Documentation for CusToM:
Customizable Toolbox for Musculoskeletal simulation

Antoine Muller, Charles Pontonnier, Pierre Puchaud, Georges Dumont

Contents

1 Introduction **2**

2 Targeted audience **2**

3 Installation instructions **2**

4 Graphical User Interface (GUI) **2**
4.1 Musculoskeletal model generation . . . 2
4.2 Motion analysis . . . 6

5 Results **12**
5.1 BiomechanicalModel . . . 12
5.2 ExperimentalData . . . 14
5.3 InverseKinematicsResults . . . 14
5.4 ExternalForcesComputation . . . 15
5.5 InverseDynamicsResults . . . 15
5.6 MuscleForcesEstimationResults . . . 15
5.7 Visualization . . . 15

6 Tutorials and examples **16**
6.1 Tutorial #1: External forces prediction on a side-step motion . . . 17
6.2 Tutorial #2: Kinematics analysis of a pick-and-place motion realized in virtual environment 20
6.3 Tutorial #3: Lower limbs muscle forces estimation on a cycling motion . . . 22

7 Research studies using CusToM **25**

8 Appendix 1: Marker sets **29**

1 Introduction

Customizable Toolbox for Musculoskeletal simulation (CusToM) is a MATLAB® toolbox aimed at performing inverse dynamics based musculoskeletal analyzes. CusToM exhibits several features. It can generate a personalized musculoskeletal model, and can solve from motion capture data inverse kinematics, external forces estimation, inverse dynamics and muscle forces estimation problems.

According to user choices, the musculoskeletal model generation is achieved thanks to libraries containing pre-registered models. These models consist of body parts osteoarticular models, set of markers or set of muscles to be combined together. From an anthropometric based model, the geometric, inertial and muscular parameters are calibrated to fit the size and mass of the subject to be analyzed. Then, from motion capture data, the inverse kinematics step computes joint coordinates trajectories against time. Then, joint torques are computed thanks to an inverse dynamics step. To this end, external forces applied to the subject have to be known. They may be directly extracted from experimental data – as platform forces – or be estimated from motion data by using the equations of motion in an optimization scheme. Last, muscle forces are estimated. It consists in finding a repartition of muscle forces respecting the joint torques and representing the central nervous system strategy.

For a large set of musculoskeletal models and motion data, CusToM can easily performs all of the analyzes described above. CusToM has been created as a modular tool to let the user being as free and autonomous as possible. The osteoarticular models, set of markers and set of muscles are defined as bricks customizable and adaptable with each other. The design or the modification of a musculoskeletal model is simplified thanks to this modularity. Following the same idea, some methods are defined as adaptable bricks. Testing new cost functions in the optimization schemes, changing performance criteria or creating alternative motion analysis methods can be done in a relatively easy way.

A user interface has been developed to facilitate the data management and the model definition during a given study.

2 Targeted audience

CusToM is mainly addressed to PhD students, Master students, Post-doctoral fellows, academics or anyone interested in motion analysis applications (e.g. coachs, ergonomists,...). The toolbox does not ask a high level in coding or computer science to run classical analyses and may also be used as a teaching support.

3 Installation instructions

CusToM was implemented and tested with the MATLAB® 2018a version and requires the Symbolic Math Toolbox™, the Optimization Toolbox™ and the Parallel Computing Toolbox™ to run.

After downloading the main folder named `CusToM` and placing it in a relevant location on the computer, the installation only consists in running the `Installation` function.

4 Graphical User Interface (GUI)

To perform a new study, the user has to create a new folder containing all the experimental data to be analyzed. This folder has to be the working folder of MATLAB during the study.

4.1 Musculoskeletal model generation

The first step consists in defining the musculoskeletal model parameters. To this end, the first user interface (Figure 1-4) is opened by executing `GenerateParameters`. The `GenerateParameters` function can be found in `...\CusToM\Functions\Interface`.

It is composed of two parts. The left one allows the user to choose all the musculoskeletal model parameters on different tabs (1.1). The right one allows the visualization of the model resulting from the parameters tuning. This interface generates a parameters file named `ModelParameters` by clicking on the **Generate parameters** button (1.3). This file is created in the current working folder of MATLAB.

The first tab (Figure 1) contains the general parameters. The size and the mass of the subject are defined here. It also contains the button **Load parameters** (1.4) allowing to load a `ModelParameters` file already computed for another study.

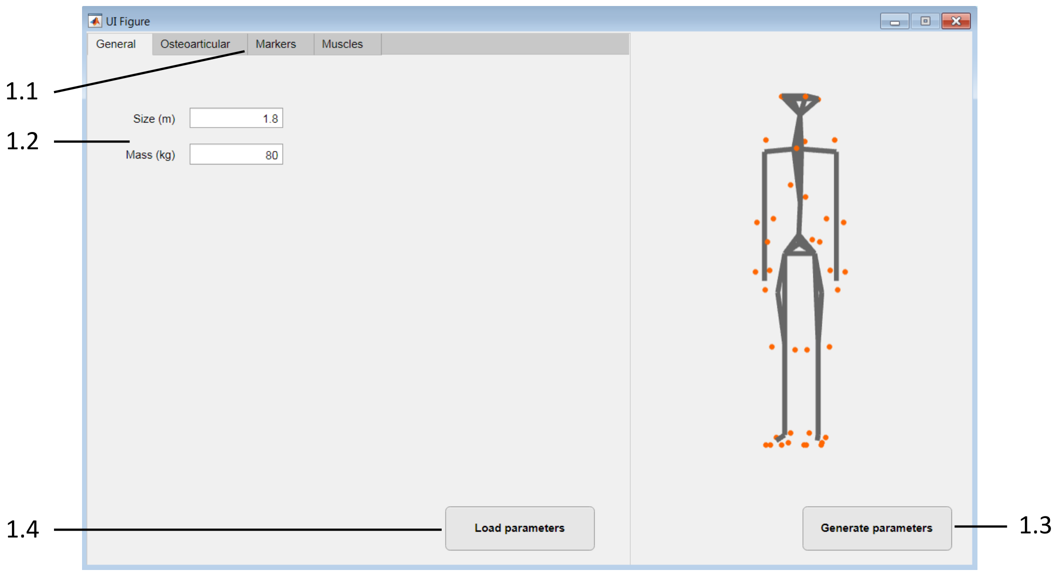
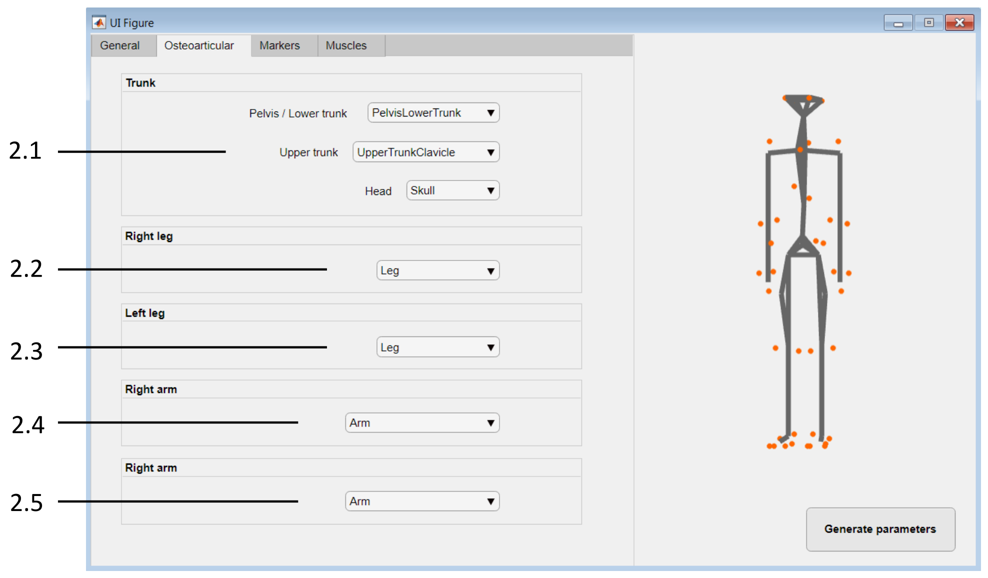


Figure 1: First tab of the graphical user interface for the definition of the musculoskeletal model parameters: general parameters

The second tab (Figure 2) allows the user to define the osteoarticular model. The whole body is divided into 5 parts: trunk (2.1), right leg (2.2), left leg (2.3), right arm (2.4) and left arm (2.5). The trunk is itself divided into the pelvis and lower trunk, the upper trunk and the head parts. For each of these parts, the body part model is chosen in a list of pre-defined models. A field **NoModel** can be used to define a partial model or a particular morphology. A user interested in defining new models can do it by defining a new model file in the corresponding folder and following the syntax of the pre-defined ones. The created model will then appear in the interface as a new choice for a given body part. For example, to create a new oestoarticular arm model, you can go to `...\CusToM\Functions\Models\Osteoarticular\Arm` and use the current models to create a new one.

The third tab (Figure 3) contains parameters about the markers set. The choice of the markers set into a pre-defined list is done by using a drop down menu (3.1). According to the chosen set, additional options may be required (3.2). This could be a variable parameters on a given markers set, for example, the number of markers used on hands. Each marker is thus represented by a button (3.3). Each of them allows to remove or to add the considered marker from the set. A user interested in defining new markers sets can do it by defining a new markers set file in the corresponding folder and following the syntax of the pre-defined ones. The created markers set will then appear in the interface as a new choice in the drop down menu. To create a new marker set, you can go to `...\CusToM\Functions\Models\Markers` and duplicate one of the current markers sets to create a new one. Each marker of a marker set is associated to an anatomical point defined on a segment of the oestoarticular model. The set is defined as a cell table and each marker is declared with the following syntax: `{'C3D_markername', 'Model_markername', {'Off';'On';'Off'}}`. The first string is the name of the marker that the toolbox will search in the C3D, the second string is the name of the anatomical point to be found in the model (please notice that these two strings may be completely different), the 3 last strings are the Boolean allowing the optimization of the marker position on the model in the 3 local directions of the corresponding segment. Therefore, if you found some existing markers well placed but badly named for your own marker set, you can just use the anatomical position and change the marker name to be found in the C3D file.
You can refer to section 8 for more details about the current marker sets implemented in the toolbox.

The fourth tab (Figure 4) allows to add muscles on the model. The toolbox contains several pre-



Trunk

Pelvis / Lower trunk: PelvisLowerTrunk

Upper trunk: UpperTrunkClavicle

Head: Skull

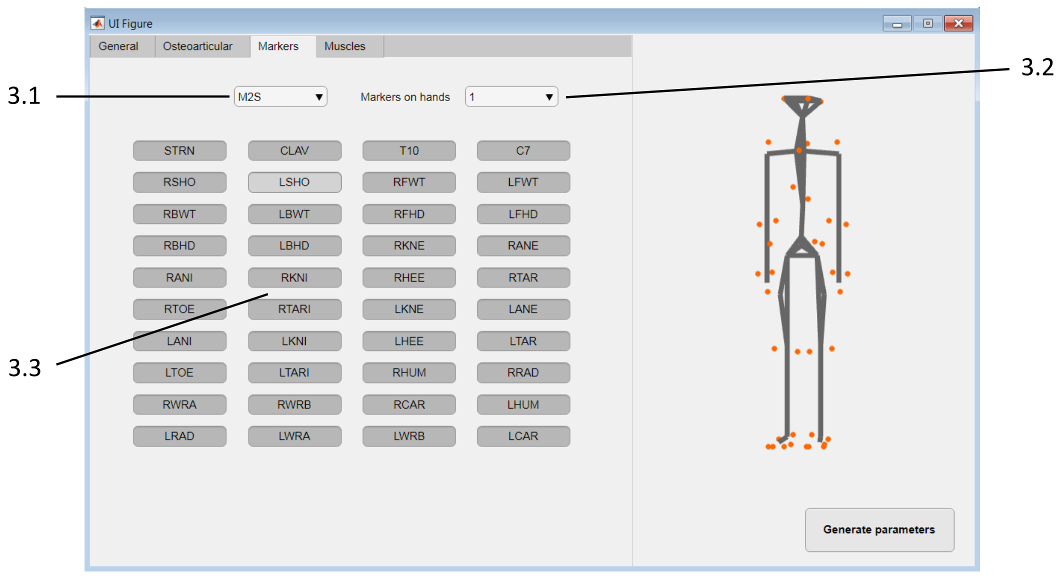
Right leg: Leg

Left leg: Leg

Right arm: Arm

Right arm: Arm

Figure 2: Second tab of the graphical user interface for the definition of the musculoskeletal model parameters: body parts models.



M2S

Markers on hands: 1

Figure 3: Third tab of the GUI for the definition of the musculoskeletal model parameters: marker set.

defined sets of muscles. The addition or the deletion of a muscle set is done by using the **Delete** (4.3) and **Add** (4.4) buttons. For each added muscles set (4.1), additional information (4.2) as the desired side of the muscles may be required. A user interested in defining new muscles sets can do it by defining a new muscles set file in the corresponding folder and following the syntax of the pre-defined ones. The created muscles set will then appear in the interface as a new choice in the drop down menu. To create a new muscles set, you can go to `...\CusToM\Functions\Models\Muscles` and use the current muscles sets to create a new one. Each muscle of a muscle set is associated to force generation parameters and an anatomical path defined with anatomical points of the osteoarticular model. The set is defined as a cell table and each muscle is a cell declared with the following syntax: `{'Muscle_Name',` f_0`,` l_0`,` K_t`,` l_s`,` α_0`,` `{'Muscle_origin_on_model', 'Muscle_via_point1', ..., 'Muscle_insertion_on_model'}}`. The first string is the name of the muscle, f_0 the maximal isometric tension developed by the muscle in N, l_0 is the rest length in m, K_t is the tendon stiffness (adimensional), l_s is the tendon slack length in m, $alpha_0$ is the pennation angle in degrees, and the final cell of strings is the muscle path (described with anatomical points defined in the osteoarticular model). Therefore, if you found some existing muscles that you would use or modify in your own set, you can just use the existing anatomical positions or add new ones to change the path.

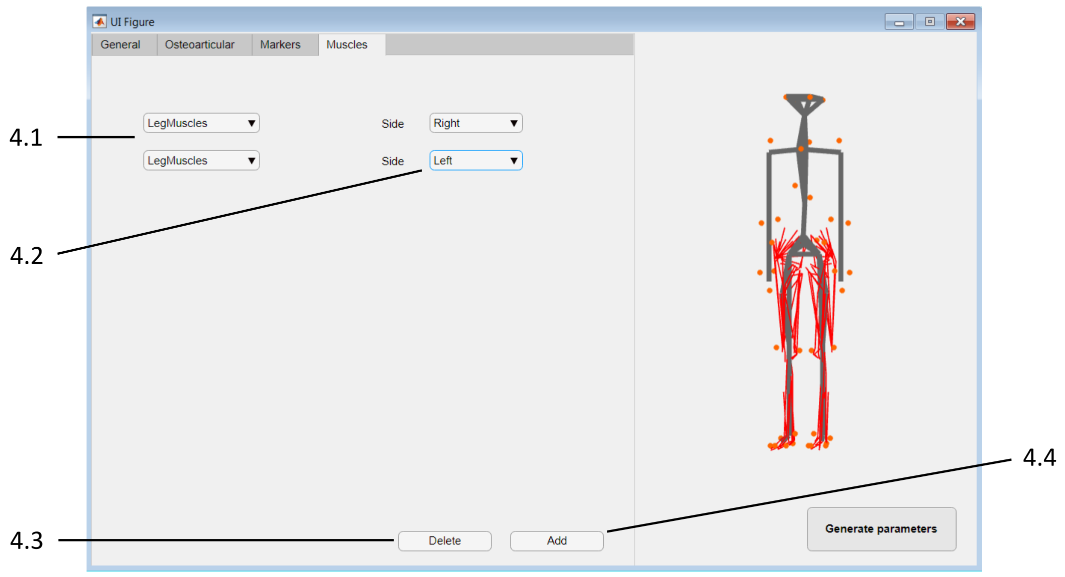


Figure 4: Fourth tab of the graphical user interface for the definition of the biomechanical model parameters: muscles set.

The file `ModelParameters` contains all the musculoskeletal model parameters and is automatically generated by the interface shown above. It is a structure which contains all these fields:

- *Size*: subject size;
- *Mass*: subject mass;
- *PelvisLowerTrunk*: function handle calling the model for the pelvis and lower trunk;
- *UpperTrunk*: function handle calling the model for the upper trunk model;
- *Head*: function handle calling the model for the head;
- *RightLeg*: function handle calling the model for the right leg;
- *LeftLeg*: function handle calling the model for the left leg;
- *RightArm*: function handle calling the model for the right arm;
- *LeftArm*: function handle calling the model for the left arm;
- *Markers*: function handle calling the markers set;
- *MarkersOptions*: additional options associated to the chosen markers set;
- *MarkersRemoved*: cell array containing the names of the markers to be ignored in the analysis compared to the ones contained in the chosen markers set;
- *Muscles*: cell array containing functions handle calling the muscle models;
- *MusclesOptions*: additional options associated to the chosen muscles set. The size and the structure of *MusclesOptions* is the same as the field *Muscles*.

4.2 Motion analysis

The second user interface (Figure 5) consists in defining all the analysis parameters. It can be opened by executing `Analysis`. This interface generates a parameters file name `AnalysisParameters` by clicking on the **Run** button (5.7). All the motion analysis computations are thus automatically launched. The button `Load parameters` (5.2) allows to load a `AnalysisParameters` file already computed for another study. The interface represents the different steps available for the motion analysis. It contains 3 calibration steps: the geometrical calibration, the inertial calibration and the muscular calibration, and four analysis steps: the inverse kinematics, the external forces estimation, the inverse dynamics and the muscle forces estimation.

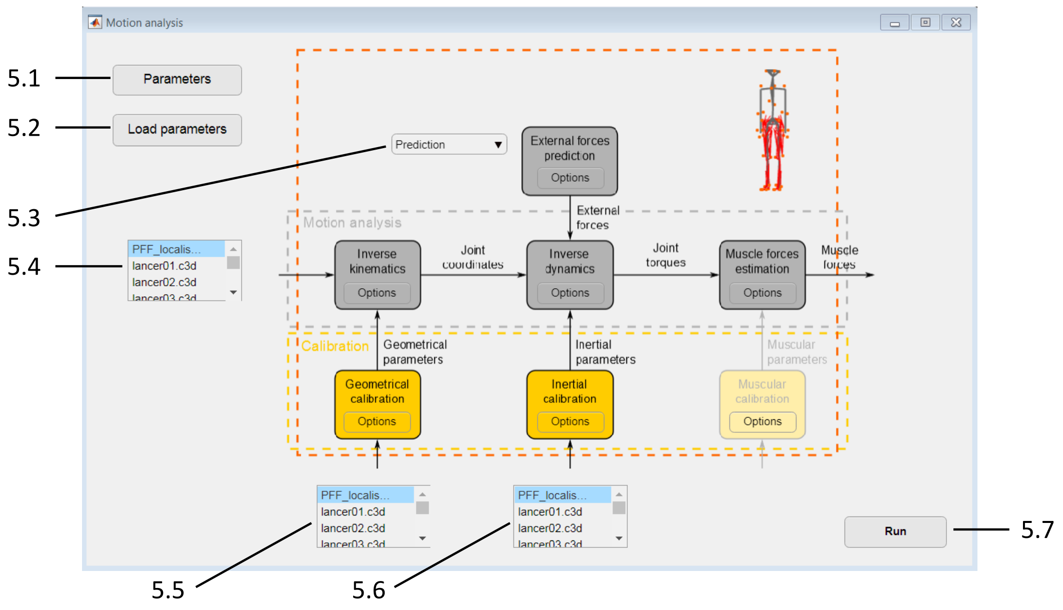


Figure 5: GUI for the motion analysis parameters.

The list box (5.4) allows to select files that have to be studied in the analysis steps. It contains the name of all *c3d* files that are in the current folder. The field *filename* on the `AnalysisParameters` file thus contains a cell array with all the selected names.

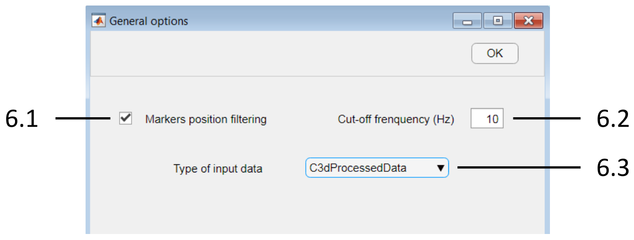
The list boxes (5.5) and (5.6) allow to select the files used respectively for the geometrical calibration and for the inertial calibration. Only one file could be selected for each list box.

General parameters of the motion analysis are available by clicking on the **Parameters** button (5.1) (Figure 6). If the check box (6.1) is active, the markers position extracted from the *c3d* file will be filtered (4-th order Butterworth low pass filter with no phase shift).The cut-off frequency of the filter can be modified (6.2). The drop down (6.3) can be used to choose a specific type of input data. Currently, only C3D files are supported. A user interested in using original input data (Kinect data, ...) can do it by defining a input handling function in the corresponding folder and following the syntax of the predefined ones. The created function and data type will then appear in the interface as a new choice in the drop down menu.

In the `AnalysisParameters` file, the general parameters are stored in the field *General* which itself contains these fields:

- *FilterActive*: logical value to select if the markers position have to be filtered or not;
- *FilterCutOff*: numerical value of the filter cut-off frequency;
- *InputData*: function handle calling the function to extract and process c3d data.

Moreover, several additional options for each step of the motion analysis are available. Each of them is accessible by clicking the **Options** button associated to the step. For each step, the different options are detailed in the following sections.



Markers position filtering ✓

Cut-off frenquency (Hz) 10

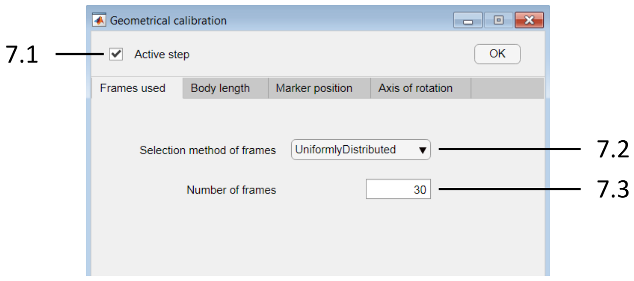
Type of input data C3dProcessedData

Figure 6: GUI for the general parameters of the motion analysis.

Geometrical calibration

In the user interface (Figure 7-10), a check box (7.1) allows to activate or not the geometrical calibration step. If this step is activated, several options are available on four tabs.

The first tab (Figure 7) contains parameters about the frames used. The function used to select the frames throughout the motion is chosen thanks to the drop down menu (7.2). The number of frames to select is defined in the box (7.3).



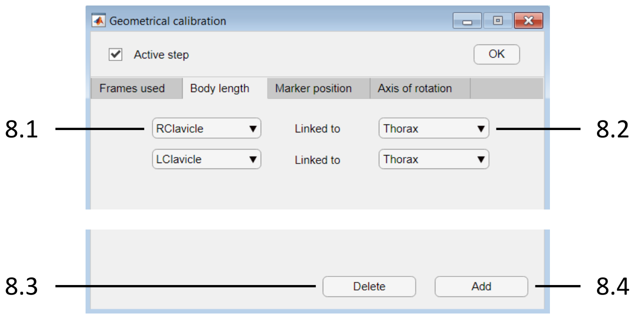
Active step ✓

Selection method of frames UniformlyDistributed

Number of frames 30

Figure 7: First tab of the GUI for the definition of the geometrical calibration parameters.

The second tab (Figure 8) contains parameters about the calibration of body part lengths. It allows to link two segments lengths, applying the same homothety coefficient for these two segments as a constraint in the optimization scheme. Thanks to the **Delete** (8.3) and **Add** (8.4) buttons, users can delete and add constraints. For each of them, the two solids are selected by using the drop downs menus (8.1) and (8.2). One solid can be selected only one time on the left drop downs menus. Otherwise, only the last constraint will be taken into account.



Active step ✓

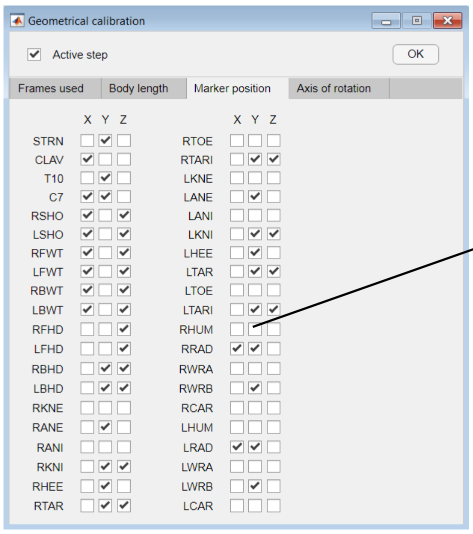
RClavicle Linked to Thorax

LClavicle Linked to Thorax

Figure 8: Second tab of the GUI for the definition of the geometrical calibration parameters.

The third tab (Figure 9) contains parameters about the calibration of the local positions of the model markers. The calibration of the local position of a marker could be blocked in one or more directions. So, for each marker, three check boxes (9.1) allow to fix or liberate the calibration of its local position along the x-axis, y-axis or z-axis (a checked box corresponds to a permitted calibration).

The fourth tab (Figure 10) contains parameters about the calibration of the axis of rotation orienta-

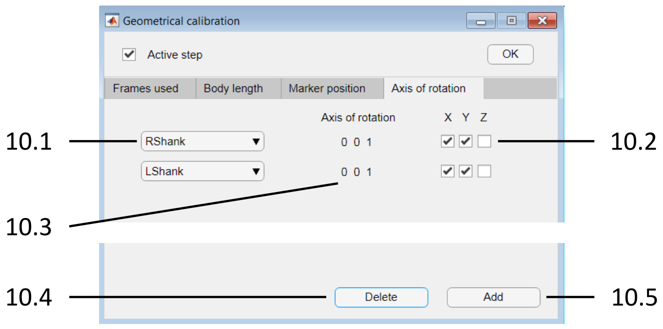


☑ Active step

	X	Y	Z		X	Y	Z
STRN	☐	☑	☐	RTOE	☐	☐	☐
CLAV	☑	☐	☐	RTARI	☐	☑	☑
T10	☐	☑	☐	LKNE	☐	☐	☐
C7	☑	☑	☐	LANE	☐	☑	☐
RSHO	☑	☐	☑	LANI	☐	☐	☐
LSHO	☑	☐	☑	LKNI	☐	☑	☑
RFWT	☑	☐	☑	LHEE	☐	☑	☐
LFWT	☑	☐	☑	LTAR	☐	☑	☑
RBWT	☑	☐	☑	LTOE	☐	☐	☐
LBWT	☑	☐	☑	LTARI	☐	☑	☑
RFHD	☐	☐	☑	RHUM	☐	☐	☐
LFHD	☐	☐	☑	RRAD	☑	☑	☐
RBHD	☐	☑	☑	RWRA	☐	☐	☐
LBHD	☐	☑	☑	RWRB	☐	☑	☐
RKNE	☐	☐	☐	RCAR	☐	☐	☐
RANE	☐	☑	☐	LHUM	☐	☐	☐
RANI	☐	☐	☐	LRAD	☑	☑	☐
RKNI	☐	☑	☑	LWRA	☐	☐	☐
RHEE	☐	☑	☐	LWRB	☐	☑	☐
RTAR	☐	☑	☑	LCAR	☐	☐	☐

Figure 9: Third tab of the GUI for the definition of the geometrical calibration parameters.

tion. Thanks to the **Delete** (10.4) and **Add** (10.5) buttons, the user can delete and add constraints. For each of them, the axis is selected thanks to its associated solid by using the drop down menu (10.1). The orientation of this axis is indicated in (10.3). The three check boxes (10.2) allow to modify the orientation of the axis along the x-axis, the y-axis or the z-axis (a checked box corresponds to a permitted calibration).



☑ Active step

	Axis of rotation	X	Y	Z
RShank ▾	0 0 1	☑	☑	☐
LShank ▾	0 0 1	☑	☑	☐

Figure 10: Fourth tab of the GUI for the definition of the geometrical calibration parameters.

In the `AnalysisParameters` file, the geometrical calibration parameters are stored in the field *CalibIK* which itself contains all of these fields:

- *Active*: logical value to activate or not the geometrical calibration step;
- *filename*: characters string containing the name of the file used;
- *Frames.Method*: function handle calling the function to select the frames throughout the motion;
- *Frames.NbFrames*: number of frames to be used in the geometrical calibration;
- *LengthAdd*: cell array containing the names of the solids involved in the additional constraints – compared to constraints initially defined in the model;
- *LengthDelete*: cell array containing the name of the solids of the deleted constraints – compared to constraints initially defined in the model;
- *MarkersCalibModif*: cell array containing the local calibrated directions of markers where modifications were applied compared to informations initially defined in the model;
- *AxisAdd*: cell array containing the orientation axes of the calibrated axes of rotation added to the ones being already defined in the model;

- *AxisDelete*: cell array containing the orientation of the calibrated axis of rotation deleted in the model.

Inertial calibration

In the user interface (Figure 11), a check box (11.1) allows to activate or not the inertial calibration step. If this step is activated, several options are available.

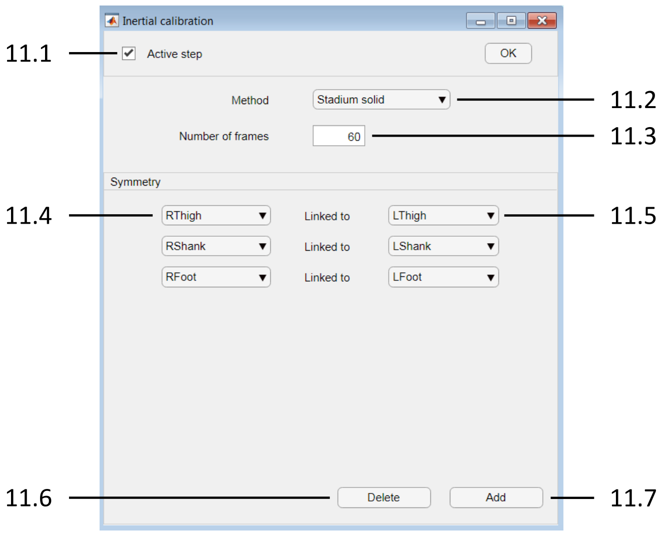


Figure 11: GUI for the definition of the inertial calibration parameters.

The inertial calibration method is selected thanks to the drop down menu (11.2). The number of frames to select is defined in the editable field (11.3). Some options could be added to link the inertial parameters of two symmetric solids through the definition of additional constraints. Thanks to the **Delete** (11.6) and **Add** (11.7) buttons, users can delete and add these constraints. For each of them, the two solids are selected by using the drop downs menus (11.4) and (11.5).

In the `AnalysisParameters` file, the inertial calibration parameters are stored in the field *CalibID* which itself contains all of these fields:

- *Active*: logical value to activate or not the inertial calibration step;
- *filename*: characters string containing the name of the file used;
- *Frames.NbFrames*: number of frames used;
- *Method*: function handle calling the inertial calibration method;
- *Symmetry*: array containing the position of solids in the osteoarticular model where a constraint of symmetry has been added.

Muscular calibration

At this moment, no validated muscular calibration is available. This step is thus deactivated. Muscular parameters of the models available in the current version are based on anthropometric data.

Inverse kinematics

In the user interface (Figure 12), a check box (12.1) allows to activate or not the inverse kinematics step. If this step is activated, several options are available.

The inverse kinematics method is selected thanks to the drop down menu (12.2). The check box (12.3) allows to choose if the joint coordinates coming from this step will be filtered (4-th order Butterworth low pass filter with no phase shift) or not. If this box is selected, the cut-off frequency can be modified

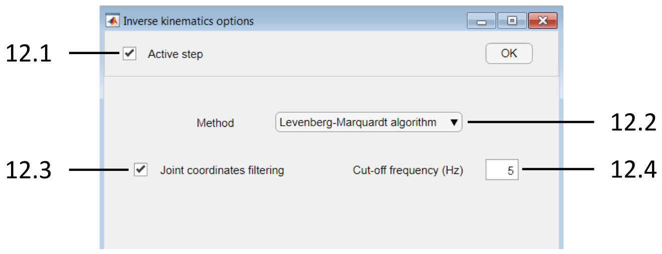

Inverse kinematics options
12.1
Active step
OK
Method
Levenberg-Marquardt algorithm
12.2
12.3
Joint coordinates filtering
Cut-off frequency (Hz)
5
12.4

Figure 12: GUI for the definition of the inverse kinematics parameters.

(12.4).

In the `AnalysisParameters` file, the inverse kinematics parameters are stored in the field *IK* which itself contains all of these fields:

- *Active*: logical value to activate or not the inverse kinematics step;
- *Method*: numerical value calling the inverse kinematics function (1 for the optimization method; 2 for the Levenberg-Marquardt algorithm);
- *FilterActive*: logical value to select if the joint coordinates will be filtered or not;
- *FilterCutOff*: numerical value of the filter cut-off frequency.

External forces computation

The external forces can be computed by three different methods (chosen thanks to (5.3)): either there is no external force, either external forces are based on experimental data or external forces are predicted from the motion. The field *ID.InputData* of the `AnalysisParameters` file contains this information (numerical value 0 for no external external force, 1 for the experimental data and 2 for the prediction).

If there is no external force, no additional option is available.

If external forces are based on experimental data, additional options are defined in the user interface (Figure 13). The check box (13.1) allows to choose if the external forces data will be filtered (4-th order Butterworth low pass filter with non phase shift) or not. If this box is selected, the cut-off frequency can be modified (13.2). The drop down menu (13.3) allows to choose which function is used to extract and process c3d data. For each platform detected on the c3d, the solid with which it is in contact is selected thanks to the drop downs menus (13.4). A field **NoContact** can be used to ignore platforms data.

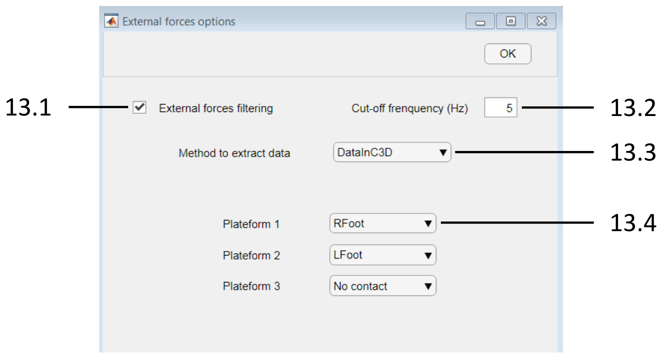


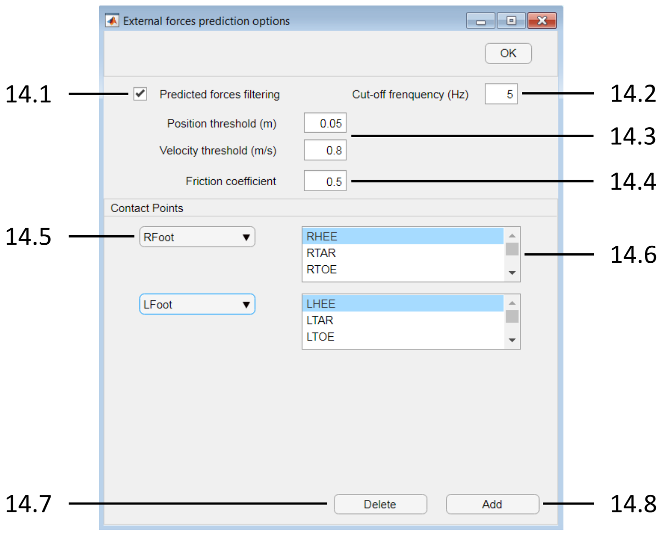
Figure 13: GUI for the definition of the external forces computation parameters when they are based on experimental data.

In the `AnalysisParameters` file, the external forces computation parameters when they are based on experimental data are stored in the field *ExternalForces* which itself contains of these fields:

- *FilterActive*: logical value to select if the joint coordinates are filtered or not;

- *FilterCutOff*: filter cut-off frequency;
- *Method*: function handle calling the function to extract and process c3d data;
- *Options*: cell array containing the names of solids which are in contact with the different platforms.

If external forces are computed thanks to a prediction method, additional options are defined on the user interface (Figure 14). The check box (14.1) allows to choose if the external forces data will be filtered (4-th order Butterworth low pass filter with non phase shift) or not. If this box is selected, the cut-off frequency can be modified (14.2). The position and velocity thresholds and the friction coefficient are respectively defined by using the editable fields (14.3) and (14.4). Each contact point is chosen by using anatomical points available in the model. The drop down menu (14.5) allows to select a solid and the list boxes (14.6) allows to select a set of points into the associated solid. The deletion or the addition of a solid is made thanks to **Delete** (14.7) and **Add** (14.8) buttons.



Predicted forces filtering ✓
Cut-off frenquency (Hz) 5
Position threshold (m) 0.05
Velocity threshold (m/s) 0.8
Friction coefficient 0.5

RFoot: RHEE, RTAR, RTOE
LFoot: LHEE, LTAR, LTOE

Figure 14: GUI for the definition of the external forces computation parameters when there are based on a prediction method.

In the `AnalysisParameters` file, the external forces computation parameters when they are based on a prediction method are stored in the field *Prediction* which itself contains all of these fields:

- *FilterActive*: logical value to select if the joint coordinates are filtered or not;
- *FilterCutOff*: filter cut-off frequency;
- *PositionThreshold*: position threshold;
- *VelocityThreshold*: velocity threshold;
- *FrictionCoef*: friction coefficient;
- *ContactPoint*: cell array containing the names of the anatomical points defined as the contact points.

Inverse Dynamics

In the user interface (Figure 15), a check box (15.1) allows to activate or not the inverse dynamics step. No additional options are available for this step.

The field *ID.Active* of the `AnalysisParameters` file contains the logical value to activate or not the inverse dynamics step.

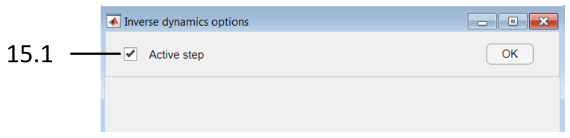


Figure 15: GUI for the definition of the inverse dynamics parameters.

Muscle forces estimation

In the user interface (Figure 16), a check box (16.1) allows to activate or not the muscle forces estimation step. If this step is activated, several options are available.

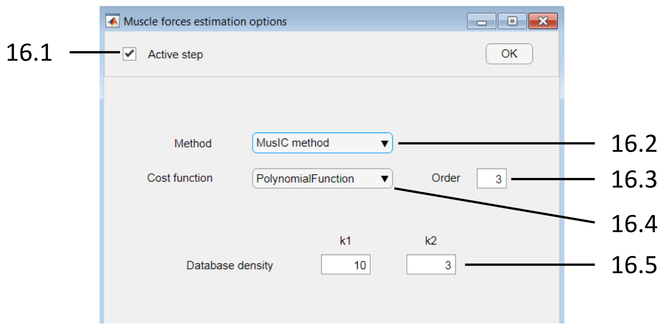


Figure 16: GUI for the definition of the muscle forces estimation parameters.

The muscle forces estimation method is selected thanks to the drop down menu (16.2). If the MusIC method is selected, the densities of the database can be modified (16.5). The cost function used to solve the force sharing problem is chosen by using the drop down menu (16.4). According to the cost function used, additional options (16.3) – as the order of the polynomial function – could be required. A user interested in using original cost functions can do it by defining a new cost function in the corresponding folder and following the syntax of the predefined ones. The created function and data type will then appear in the interface as a new choice in the drop down menu.

In the `AnalysisParameters` file, the muscle forces estimation parameters are stored in the field *Muscles* which itself contains all these fields:

- *Active*: logical value to activate or not the muscle forces estimation step;
- *Method*: muscle forces estimation method (1 for the use of an optimization method; 2 for the use of the MusIC method);
- *DatabaseDensity*: array containing the densities of the database used for the MusIC method;
- *Costfunction*: function handle calling the cost function used to solve the force sharing problem;
- *CostfunctionOptions*: numerical value containing possibly options of the cost function used.

5 Results

The results of the different steps are automatically saved. The following sections detail the structure of the results. All of the numerical data is stored in SI units.

5.1 BiomechanicalModel

`BiomechanicalModel` is directly saved in the current folder. It contains all the parameters of the musculoskeletal model, independently from the data to analyze. This structure is composed of 7 fields.

OsteoArticularModel is a structure where each data represents a solid associated to a joint and contains all of these fields:

- *name*: characters string containing the solid name;
- *sister / child / mother*: numerical values representing the model structure tree – each value corresponds to the number of the solid in the structure.
- $\boldsymbol{a}$: (3x1) array containing the joint axis – defined in the mothers reference system;
- *joint*: numerical value representing the type of joint (1 for a revolute joint; 2 for a prismatic joint);
- *limit_inf / limit_sup*: numerical values containing the joint limits;
- $\boldsymbol{u}$ / *theta*: (3x1) array and numerical value representing a fixed rotation after the joint of *theta* angle according to u axis;
- $\boldsymbol{b}$: (3x1) array containing the joint position – defined in the mothers reference system;
- $\boldsymbol{c}$: (3x1) array containing the position of the center of mass – defined in the current solid reference system;
- m: numerical value containing the solid mass;
- $\boldsymbol{I}$: (3x3) array containing the solid inertia matrix – defined in the current solid reference system;
- *anat_position*: cell array containing the set of anatomical positions, each defined by its name (characters string) and its local position on regards to the center of mass ((3x1) array);
- *Visual*: logical value to identify the real solids;
- *calib_k_constraint*: numerical value representing the number of the solid which had the same homothety coefficient for the geometrical calibration, empty array otherwise;
- *L*: cell array containing the names of two anatomical positions which represent the centers of the stadium solid used for the inertial calibration;
- *KinematicsCut*: numerical value containing the number of the geometrical cut achieved on this solid, empty array otherwise;
- *ClosedLoop*: cell array containg the name of an anatomical position (characters string) and its position for the closed loop – defined in the current solid reference system –, empty array otherwise;
- *linear_constraint*: (2x1) array containing the number of a solid and the linear coefficient which link the joint coordinates of the two solids, empty array otherwise;
- *limit_alpha*: array containing the limits of axis orientation variation during the geometrical calibration, empty array if it is not possible to calibrate it;
- v: array representing axis which have to be calibrated during the geometrical calibration.

Markers is a structure where each data represents a marker and contains all of these fields:

- *name*: characters string containing the marker name;
- *anat_position*: characters string containing an anatomical landmark name where the marker is located;
- *calib_dir*: cell array indicating the local direction in which the marker position has to be calibrated during the geometrical calibration – 'On' for a calibrated direction, 'Off' otherwise.
- *exist*: logical value indicating if the marker is defined on the osteoarticular model;
- *num_solid*: numerical value indicating the number of the solid containing the marker;
- *num_markers*: numerical value indicating the number of the anatomical position on its associated solid where the marker is located.

Muscles is a structure where each data represents a muscle and contains all of these fields:

- *name*: characters string containing the muscle name;

- *f0 / l0 / Kt / ls / alpha0*: numerical values containing the muscular properties (maximum isometric force / optimal fiber length / tendon stress-strain constant / tendon slack length / pennation angle at muscle optimal fiber length);
- *path*: cell array containing the anatomical names (characters string) of the muscle path;
- *exist*: logical value indicating if the muscle is defined on the osteoarticular model;
- *num_solid*: array indicating the numbers of the solids containing the muscle;
- *num_markers*: array indicating the numbers of the anatomical positions on their associated solids where the muscle is located.

MomentArms is a cell array containing the moment arms of each muscle for each joint. If the value is not zero, the moment arm is expressed as an anonymous function according to the joint coordinates.

MuscularCoupling is an array containing the muscular coupling matrix.

MusICDatabase is a structure containing the database used for the MusIC method.

GeometricalCalibration is a structure containing the different results of the geometrical calibration. It contains all of these fields:

- *frame_calib*: array containing the number of frames used for the calibration;
- *crit*: array containing the stop criteria value for each iteration;
- *errorm*: cell array containing, for each iteration, the differences between the experimental markers positions and the reconstructed model markers positions;
- *k_calib*: array containing the homothety coefficients;
- *p_calib*: array containing the local variation of each model marker;
- *alpha_calib*: array containing the axis orientation variation.

5.2 ExperimentalData

One `ExperimentalData` is saved per studied motion in the associated folder. This file is generated during the inverse kinematics step. It contains all variables of experimental motion data and is organized into these fields:

- *FirstFrame*: numerical value of the number of the first frame;
- *LastFrame*: numerical value of the number of the last frame;
- *MarkerPositions*: structure containing, for each marker (identified thanks to their names), their experimental positions;
- *Time*: array containing the time vector.

5.3 InverseKinematicsResults

One `InverseKinematicsResults` is saved per studied motion on the associated folder. This file is generated during the inverse kinematics step and contains all the results of this step. It is organized into these fields:

- *JointCoordinates*: array containing the joint coordinates at each time;
- *FreeJointCoordinates*: array containing the joint coordinates of the 6-dof joint at each time;
- *ReconstructionError*: array containing the differences between the experimental markers positions and the reconstructed model markers positions at each time.

5.4 ExternalForcesComputation

One `ExternalForcesComputation` is saved per studied motion in the associated folder. This file is generated during the external forces computation step and contains all the results of this step. It could contain three different fields according to the chosen step: *NoExternalForce*, *ExternalForcesExperiments* or *ExternalForcesPrediction*. Each of these fields contains the same structure where each data represents external forces at one time. Considering one of them, for each solid, a (3x2) array (field *fext*) represents the forces and torques applied on this solid – expressed on the global frame. The associated field *Visual* contains the same information adapted for the visualization of the results.

5.5 InverseDynamicsResults

One `InverseDynamicsResults` is saved per studied motion on the associated folder. This file is generated during the inverse dynamics step and contains all the results of this step. It is organized into these fields:

- *JointTorques*: array containing the joint torques at each time;
- *DynamicResiduals*: array containing the dynamic residuals at each time, that is the joint torques of the 6-dof joint.

5.6 MuscleForcesEstimationResults

One `MuscleForcesEstimationResults` is saved per studied motion on the associated folder. This file is generated during the muscle forces estimation step and contains all the results of this step. It is organized into these fields:

- *MuscleForces*: array containing the muscle forces at each time;
- *MuscleActivations*: array containing the muscle activations at each time.

5.7 Visualization

By executing `GenerateAnimate`, the user interface (Figure 17) allows to visualize different results.

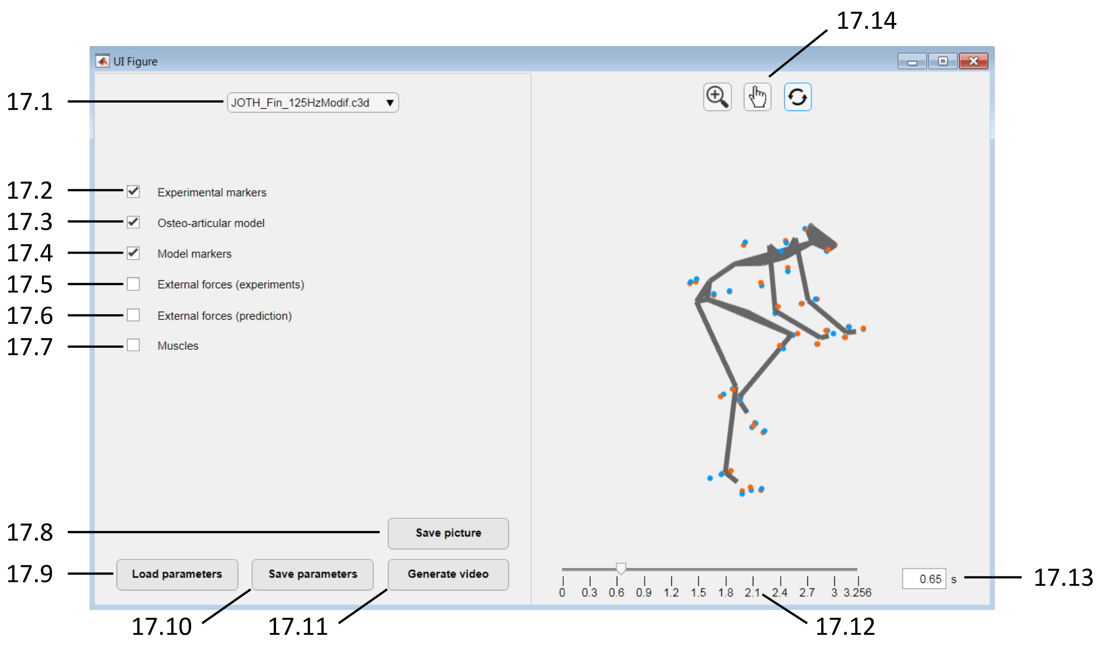


Figure 17: GUI for the animation.

The motion is selected by using the drop down menu (17.1). All elements to visualize could be selected thanks to the check box (17.2-7). Navigation buttons (17.14) allow to adjust the view. The time is set thanks to the slider (17.12) or thanks to the editable field (17.13). A copy of the picture on a *png* format could be done thanks to the button (17.8). A video of all the motion on a *avi* format could be done thanks to the button (17.11). The button (17.10) allows to save the different parameters of the animation (view, elements to visualize) into the `AnimateParameters` file. The button (17.9) allows to load the parameters of a previous study.

6 Tutorials and examples

3 tutorials extracted from research works are available in the current release. The first one consists in predicting the ground reaction forces on a sidestep motion. The second tutorial consists in analyzing kinematics of a pick-and-place task realized in a Virtual Reality environment (holding a Head-Mounted-Display). The third tutorial consists in estimating the lower limbs muscle forces during a cycling motion. The tutorials are also illustrated by videos available in the repository. You can either follow the instructions below or the videos to run these examples.

6.1 Tutorial #1: External forces prediction on a side-step motion

The third example is a side-step motion. It is extracted from a database currently being developed for population characterization. The objective of the tutorial is to compare measured and predicted external forces.

Preparation

1. First of all, once MATLAB is launched, you need to run the `Installation` function. It permits to add the `CusToM\Functions` folder and its subfolders to the MATLAB path. This folder contains all the functions, models and dependencies used in the toolbox.

2. Second, make the `CusToM\Examples\Side_Step` being the current working folder of MATLAB. The folder contains a C3D file and a post-processing file used at the end of this tutorial.

Generation of the Model Parameters

The following enables the creation of the musculoskeletal model parameters. For more information about the GUI, please refer to section 4.1.

1. To generate the parameters of the model, execute `GenerateParameters` in the MATLAB command window.

2. Once the `GenerateParameters` GUI opened, first complete the size and weight of the subject. In the current example, the subject is $1.75m$ tall ad weighs $70kg$.

3. Select the `Osteoarticular` tab. The model used here is a whole body model. It is necessary to select the different body parts to assemble the model.

   - In the `Trunk` section, select `PelvisLowerTrunk`, `UpperTrunkClavicle`, and `Skull` in the corresponding drop down menus. This is the classical analysis model used in CusToM, exhibiting 6 dofs between the pelvis (mobile base of the model) and the environment, 6 dofs in the spine (lower and upper trunk nodes), 3 dofs at the neck and 6 additional dofs allowing the upper limbs to raise during the motion (clavicle joints).
   - In the `Right Leg` and `Left Leg` sections, just select `Leg`. This is a classical leg model, with 3 dofs at the hip, 1 dof at the knee and 2 dofs at the ankle.
   - In the `Right Arm` and `Left Arm` sections, just select `Arm`. This is a classical arm model, with 3 dofs at the shoulder, 2 dofs at the elbow and 2 dofs at the wrist.

4. Select the `Markers tab`. In this example, a whole body set of markers has been used. You can select it with the drop down menu as `Marker_set2`. In the `Markers on hands` drop down menu, select 1 marker for the hands. This set of marker is less complete than the `Marker_set1` but is a classical ISB-friendly set.

5. Since the aim of the tutorial is to run a external forces prediction method, there is no need to add muscles to the model. However, feel free to add some if you want to test a muscle forces estimation method on this example.

6. All the parameters of the model are now properly set. You can generate the ModelParameters file by clicking `GenerateParameters`. The `Analysis` GUI is automatically opened.

Generation of the Analysis Parameters #1: External forces from experiment

The `Analysis` GUI is automatically opened once the `GenerateParameters` button is clicked. However, to run it independently from the `GenerateParameters` GUI, you can launch it with the command `Analysis`. For more information about the GUI, please refer to section 4.2. Please note that the Analysis GUI automatically load the ModelParameters to update the visualization of the model on the top right of the window. The Analysis GUI automatically lists the C3D files found in the current working folder of MATLAB.

In this first part, the inverse dynamics analysis will be run with the measured external forces to have a comparison point with the predicted data.

1. In the C3D list on the left (files to process), select the `ChgtDirection04.c3d` file. Please note that you may here select as many files as you need to compute the same analysis on all of these files.

2. Click the `Inverse Kinematics Options`.

   - Check the step as active.
   - In the method drop down menu, select the `Levenberg-Marquardt` method that is the fastest available.
   - Check the joint coordinates filtering as active with a Cut-off frequency of $5Hz$.
   - Click `ok` to save the parameters and close the `Inverse Kinematics Options` tab.

3. Click the `Geometrical Calibration Options`.

   - Check the step as active.
   - In the `Frames` tab, select the method `UniformlyDistributed` and choose a number of frames of 30. This is the recommendations issued from the papers [11, 7].
   - In the `Body length` tab, make sure that both `RClavicle` and `LClavicle` lengths are linked to the `Thorax` length. If this is not the case, click the Add button to generate these constraints.
   - In the `Marker Position` tab, the markers default local directions to optimize are automatically loaded from the `Marker_set2`. You can modify it the way you want. In the current tutorial, we keep the default values.
   - In the `Axis of rotation` tab, you can also add some joint axes directions to optimize. In the current tutorial, none of the axes of rotations are optimized.
   - Click `ok` to save the parameters and close the `Geometrical Calibration Options` tab.
   - In the list below the `Geometrical Calibration` box, select the `ChgtDirection04.c3d` file to be used for the calibration.

4. Click the `Inverse Dynamics Options`. Make sure that the step is active, and close the tab.

5. In the `External Forces` tab, select `from experiments`. In this first part, we will run an analysis with the measured data to be able to compare after with the predicted one.

6. Click the `External forces Options`.

   - Check that the `External forces filtering` is active, with a cut-off frequency of $5Hz$.
   - In the drop down menu `Method to extract data`, select `DataInC3D`. This is the classical way to proceed: the C3D contains the external forces measures.
   - 3 drop down menus appears below the Method. For `Plateform 1`, select `LFoot` (the subject placed his left foot on this platform during the trial). For `Plateform 2`, select `RFoot` (the subject placed his right foot on this platform during the trial). Platform 3 was inactive during this trial.
   - Click `ok` to save the parameters and close the `External forces Options` tab.

7. Since all the analysis parameters have been set, you can run the analysis by clicking the `Run` button. The application automatically saves the `AnalysisParameters`, and launch the `Main` script containing the calls to the analysis functions and the model generation functions. The application first checks if the model exists. If this is not the case, the model is created. It may take some time (several preliminary computations are launched at this time), but it is executed once per subject. If the model already exists, the analysis is directly executed on the specified data with this model.

Generation of the Analysis Parameters #2: Predicted external forces

The file will now be analyzed with a force prediction method for the external forces instead of measuring them directly. A paper explaining the prediction method used here is currently being under review.

1. Run the `Analysis` GUI from the command window.

2. Load the parameters you saved during the previous study by clicking `Load Parameters` and selecting the `AnalysisParamaters.mat`.

3. In the `External Forces` tab, select `Prediction`.

4. Click the `External forces prediction Options`.

   - Check that the `Predicted forces filtering` is active, with a cut-off frequency of $5Hz$.
   - Check that the position and velocity thresholds are at $0.05m$, $0.8m/s$ respectively, and that the friction coefficient is set at 0.5.
   - In the contact points menu, click `Add` two times to add contact points on both feet.
   - In the first drop down menu, select `RFoot`. Then select in all the anatomical points defined on the right foot, the points called `RFootPrediction1,...,RFootPrediction14`. These points are uniformly distributed below the foot to create elementary forces respecting the dynamics equations during the motion. To select several points, you can use Crtl+click or Shift+click shortcuts.
   - In the second drop down menu, select `LFoot`. Then select in all the anatomical points defined on the left foot, the points called `LFootPrediction1, ..., LFootPrediction14`.
   - Click `ok` to save the parameters and close the `External forces Options` tab.

5. Since all the analysis parameters have been set, you can run the analysis by clicking the `Run` button. Here, the model already exists, therefore the analysis is directly executed on the specified data with this model.

Post-processing

Once the analysis terminated, data can be post-processed the way you want.

1. To generate an animation of the motion, you can run the `GenerateAnimate` GUI from the MATLAB command window. Check all the boxes excepted the `Muscles` since no muscle forces were computed in this study. Adapt the point of view the way you want, and finally click `Generate Video` to proceed. The film edition takes a while, since it aggregates frame per frame MATLAB figures. Additional visualization features will be added in the next developments of the toolbox.

2. An example of post-processing can be found in the folder, called `PostProcessingSideStep.m`. You can open it and then run it to compare measured and predicted external forces.

6.2 Tutorial #2: Kinematics analysis of a pick-and-place motion realized in virtual environment

The second example is a pick-and-place motion realized in a virtual environment with an haptic device. The task is similar to the ones analyzed in [6]. Here since no direct measurement of the external forces was made, the study focused only on kinematics features.

Preparation

1. First of all, once MATLAB is launched, you need to run the `Installation` function. It permits to add the `CusToM\Functions` folder and its subfolders to the MATLAB path. This folder contains all the functions, models and dependencies used in the toolbox.
2. Second, make the `CusToM\Examples\VR_pick_and_place` being the current working folder of MATLAB. The folder contains a C3D file and a post-processing file used at the end of this tutorial.

Generation of the Model Parameters

The following enables the creation of the kinematical model parameters. For more information about the GUI, please refer to section 4.1.

1. To generate the parameters of the model, execute `GenerateParameters` in the MATLAB command window.
2. Once the `GenerateParameters` GUI opened, first complete the size and weight of the subject. In the current example, the subject is $1.78m$ tall ad weighs $60kg$.
3. Select the `Osteoarticular` tab. The model used here is an {upper limb, trunk, head} model. It is necessary to select the different body parts to assemble the model.
   - In the `Trunk` section, select `PelvisLowerTrunk`, `UpperTrunkClavicle`, and `Skull` in the corresponding drop down menus. This is the classical analysis model used in CusToM, exhibiting 6 dofs between the pelvis (mobile base of the model) and the environment, 6 dofs in the spine (lower and upper trunk nodes), 3 dofs at the neck and 6 additional dofs allowing the upper limbs to raise during the motion (clavicle joints).
   - In the `Right Leg` and `Left Leg` sections, just select `NoModel`. The experimentation did not focus on leg motions, therefore the legs were not equipped with markers.
   - In the `Right Arm` section just select `Arm`. This is a classical arm model, with 3 dofs at the shoulder, 2 dofs at the elbow and 2 dofs at the wrist.
   - In the `Left Arm` section just select `NoModel`. The left arm was supposed to be static during the task and were not equipped with markers.
4. Select the `Markers tab`. In this example, a specific set of markers has been used. You can select it with the drop down menu as `Marker_set3`. In the `Markers on hands` drop down menu, select 1 marker for the hands. This set of marker is specific, especially with the head motion that was tracked directly with the HMD motion through markers.
5. Since the aim of the tutorial is to run a kinematic analysis, there is no need to add muscles to the model. However, feel free to add some if you want to test a muscle forces estimation method on this example.
6. All the parameters of the model are now properly set. You can generate the ModelParameters file by clicking `GenerateParameters`. The `Analysis` GUI is automatically opened.

Generation of the Analysis Parameters

The `Analysis` GUI is automatically opened once the `GenerateParameters` button is clicked. However, to run it independently from the `GenerateParameters` GUI, you can launch it with the command `Analysis`. For more information about the GUI, please refer to section 4.2. Please note that the Analysis GUI automatically load the ModelParameters to update the visualization of the model on the top right of the window. The Analysis GUI automatically lists the C3D files found in the current working folder of MATLAB.

1. In the C3D list on the left (files to process), select the `Record_12.c3d` file. Please note that you may here select as many files as you need to compute the same analysis on all of these files.

2. Click the `Inverse Kinematics Options`.

   - Check the step as active.
   - In the method drop down menu, select the `Levenberg-Marquardt` method that is the fastest available.
   - Check the joint coordinates filtering as active with a Cut-off frequency of $5Hz$.
   - Click `ok` to save the parameters and close the `Inverse Kinematics Options` tab.

3. Click the `Geometrical Calibration Options`.

   - Check the step as active.
   - In the `Frames` tab, select the method `UniformlyDistributed` and choose a number of frames of 30. This is the recommendations issued from the papers [11, 7].
   - In the `Body length` tab, make sure that both `RClavicle` and `LClavicle` lengths are linked to the `Thorax` length. If this is not the case, click the Add button to generate these constraints.
   - In the `Marker Position` tab, the markers default local directions to optimize are automatically loaded from the `Marker_set3`. You can modify it the way you want. In the current tutorial, we keep the default values.
   - In the `Axis of rotation` tab, you can also add some joint axes directions to optimize. In the current tutorial, none of the axes of rotations are optimized.
   - Click `ok` to save the parameters and close the `Geometrical Calibration Options` tab.
   - In the list below the `Geometrical Calibration` box, select the `Record_12.c3d` file to be used for the calibration.

4. Click the `Inverse Dynamics Options`. Make sure that the step is inactive, and close the tab.

5. Since all the analysis parameters have been set, you can run the analysis by clicking the `Run` button. The application automatically saves the `AnalysisParameters`, and launch the `Main` script containing the calls to the analysis functions and the model generation functions. The application first checks if the model exists. If this is not the case, the model is created. It may take some time (several preliminary computations are launched at this time), but it is executed once per subject. If the model already exists, the analysis is directly executed on the specified data with this model.

Post-processing

Once the analysis terminated, data can be post-processed the way you want.

1. To generate an animation of the motion, you can run the `GenerateAnimate` GUI from the MATLAB command window. Check the boxes `Experimental Markers`, `Osteoarticular Model` and `Model Markers` only, since no data is available for the other visualization features. Adapt the point of view the way you want, and finally click `Generate Video` to proceed. The film edition takes a while, since it aggregates frame per frame MATLAB figures. Additional visualization features will be added in the next developments of the toolbox.

2. An example of post-processing can be found in the folder, called `PostProcessingVR.m`. You can open it and then run it to extract joint angles against time for example.

6.3 Tutorial #3: Lower limbs muscle forces estimation on a cycling motion

The first example is extracted from [19]. This research work consisted in linking the symmetry, the performance and the health during a cycling motion. The muscular symmetry is analyzed by the motion analysis containing an inverse kinematics step, an inverse dynamics step and a muscle forces estimation step. The markers set used is that called `Marker_set2` in CusToM. External forces applied on each foot were measured with two mobile platforms located on pedals.

Preparation

1. First of all, once MATLAB is launched, you need to run the `Installation` function. It permits to add the `CusToM\Functions` folder and its subfolders to the MATLAB path. This folder contains all the functions, models and dependencies used in the toolbox.
2. Second, make the `CusToM\Examples\Cycling` being the current working folder of MATLAB. The folder contains the C3D file and the CSV files to be processed. This is the classical way used in CusToM to launch an analysis: all of the data of a given subject is gathered in a unique folder being the current working folder of MATLAB. The folder also contains a post-processing script used at the end of the tutorial.

Generation of the Model Parameters

The following section enables the creation of the musculoskeletal model parameters. For more information about the GUI, please refer to section 4.1.

1. To generate the parameters of the model, execute `GenerateParameters` in the MATLAB command window.
2. Once the `GenerateParameters` GUI opened, first complete the size and weight of the subject. In the current example, the subject is $1.84m$ tall and weighs $70kg$.
3. Select the `Osteoarticular` tab. The model used here is a whole body model. It is necessary to select the different body parts to assemble the model.
   - In the `Trunk` section, select `PelvisLowerTrunk`, `UpperTrunkClavicle`, and `Skull` in the corresponding drop down menus. This is the classical analysis model used in CusToM, exhibiting 6 dofs between the pelvis (mobile base of the model) and the environment, 2 dofs in the spine (lower and upper trunk nodes), 2 dofs at the neck and 2 additional dofs allowing the upper limbs to raise during the motion (clavicle joints).
   - In the `Right Leg` and `Left Leg` sections, just select `Leg`. This is a classical leg model, with 3 dofs at the hip, 1 dof at the knee and 2 dofs at the ankle.
   - In the `Right Arm` and `Left Arm` sections, just select `Arm`. This is a classical arm model, with 3 dofs at the shoulder, 2 dofs at the elbow and 2 dofs at the wrist.
4. Select the `Markers tab`. In this example, a whole body set of markers has been used. You can select it with the drop down menu as `Marker_set2`. In the `Markers on hands` drop down menu, select 1 marker for the hands. This set of marker is less complete than the `Marker_set1` but is a classical ISB-friendly set.
5. Select the `Muscles` tab. This example is about estimating lower limbs muscle forces involved in cycling at a given speed. Therefore we need to equip the legs with muscles. Click the `Add` button, and select `LegMuscles` in the drop down menu. Select `Right` for the side. Proceed the same way to equip the left leg with muscles. The set of muscles is directly adapted from the simple leg model in [5]. You can see in the visualization section that the modifications made in the model update automatically the rendering.
6. All the parameters of the model are now properly set. You can generate the ModelParameters file by clicking `GenerateParameters`. The `Analysis` GUI is automatically opened.

Generation of the Analysis Parameters

The `Analysis` GUI is automatically opened once the `GenerateParameters` button is clicked. However, to run it independently from the `GenerateParameters` GUI, you can launch it with the command `Analysis`. For more information about the GUI, please refer to section 4.2. Please note that the Analysis GUI automatically load the ModelParameters to update the visualization of the model on the top right of the window. The Analysis GUI automatically lists the C3D files found in the current working folder of MATLAB.

1. In the C3D list on the left (files to process), select the `JOTH_Fin_125HzModif.c3d` file. Please note that you may here select as many files as you need to compute the same analysis on all of these files.
2. Click the `Inverse Kinematics Options`.
   - Check the step as active.
   - In the method drop down menu, select the `Levenberg-Marquardt` method that is the fastest available.
   - Check the joint coordinates filtering as active with a Cut-off frequency of $5Hz$.
   - Click `ok` to save the parameters and close the `Inverse Kinematics Options` tab.
3. Click the `Geometrical Calibration Options`.
   - Check the step as active.
   - In the `Frames` tab, select the method `UniformlyDistributed` and choose a number of frames of 30. This is the recommendations issued from the papers [11, 7].
   - In the `Body length` tab, make sure that both `RClavicle` and `LClavicle` lengths are linked to the `Thorax` length. If this is not the case, click the Add button to generate these constraints.
   - In the `Marker Position` tab, the markers default local directions to optimize are automatically loaded from the `Marker_set2`. You can modify it the way you want. In the current tutorial, we keep the default values.
   - In the `Axis of rotation` tab, you can also add some joint axes directions to optimize. In the current tutorial, none of the axes of rotations are optimized.
   - Click `ok` to save the parameters and close the `Geometrical Calibration Options` tab.
   - In the list below the `Geometrical Calibration` box, select the `JOTH_Fin_125HzModif.c3d` file to be used for the calibration. In this example, there is only one C3D file available, meaning that the calibration is made with the same file to be processed. In your own experimentation, feel free to dedicate a specific motion to the calibration, independently from the files to proceed with.
4. Click the `Inverse Dynamics Options`. Make sure that the step is active, and close the tab.
5. In the `External Forces` tab, select `from experiments`.
6. Click the `External forces Options`.
   - Check that the `External forces filtering` is active, with a cut-off frequency of $5Hz$.
   - In the drop down menu `Method to extract data`, select `Cycling`. In this specific example, the C3D file does not contain the external forces data, but 2 CSV files does. Therefore a specific function has been developed to extract the data from these files, in addition to the classical C3D file extraction. If such a feature interest you, you can explore the way the function has been developed. The function can be found in `...\Functions\ExternalForces\FromExperiments\Cycling.m`.
   - Click `ok` to save the parameters and close the `External forces Options` tab.
7. Click the `Muscle Forces Estimation options`.
   - Check that the step is active.

- In the `Method` drop down menu, select the `MusIC method`, an alternative solution to the classical optimization method that is fast and quasi-optimal [15].
- In the `Cost Function` drop down menu, choose $3rd$ order polynomial cost function.
- As recommended in [16], the database density parameters can be set to $k_1 = 4$ and $k_2 = 4$ to have an optimal database generation computation time with a minimal loss of accuracy.
- Click `ok` to save the parameters and close the `Muscle Forces Estimation options` tab.

8. Since all the analysis parameters have been set, you can run the analysis by clicking the `Run` button. The application automatically saves the AnalysisParameters, and launch the `Main` script containing the calls to the analysis functions and the model generation functions. The application first checks if the model exists. If this is not the case, the model is created. It may take some time (several preliminary computations are launched at this time), but it is executed once per subject. If the model already exists, the analysis is directly executed on the specified data with this model.

Post-processing

Once the analysis terminated, data can be post-processed the way you want.

1. To generate an animation of the motion, you can run the `GenerateAnimate` GUI from the MATLAB command window. Check all the boxes excepted the `external forces (prediction)` since no external forces were predicted in this study. Adapt the point of view the way you want, and finally click `Generate Video` to proceed. The film edition takes a while, since it aggregates frame per frame MATLAB figures. Additional visualization features will be added in the next developments of the toolbox.
2. An example of post-processing can be found in the folder, called `PostProcessingCycling.m`. You can open it and then run it to see what it does.

7 Research studies using CusToM

2018

- A. Muller, C. Pontonnier, and G. Dumont. The music method: a fast and quasi-optimal solution to the muscle forces estimation problem. *Computer methods in biomechanics and biomedical engineering*, 21(2):149–160, 2018
- A. Muller, C. Pontonnier, and G. Dumont. Music method enhancement by a sensitivity study of its performance: application to a lower limbs musculoskeletal model. *Computer Methods in Biomechanics and Biomedical Engineering*, 2018
- P. Puchaud, C. Sauret, A. Muller, N. Bideau, G. Dumont, H. Pillet, and C. Pontonnier. Preliminary comparison of eos-derived and geometrically calibrated segment lengths: inter-hip and femur cases. In *8th World Congress of Biomechanics 2018*, 2018
- P. Puchaud, C. Sauret, A. Muller, N. Bideau, G. Dumont, H. Pillet, and C. Pontonnier. Evaluation of geometrically calibrated segment lengths: preliminary results on inter-hip, femur and shank cases. In *XV International Symposium on 3D Analysis of Human Movement*, 2018

2017

- A. Muller, C. Pontonnier, and G. Dumont. Uncertainty propagation in multibody human model dynamics. *Multibody System Dynamics*, 40(2):177–192, 2017
- P. Plantard, A. Muller, C. Pontonnier, G. Dumont, H. P. Shum, and F. Multon. Inverse dynamics based on occlusion-resistant kinect data: Is it usable for ergonomics? *International Journal of Industrial Ergonomics*, 61:71–80, 2017
- A. L. Cruz Ruiz, C. Pontonnier, J. Levy, and G. Dumont. A synergy-based control solution for overactuated characters: Application to throwing. *Computer Animation and Virtual Worlds*, 28(6):e1743, 2017
- A. L. Cruz Ruiz, C. Pontonnier, and G. Dumont. Low-dimensional motor control representations in throwing motions. *Applied bionics and biomechanics*, 2017, 2017
- A. Muller. *Contributions méthodologiques à l'analyse musculo-squelettique de l'humain dans l'objectif d'un compromis précision performance*. PhD thesis, École normale supérieure de Rennes, 2017
- A. Muller, D. Haering, C. Pontonnier, and G. Dumont. Non-invasive techniques for musculoskeletal model calibration. In *Congrès Français de Mécanique*, 2017
- A. Muller, M. Chauwin, C. Pontonnier, and G. Dumont. Influence of physiological constraints on a subject-specific bsip calibration. In *XXVI Congress of the International Society of Biomechanics*, 2017
- A. Muller, F. Demore, C. Pontonnier, and G. Dumont. Music makes the muscles work together. In *XVI International Symposium on Computer Simulation in Biomechanics*, page 2, 2017

2016

- A. Muller, C. Germain, C. Pontonnier, and G. Dumont. A simple method to calibrate kinematical invariants: application to overhead throwing. In *ISBS-Conference Proceedings Archive*, volume 33, 2016
- A. Muller, C. Pontonnier, and G. Dumont. Uncertainty spreading from kinematics to dynamics in multibody human models. In *22nd Congress of the European Society of Biomechanics*, 2016

2015

- A. L. Cruz Ruiz, C. Pontonnier, J. Levy, and G. Dumont. Motion control via muscle synergies: Application to throwing. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pages 65–72. ACM, 2015
- A. Cruz Ruiz, C. Pontonnier, A. Sorel, and G. Dumont. Identifying representative muscle synergies in overhead football throws. *Computer methods in biomechanics and biomedical engineering*, 18(sup1):1918–1919, 2015
- A. Muller, C. Germain, C. Pontonnier, and G. Dumont. A comparative study of 3 body segment inertial parameters scaling rules. *Computer methods in biomechanics and biomedical engineering*, 18(sup1):2010–2011, 2015
- A. Muller, C. Pontonnier, C. Germain, and G. Dumont. Dealing with modularity of multibody models. *Computer methods in biomechanics and biomedical engineering*, 18(sup1):2008–2009, 2015

References

[1] A. Cruz Ruiz, C. Pontonnier, A. Sorel, and G. Dumont. Identifying representative muscle synergies in overhead football throws. *Computer methods in biomechanics and biomedical engineering*, 18(sup1):1918–1919, 2015.

[2] A. L. Cruz Ruiz, C. Pontonnier, and G. Dumont. Low-dimensional motor control representations in throwing motions. *Applied bionics and biomechanics*, 2017, 2017.

[3] A. L. Cruz Ruiz, C. Pontonnier, J. Levy, and G. Dumont. Motion control via muscle synergies: Application to throwing. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pages 65–72. ACM, 2015.

[4] A. L. Cruz Ruiz, C. Pontonnier, J. Levy, and G. Dumont. A synergy-based control solution for overactuated characters: Application to throwing. *Computer Animation and Virtual Worlds*, 28(6):e1743, 2017.

[5] M. Damsgaard, J. Rasmussen, S. T. Christensen, E. Surma, and M. De Zee. Analysis of musculoskeletal systems in the anybody modeling system. *Simulation Modelling Practice and Theory*, 14(8):1100–1111, 2006.

[6] S. Hilt, C. Pontonnier, and G. Dumont. Model based compensation for low mass objects haptic manipulation in virtual environments. In *International Conference on Virtual Reality and Augmented Reality*, pages 87–101. Springer, 2017.

[7] A. Muller. *Contributions méthodologiques à l'analyse musculo-squelettique de l'humain dans l'objectif d'un compromis précision performance*. PhD thesis, École normale supérieure de Rennes, 2017.

[8] A. Muller, M. Chauwin, C. Pontonnier, and G. Dumont. Influence of physiological constraints on a subject-specific bsip calibration. In *XXVI Congress of the International Society of Biomechanics*, 2017.

[9] A. Muller, F. Demore, C. Pontonnier, and G. Dumont. Music makes the muscles work together. In *XVI International Symposium on Computer Simulation in Biomechanics*, page 2, 2017.

[10] A. Muller, C. Germain, C. Pontonnier, and G. Dumont. A comparative study of 3 body segment inertial parameters scaling rules. *Computer methods in biomechanics and biomedical engineering*, 18(sup1):2010–2011, 2015.

[11] A. Muller, C. Germain, C. Pontonnier, and G. Dumont. A simple method to calibrate kinematical invariants: application to overhead throwing. In *ISBS-Conference Proceedings Archive*, volume 33, 2016.

[12] A. Muller, D. Haering, C. Pontonnier, and G. Dumont. Non-invasive techniques for musculoskeletal model calibration. In *Congrès Français de Mécanique*, 2017.

[13] A. Muller, C. Pontonnier, and G. Dumont. Uncertainty spreading from kinematics to dynamics in multibody human models. In *22nd Congress of the European Society of Biomechanics*, 2016.

[14] A. Muller, C. Pontonnier, and G. Dumont. Uncertainty propagation in multibody human model dynamics. *Multibody System Dynamics*, 40(2):177–192, 2017.

[15] A. Muller, C. Pontonnier, and G. Dumont. The music method: a fast and quasi-optimal solution to the muscle forces estimation problem. *Computer methods in biomechanics and biomedical engineering*, 21(2):149–160, 2018.

[16] A. Muller, C. Pontonnier, and G. Dumont. Music method enhancement by a sensitivity study of its performance: application to a lower limbs musculoskeletal model. *Computer Methods in Biomechanics and Biomedical Engineering*, 2018.

[17] A. Muller, C. Pontonnier, C. Germain, and G. Dumont. Dealing with modularity of multibody models. *Computer methods in biomechanics and biomedical engineering*, 18(sup1):2008–2009, 2015.

[18] P. Plantard, A. Muller, C. Pontonnier, G. Dumont, H. P. Shum, and F. Multon. Inverse dynamics based on occlusion-resistant kinect data: Is it usable for ergonomics? *International Journal of Industrial Ergonomics*, 61:71–80, 2017.

[19] C. Pouliquen. *Evaluation de l'asymétrie articulaire et musculaire au cours d'exercices exhaustif en cyclisme: apports de l'approche expérimentale et de la modélisation musculosquelettique*. PhD thesis, Université Rennes 2, 2017.

[20] P. Puchaud, C. Sauret, A. Muller, N. Bideau, G. Dumont, H. Pillet, and C. Pontonnier. Evaluation of geometrically calibrated segment lengths: preliminary results on inter-hip, femur and shank cases. In *XV International Symposium on 3D Analysis of Human Movement*, 2018.

[21] P. Puchaud, C. Sauret, A. Muller, N. Bideau, G. Dumont, H. Pillet, and C. Pontonnier. Preliminary comparison of eos-derived and geometrically calibrated segment lengths: inter-hip and femur cases. In *8th World Congress of Biomechanics 2018*, 2018.

8 Appendix 1: Marker sets

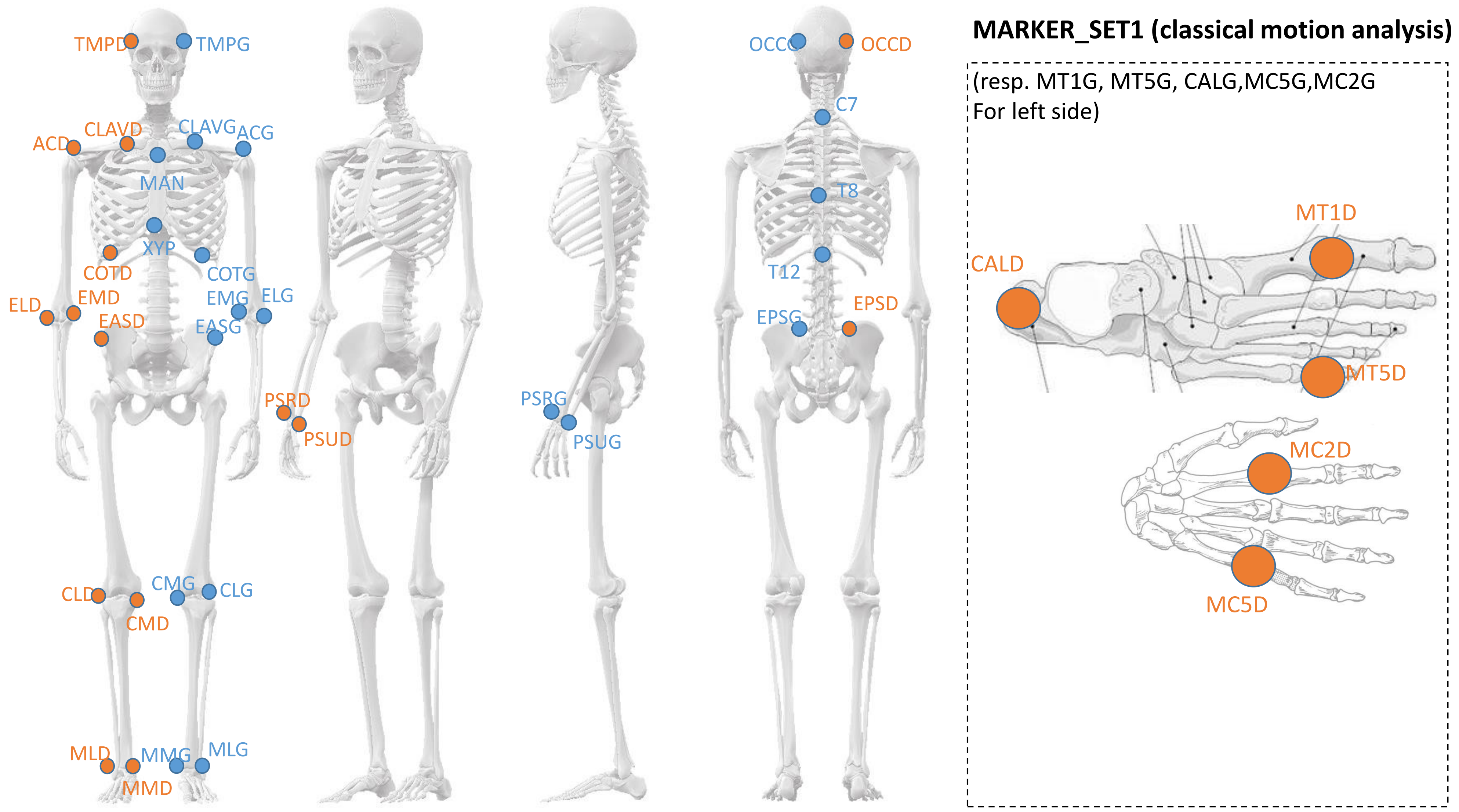


Figure 18: Marker_set1 definition

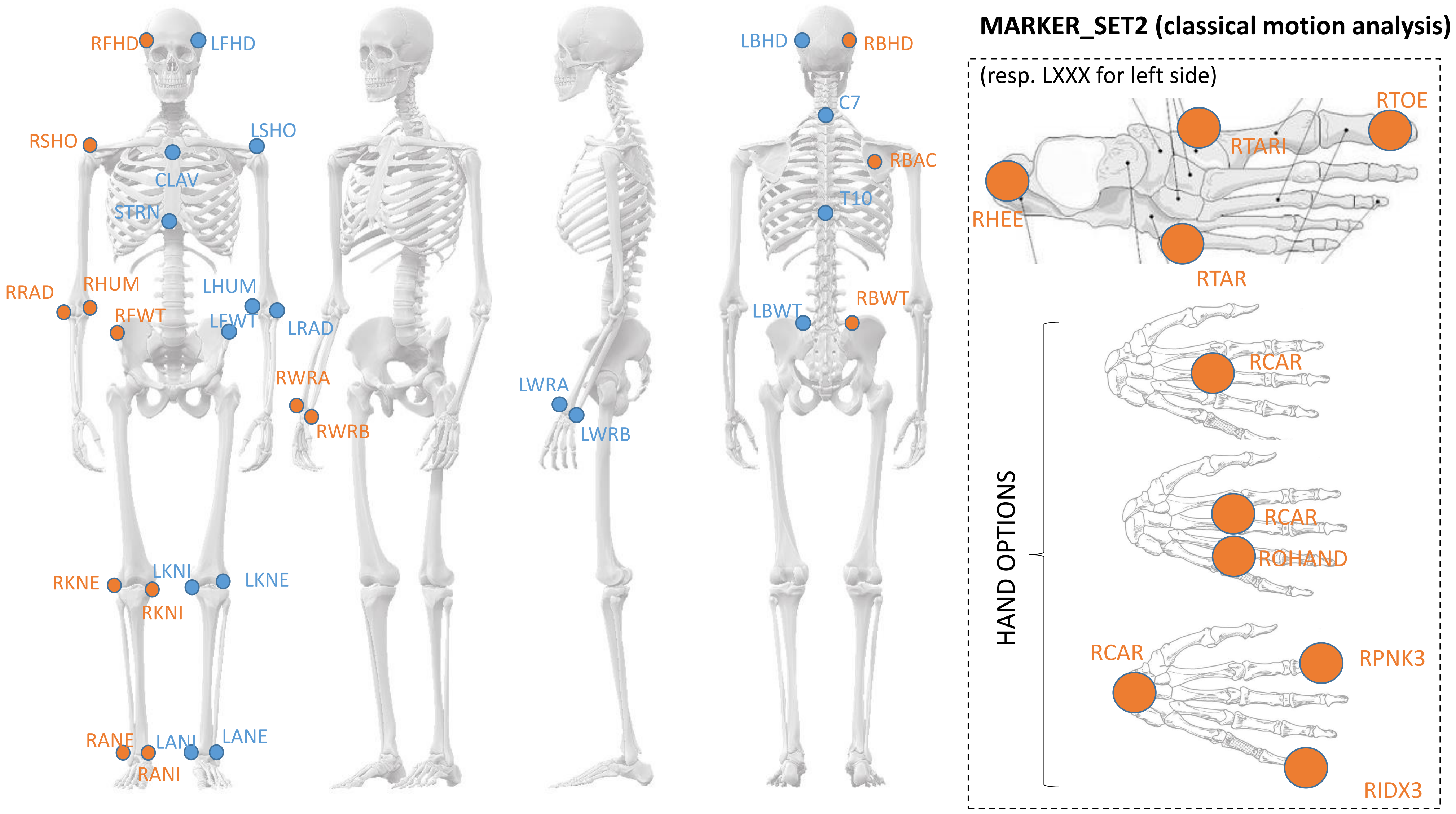


Figure 19: Marker_set2 definition

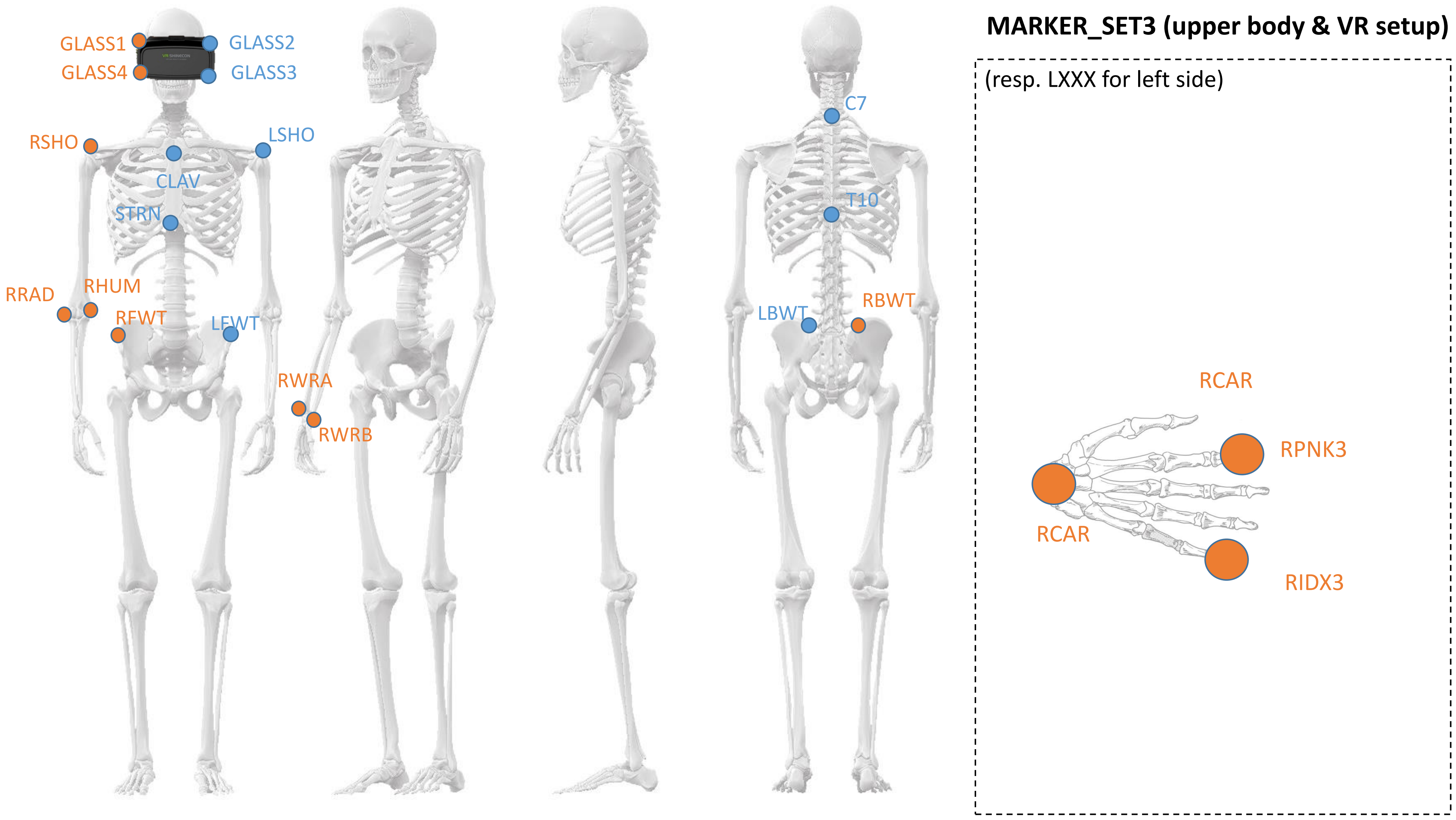


Figure 20: Marker_set3 definition

MARKER_SET4 (Kinect setup)

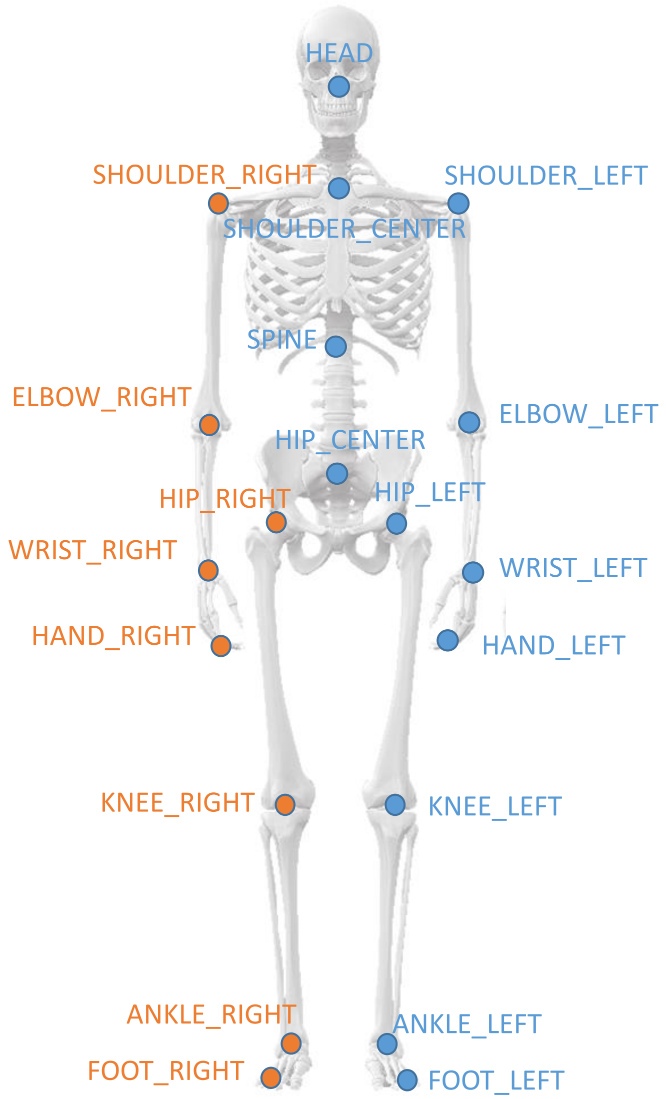


Figure 21: Marker_set4 definition

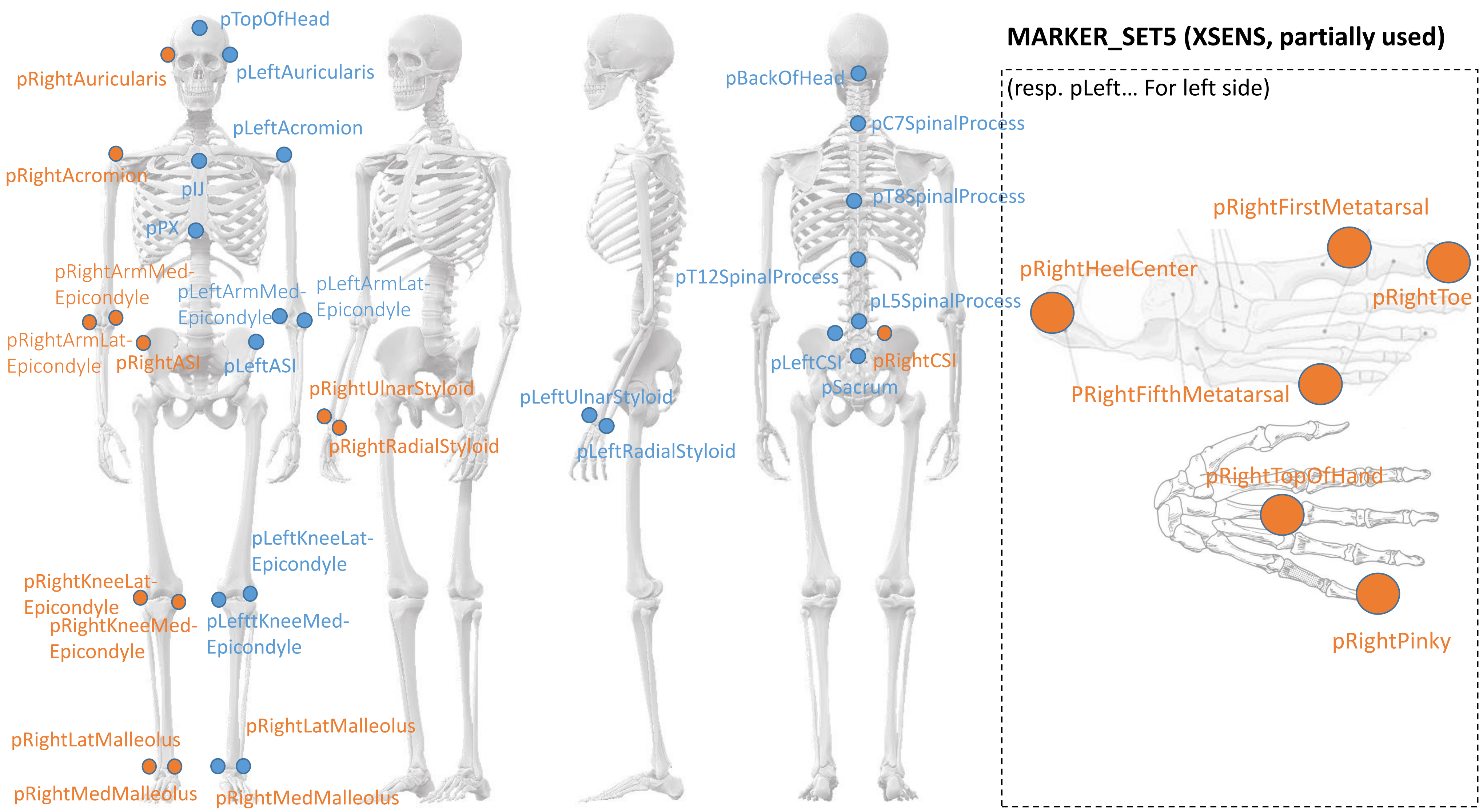


Figure 22: Marker_set5 definition