY_x` [*double vector*] .... $1\times f_{fo}$ vector of separated ground reaction forces in global y direction in N
- `grfZ` [*double vector*] .... $1\times f_{fo}$ vector of total ground reaction forces in global z direction in N
- `grfZ_x` [*double vector*] .... $1\times f_{fo}$ vector of separated ground reaction forces in global z direction in N
- `copX_x` [*double vector*] .... $1\times f_{fo}$ vector of center of pressure positions in global x coordinates in mm
- `copY_x` [*double vector*] .... $1\times f_{fo}$ vector of center of pressure positions in global y coordinates in mm
- `copZ_x` [*double vector*] .... $1\times f_{fo}$ vector of center of pressure positions in global z coordinates in mm
- `forceSensorXx` [*double vector*] .... $1\times f_{fo}$ vector of force sensor data in global x direction in N
- `forceSensorY_xx` [*double vector*] .... $1\times f_{fo}$ vector of force sensor data in global y direction in N
- `forceSensorZx` [*double vector*] .... $1\times f_{fo}$ vector of force sensor data in global z direction in N
- `treadmillVelocity` [*double vector*] .... $f_{fo}\times 1$ vector of treadmill velocities in $\frac{m}{s}$

`muscle` [*struct*]

- `activities` [*struct*]
  - `filteredActivities` [*double matrix*] .... $m\times f_{mu}$ matrix of filtered muscle activities in V
  - `normalizedActivities` [*double matrix*] .... $m\times f_{mu}$ matrix of normalized muscle activities in V
- `filterAlgorithm` [*string*] .... Filter algorithm string
- `filterWindowSize` [*integer scalar*] .... Filter window size
- `frameRate` [*integer scalar*] .... Frame rate in Hz
- `frames` [*integer scalar*] .... Total frame number f_{mu}
- `maximumValues` [*double vector*] .... $1\times m$ vector of maximum activities used in normalization in V
- `minimumValues` [*double vector*] .... $1\times m$ vector of minimum activities used in normalization in V
- `muscleLabels` [*cell array*] .... $1\times m$ array of muscle labels

`events` [*struct*]

- `eventStart_x` [*double vector*] .... $1\times e$ vector of event start times in s
- `eventEnd_x` [*double vector*] .... $1\times e$ vector of event end times in s
- `contactPhase_x` [*boolean vector*] .... $1\times f_{fo}$ vector of booleans indicating ground contact

- `grfCorrection_x` [*boolean vector*] . 1×e vector of booleans indicating ground reaction force correction
- `copCorrection_x` [*boolean vector*] .... 1×e vector of booleans indicating center of pressure correction

- `ground` [*struct*]
  - `groundPosition` [*double vector*] .... Global origin in global coordinates in mm
  - `groundNormal` [*double vector*] .... Ground normal in global coordinates in mm
  - `translationMotion2Force` [*double vector*] .... 1×3 translation vector in mm
  - `rotationMotion2Force` [*double matrix*] .... 3×3 rotation matrix
  - `sensorLabels` [*cell array*] .... 1×n array of force sensor labels
  - `sensorX` [*double vector*] .... n×1 vector of force sensor positions in global x coordinates in mm
  - `sensorY` [*double vector*] .... n×1 vector of force sensor positions in global y coordinates in mm
  - `sensorZ` [*double vector*] .... n×1 vector of force sensor positions in global z coordinates in mm

Raw motion data (`#-RawMotion.mat`)

- `meta` [*struct*]
  - `subject` [*string*] .... Subject identifier string
  - `experiment` [*string*] .... Measurement identifier string
  - `date` [*string*] .... Measurement date string
  - `duration` [*double scalar*] .... Total duration in s
  - `startTime` [*double scalar*] .... Motion task start time in s
  - `endTime` [*double scalar*] .... Motion task end time in Hz
  - `author` [*string*] .... Author string
  - `license` [*string*] .... License string
  - `version` [*string*] .... Version number string

- `Motion_x` [*struct*]
  - `File` [*string*] .... Filename string
  - `Timestamp` [*string*] .... Timestamp string
  - `FrameRate` [*integer scalar*] .... Frame rate in Hz
  - `Frames` [*integer scalar*] .... Total frame number f_{rmo}
  - `StartFrame` [*integer scalar*] .... Start frame number
  - `Trajectories` [*struct*]
    - `Labeled` [*struct*]
      - `Count` [*integer scalar*] .... Number of labeled markers l
      - `Labels` [*cell array*] .... 1×l array of labeled markers
      - `Data` [*double matrix*] .... l×4×f_{rmo} matrix of labeled marker positions* and residuals in mm
    - `Unidentified` [*struct*]
      - `Count` [*integer scalar*] .... Number of unidentified trajectories u

- Data [*double matrix*] .... $u{\times}4{\times}f_{rmo}$ matrix of unidentified trajectories* and residuals in mm
- Discarded [*struct*]
  - Count [*integer scalar*] .... Number of discarded trajectories d
  - Data [*double matrix*] .... $d{\times}4{\times}f_{rmo}$ matrix of discarded trajectories* and residuals in mmS

* The local reference frame of the raw marker and trajectory coordinates differs from the global reference frame applied in the HUMOD Database.

Raw muscle data (#-RawMotion.mat)

- meta [*struct*]
  - subject [*string*] .... Subject identifier string
  - experiment [*string*] .... Measurement identifier string
  - date [*string*] .... Measurement date string
  - duration [*double scalar*] .... Total duration in s
  - startTime [*double scalar*] .... Motion task start time in s
  - endTime [*double scalar*] .... Motion task end time in Hz
  - author [*string*] .... Author string
  - license [*string*] .... License string
  - version [*string*] .... Version number string
- fs [*integer scalar*] .... Frame rate in Hz
- cnt [*integer scalar*] .... Number of used channels c
- lbl [*cell array*] .... $1{\times}c$ array of channel labels
- xxx_x [*double vector*] .... xxx_x muscle activities in V
- header [*struct*] .... Supplemental information

Raw force data (#-RawMotion.mat)

- meta [*struct*]
  - subject [*string*] .... Subject identifier string
  - experiment [*string*] .... Measurement identifier string
  - date [*string*] .... Measurement date string
  - duration [*double scalar*] .... Total duration in s
  - startTime [*double scalar*] .... Motion task start time in s
  - endTime [*double scalar*] .... Motion task end time in Hz
  - author [*string*] .... Author string
  - license [*string*] .... License string
  - version [*string*] .... Version number string
- fs [*integer scalar*] .... Frame rate in Hz

- cnt [*integer scalar*] .... Number of measured values v
- t [*double scalar*] .... Measurement end time in s
- lbl [*cell array*] .... $1 \times v$ array of value labels
- Sgnl [*double vector*] .... $w \times 1$ vector of the unused signal channel data
- vB [*double vector*] .... $w \times 1$ vector of unfiltered treadmill velocities in $\frac{m}{s}$
- vBden [*double vector*] .... $w \times 1$ vector of filtered treadmill velocities in $\frac{m}{s}$
- Fx [*double vector*] .... $w \times 1$ vector of total ground reaction forces in local x direction* in N
- Fxx [*double vector*] .... $w \times 1$ vector of separate ground reaction forces in local x direction* in N
- Fy [*double vector*] .... $w \times 1$ vector of total ground reaction forces in local y direction* in N
- Fyx [*double vector*] .... $w \times 1$ vector of separate ground reaction forces in local y direction* in N
- Fzx [*double vector*] .... $w \times 1$ vector of total ground reaction forces in local z direction* in N
- Fzxx [*double vector*] .... $w \times 1$ vector of separate ground reaction forces in local z direction* in N

* The local reference frame of the raw ground reaction forces is the original reference frame used by the manufacturer (Tecmachine, France) and differs from the global reference frame applied in the HuMoD Database.

Ground reference (GroundReference.mat)

- meta [*struct*]
  - subject [*string*] .... Subject identifier string
  - experiment [*string*] .... Measurement identifier string
  - duration [*double scalar*] .... Total duration in s
- motion [*struct*]
  - frameRate [*integer scalar*] .... Frame rate in Hz
  - frames [*integer scalar*] .... Total frame number f_{gr}
  - markerLabels [*cell array*] .... $1 \times t$ array of reflective marker labels
  - markerX [*double matrix*] ..... $t \times f_{gr}$ matrix of reflective marker positions in global x coordinates in mm
  - markerY [*double matrix*] ..... $t \times f_{gr}$ matrix of reflective marker positions in global y coordinates in mm
  - markerZ [*double matrix*] ..... $t \times f_{gr}$ matrix of reflective marker positions in global z coordinates in mm
  - markerE [*double matrix*] .... $t \times f_{gr}$ matrix of reflective marker residuals in mm

F References

[Bell1990] A L Bell, D R Pedersen, and R A Brand. “A Comparison of the Accuracy of Several Hip Center Location Prediction Methods”. In: *Journal of Biomechanics* 23.6 (1990), pp. 617–621.

[Davis1991] R B Davis, S Õunpuu, D Tyburski, and J R Gage. “A gait analysis data collection and reduction technique”. In: *Human Movement Science* 10.5 (1991), pp. 575–587.

[DeGroote2008] F De Groote, T De Laet, I Jonkers, and J De Schutter. “Kalman smoothing improves the estimation of joint kinematics and kinetics in marker-based human gait analysis.” In: *Journal of Biomechanics* 41.16 (2008), pp. 3390–3398.

[Dempster1955] W T Dempster. *Space requirements of the seated operator*. Tech. rep. 55-159. 1955, p. 254.

[Dumas2007a] R Dumas, L Chèze, and J-P Verriest. “Adjustments to McConville et al. and Young et al. body segment inertial parameters.” In: *Journal of Biomechanics* 40.3 (2007), pp. 543–53.

[Dumas2007b] R Dumas, L Chèze, and J-P Verriest. “Corrigendum to "Adjustments to McConville et al. and Young et al. body segment inertial parameters"”. In: *Journal of Biomechanics* 40.7 (2007), pp. 1651–1652.

[Dumas2015] R Dumas, T Robert, L Cheze, and J-P Verriest. “Thorax and abdomen body segment inertial parameters adjusted from McConville et al. and Young et al.” In: *International Biomechanics* 2.1 (2015), pp. 113–118.

[Harrington2007] M E Harrington, a B Zavatsky, S E M Lawson, Z Yuan, and T N Theologis. “Prediction of the hip joint centre in adults, children, and patients with cerebral palsy based on magnetic resonance imaging.” In: *Journal of Biomechanics* 40.3 (2007), pp. 595–602.

[Hermens2000] H J Hermens, B Freriks, C Disselhorst-Klug, and R Günter. “Development of recommendations for SEMG sensors and sensor placement procedures”. In: *Journal of Electromyography and Kinesiology* 10.5 (2000), pp. 361–374.

[Hicks1953] J H Hicks. “The mechanics of the foot: I. The joints”. In: 87.4 (1953), pp. 345–357.

[Konrad2005] P Konrad. *The ABC of EMG - A Practical Introduction to Kinesiological Electromyography*. 1st ed. Noraxon, 2005.

[Leardini1999] A Leardini, A Cappozzo, F Catani, S Toksvig-Larsen, A Petitto, V Sforza, G Cassanelli, and S Giannini. “Validation of a functional method for the estimation of hip joint centre location.” In: *Journal of Biomechanics* 32.1 (1999), pp. 99–103.

[Reed1999] M P Reed, M A Manary, and L W Schneider. *Methods for Measuring and Representing Automobile Occupant Posture*. Tech. rep. 724. 1999.

[Seidel1995] Geoffrey K. Seidel, David M. Marchinda, Marcel Dijkers, and Robert W. Soutas-Little. “Hip joint center location from palpable bony landmarks - A cadaver study”. In: *Journal of Biomechanics* 28.8 (1995), pp. 995–998.

[Villeger2014] D Villeger, A Costes, B Watier, and P Moretto. “An algorithm to decompose ground reaction forces and moments from a single force platform in walking gait”. In: *Medical Engineering and Physics* 36.11 (2014), pp. 1530–1535.

[Wu1995] G Wu and P R Cavanagh. “ISB Recommendations for Standardization in the Reporting of Kinematic Data”. In: *Journal of Biomechanics* 28.10 (1995), pp. 1257–1261.

[Wu2002] G Wu et al. “ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part I: ankle, hip, and spine”. In: *Journal of Biomechanics* 35.4 (2002), pp. 543–548.

[Wu2005] G Wu et al. “ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion—Part II: shoulder, elbow, wrist and hand”. In: *Journal of Biomechanics* 38.5 (2005), pp. 981–992.

[Yu2004] B M Yu, K V Shenoy, and M Sahani. *Derivation of Extended Kalman Filtering and Smoothing Equations*. Tech. rep. Stanford University, 2004, pp. 1–5.

[Zatsiorsky1998] V M Zatsiorsky. *Kinematics of Human Motion*. 1st ed. Human Kinetics, 1998, p. 419.