Adult-child musculoskeletal model and motion analysis

Course: Medical Robotics
Professor: Marinela Vendittelli

Authors:
Catia Carocci
Denise Landini – 1938388
Alessandro Lambertini – 1938390

La Sapienza University - Roma
Academic year: 2020-2021

Summary

1. INPUT DATA

In the first part of the project is required:

- The processing of the raw data collected through motion capture (*MOCAP*) relating to 18 walking trials of a healthy 7-year-old girl (Sirine) and 6 walking trials of her twin that has spastic CP (Lina);
- The visualization of the same data in OpenSim using a simplified model in which the upper part of the body is approximated only to the torso.

In the second part we are going to use the processed data in order to do the **equilibrium analysis** of the twins:

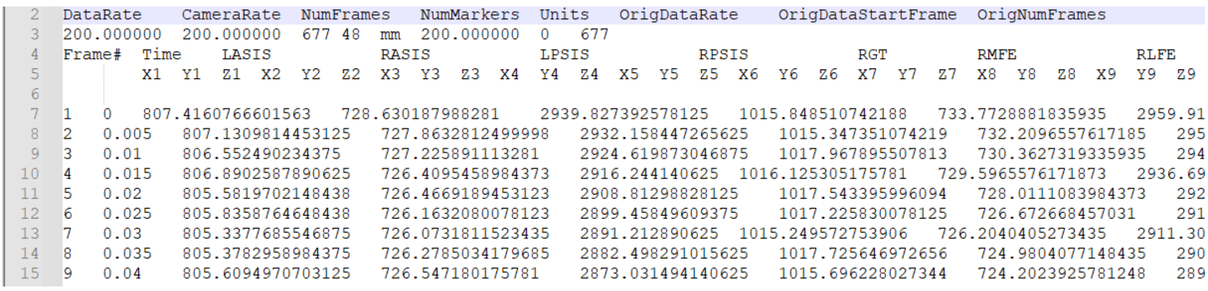
1. Scaling of the muscoloskeletal models;
2. Kinematic and Dynamic analysis;
3. Computation of the distance between Minimum Moment Axis (Δ) and the Center of Mass (CoM) [3].

The useful data of each brand are stored in .c3d format files, through which it is possible to store information on the markers, EMG, force plate and events in binary form. It was therefore necessary to convert the .c3d files into OpenSim compatible formats. The conversion was implemented in Matlab through the **`c3dExport.m`** [1] function which allowed data to be written in .trc and .mot formats.

- **.mot** files contain information about the forces and moments measured by the force plates during walking;
- **.trc** files specify the positions of the different markers in time during the motion capture trial.

1.1. TRC FILES (Marker Data)

The files have been processed in **`dataProcessing.m`** and used for viewing the various trials in OpenSim and are presented as follows:



```
2  DataRate    CameraRate  NumFrames   NumMarkers  Units   OrigDataRate    OrigDataStartFrame  OrigNumFrames
3  200.000000  200.000000  677 48  mm  200.000000  0   677
4  Frame#  Time    LASIS           RASIS           LPSIS           RPSIS           RGT         RMFE            RLFE
5          X1  Y1  Z1  X2  Y2  Z2  X3  Y3  Z3  X4  Y4  Z4  X5  Y5  Z5  X6  Y6  Z6  X7  Y7  Z7  X8  Y8  Z8  X9  Y9  Z9
6  
7  1   0   807.4160766601563   728.630187988281    2939.827392578125   1015.848510742188   733.7728881835935   2959.91
8  2   0.005   807.1309814453125   727.8632812499998   2932.158447265625   1015.347351074219   732.2096557617185   295
9  3   0.01    806.552490234375    727.225891113281    2924.619873046875   1017.967895507813   730.3627319335935   294
10 4   0.015   806.8902587890625   726.4095458984373   2916.244140625  1016.125305175781   729.5965576171873   2936.69
11 5   0.02    805.5819702148438   726.4669189453123   2908.81298828125    1017.543395996094   728.0111083984373   292
12 6   0.025   805.8358764648438   726.1632080078123   2899.45849609375    1017.225830078125   726.672668457031    291
13 7   0.03    805.3377685546875   726.0731811523435   2891.212890625  1015.249572753906   726.2040405273435   2911.30
14 8   0.035   805.3782958984375   726.2785034179685   2882.498291015625   1017.725646972656   724.9804077148435   290
15 9   0.04    805.6094970703125   726.547180175781    2873.031494140625   1015.696228027344   724.2023925781248   289
```

Figure 1: *Visualization of marche 18.trc of Sirine by using Notepad++.*

As we can see in **Figure 1**, in lines 2-3, for each of the 18 measurements (6 for Lina) we have information relating to:

- Sampling frequency [Hz] (DataRate);
- Total number of frames (NumFrames);
- Total number of markers (NumMarkers);
- Units of measurement of recorded values (Units).

The first column of line 4 (Frame#) refers to the numbering of the frames; The second column of line 4 (Time) indicates the instant in which that data was acquired.

The following columns list the different markers used in order; through the corresponding name you have information on the particular position of the marker on the subject's body (i.e., LASIS = Left Antero Superior Iliac Spine). For each marker there are three values which correspond to the spatial coordinated (x, y, z) for each instant of time.

In order to read the data from the file .trc more efficiently through Matlab, the **`loadDataFromTRC.m`** function was used, which requires as input the .trc file that you want to convert into variable or objects that it inserts in the workspace.

So, we build a matrix (m_cols) whose first two columns represent the frames and the time instants in which they were acquired, and the remaining ones are precisely the x, y and z components of each marker. Every three columns there are data relating to a different marker, as we can see in **Figure 2**.

Variables - m_cols

m_cols

677x147 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0	807.4161	728.6302	2.9398e+03	1.0158e+03	733.7729	2.9599e+03	855.9110	762.9460	3.0699e+03	928.3724
2	2	0.0050	807.1310	727.8633	2.9322e+03	1.0153e+03	732.2097	2.9523e+03	856.3193	761.7199	3.0621e+03	928.4947
3	3	0.0100	806.5525	727.2259	2.9246e+03	1.0180e+03	730.3627	2.9433e+03	856.6849	760.6412	3.0544e+03	928.5626
4	4	0.0150	806.8903	726.4095	2.9162e+03	1.0161e+03	729.5966	2.9367e+03	857.0802	759.7244	3.0464e+03	928.7776
5	5	0.0200	805.5820	726.4669	2.9088e+03	1.0175e+03	728.0111	2.9275e+03	857.4123	758.9622	3.0381e+03	928.9373
6	6	0.0250	805.8359	726.1632	2.8995e+03	1.0172e+03	726.6727	2.9189e+03	857.4697	758.3734	3.0298e+03	928.9451
7	7	0.0300	805.3378	726.0732	2.8912e+03	1.0152e+03	726.2040	2.9113e+03	857.4058	757.9381	3.0211e+03	929.0350
8	8	0.0350	805.3783	726.2785	2.8825e+03	1.0177e+03	724.9804	2.9024e+03	856.9780	757.8494	3.0124e+03	928.8389
9	9	0.0400	805.6095	726.5472	2.8730e+03	1.0157e+03	724.2024	2.8939e+03	856.3708	758.0247	3.0034e+03	928.8514

Figure 2: *Matrix of the measurement (m_cols) build by the function loadDataFromTRCe.m of marche 18.trc of Sirine.*

1.2. MOT FILES (Force Plate Data)

The files have been processed in **`dataProcessing.m`** and used for viewing the various trials in OpenSim and are presented as follows:

```
nColumns=46
nRows=7490
DataType=double
version=3
OpenSimVersion=4.2-2021-03-12-fcedec9
endheader
time	ground_force_1_vx	ground_force_1_vy	ground_force_1_vz	ground_force_1_px	ground_force_1_py	ground_force_1_pz	ground_moment_1_mx
ground_moment_1_my	ground_moment_1_mz	ground_force_2_vx	ground_force_2_vy	ground_force_2_vz	ground_force_2_px	ground_force_2_py	ground_force_2_pz
ground_moment_2_mx	ground_moment_2_my	ground_moment_2_mz	ground_force_3_vx	ground_force_3_vy	ground_force_3_vz	ground_force_3_px	ground_force_3_py
ground_force_3_pz	ground_moment_3_mx	ground_moment_3_my	ground_moment_3_mz	ground_force_4_vx	ground_force_4_vy	ground_force_4_vz	ground_force_4_px
ground_force_4_py	ground_force_4_pz	ground_moment_4_mx	ground_moment_4_my	ground_moment_4_mz	ground_force_5_vx	ground_force_5_vy	ground_force_5_vz
ground_force_5_px	ground_force_5_py	ground_force_5_pz	ground_moment_5_mx	ground_moment_5_my	ground_moment_5_mz
0	0	0	0	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-15.72464278179867	0.1700239777565002	0.1212794580159315
-3.223367776011851	7.236644215575843e-18	-0.1181833687984858	7.105427357600809e-18	-8.95685769546486	-5.484493537064746e-16	0	0	0	-nan(ind)
-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	0	0	0	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	0	0	0
-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)
```

***Figure 3**: Visualization of marche 1.mot of Sirine by using Notepad++.*

As we can see in **Figure 3**, in this type of file we have information about:

- Number of columns (nColumns);
- Number of rows (nRows);
- Type of the data (DataType);
- Version of OpenSim;

- Endheader

After this, we have the first column in which there is the time and then for each force we have (x, y, z) coordinates for the right/left foot's **ground reaction force** vector, **CoP (center of pressure)** of the right/left foot and the right/left foot's **ground reaction momentum** vector.

In order to read the data from the file .mot more efficiently through Matlab, the **`loadDataFromMOT.m`** function was used, which requires as input the .mot file that you want to convert into variable or objects that it inserts in the workspace.

Variables - M_FM

M_FM

6770x46 double

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	-49.4155	4.4899	0.9560
2	5.0000e-04	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	-21.0015	0.7483	1.6478
3	1.0000e-03	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
4	0.0015	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	18.4457	-3.2825	0.7343
5	0.0020	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
6	0.0025	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
7	0.0030	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
8	0.0035	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
9	0.0040	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
10	0.0045	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
11	0.0050	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0

Figure 4*: Matrix of the measurement (M_FM) build by the function loadDataFromMOT.m of marche 18.trc of Sirine.*

The data matrix (*M_FM*) first column represent the time instants in which forces and momentums were acquired and the remaining ones are precisely the x, y and z components of ground reaction force vectors, CoP and ground reaction momentum vectors.

2. GAP FILLING

In our marches we have identified some NaN (Not Available Number) in the 3D markers. We have done the gap filling step before smoothing the marker's trajectories because if we filter the entire signal, we will reduce the distortion of the filtered signal.

In order to do this, we have used Matlab and we have identified the "holes" in the trajectory of each marker of the body. To complete this step, we have uploaded the .trc file and we have displayed the trajectory with the hole.

In **Figure 5**, we can see an example taken from marche 12 of Sirine in which we can see the gaps in LATT marker's trajectory.
The 1^{st}plot is the trajectory of the x-component that has a gap in the interval [666, 667],
the 2^{nd} plot is the trajectory of the y-component that has a gap in the interval [669, 671],
the 3^{rd} plot is the trajectory of the z-component that has a gap in the interval [747, 749].

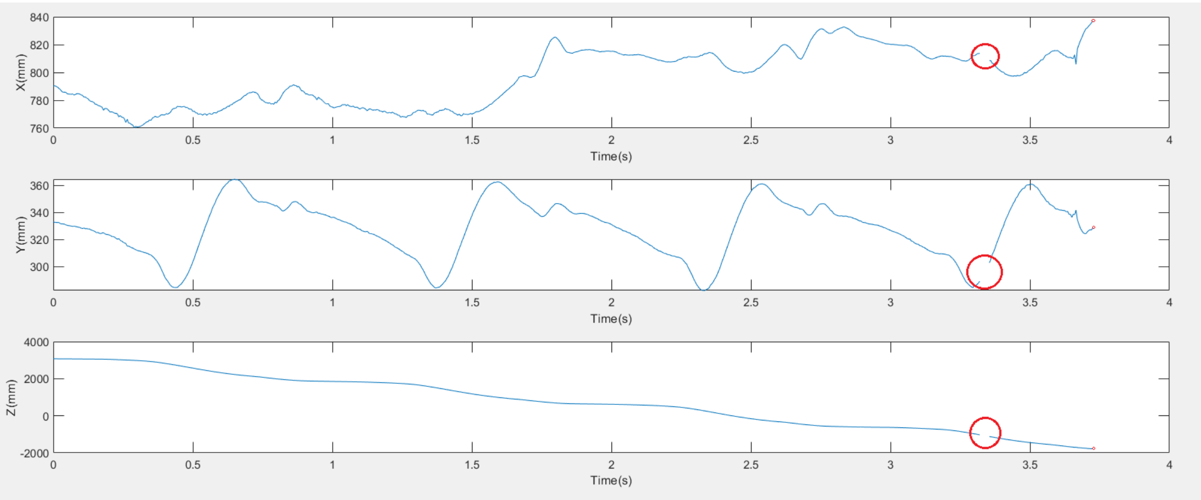


***Figure 5**: Gap filling over the x, y, z coordinates of the LATT marker of Sirine.*

As we can see in the **Table 1**, for each marche we have reported the number of markers, the number of frames, the markers in which we have identify the error, the name of the marker and the gap frames in that marker.

For example, in the Marche 6 of Sirine, we have identified three error in the markers:

- [27, 28, 29] that corresponds to LMFE marker and the gap is in the frame's interval [742-750];
- [33, 34, 35] that corresponds to RATT marker and the gap is in the frame's interval [747-750];
- [42, 43, 44] that corresponds to LATT marker and the gap is in the frame's interval [743-747].

Marche	Markers	Frames	Marker E	Name Marker	Gap E
1	147	749	-	-	-
2	147	661	-	-	-
3	198	800	[159->197]	-	-
4	201	600	[33,34,35]	RATT	[600]
			[168->200]	-	[…]
5	207	560	[156->206]	-	[…]
6	147	750	[27, 28, 29]	LMFE	[742, 750]
			[33, 34, 35]	RATT	[747, 750]
			[42, 43, 44]	LATT	[743, 747]
7	147	800	[33, 34, 35]	RATT	[799, 800]
8	147	500	[33, 34, 35]	RATT	[500]
9	147	450	-	-	-
10	147	850	[54, 55, 56]	RTT2	[850]
11	147	1054	[27, 28, 29]	LMFE	[1049, 1051], [1054]
			[42, 43, 44]	LATT	[1047, 1053]
12	147	750	[42, 43, 44]	LATT	[666, 667],[669,671],[747,749]
13	147	750	-	-	-
14	147	650	[33, 34, 35]	RATT	[650]
15	147	650	-	-	-
16	147	700	[33, 34, 35]	RATT	[699,700]
17	147	700	[33, 34, 35]	RATT	[686,687],[689,691],[693,700]
			[54,55,56]	RTT2	[698,700]
18	147	677	[6,7,8]	RASIS	[674,676]
			[27,28,29]	LMFE	[140,142],[672]
			[33, 34, 35]	RATT	[677]
			[42,43,44]	LATT	[668],[670,675],[677]
			[72,73,74]	LMFH1	[677]
			[78,79,80]	OCC	[5,8],[10,13]

***Table 1**: summary table for the gaps present in the Sirine's trajectories.*

Marche	Markers	Frames	Marker E	Name Marker	Gap E
2	147	1714	[3, 4, 5]	RASIS	[1667 ~ 1675],[1713, 1714]
			[24, 25, 26]	LATT	[1647 ~ 1704],[1712,1713]
			[42, 43, 44]	LMFH1	[1586, 1602]
			[48, 49, 50]	RMFE	[1705, 1714]
			[51, 52, 53]	RLFE	[1624 ~ 1703], [1711, 1714]
			[54, 55, 56]	RATT	[1642 ~ 1647], [1714]
			[66, 67, 68]	RTT2	[1672 ~ 1681], [1689 ~ 1714]
			[141, 142, 143]	RFT3	[1698 ~ 1706], [1712, 1714]
3	147	1748	[3, 4, 5]	RASIS	[1742, 1748]
			[102, 103, 104]	LMHE	[1748]
4	147	1780	[15, 16, 17]	LGT	[12, 13],[15, 16]
5	147	1770	[144, 145, 146]	LGT	[1, 49]
6	147	2201	[15, 16, 17]	LGT	[417, 420],[422, 426]
			[51, 52, 53]	RATT	[2201]
			[108., 109, 110]	LUS	[242, 243], [245, 246]
			[144, 145, 146]	RMFE	[1, 12]
7	147	1897	[24, 25, 26]	LSPH	[1841, 1842],[1844, 1845],[1896, 1897]
			[33, 34, 35]	LTT2	[1471, 1897]
			[48, 49, 50]	RLFE	[1805, 1809],[1811, 1813]
			[51, 52, 53]	RATT	[1897]
			[87, 88, 89]	C7	[1883, 1897]
			[144, 145, 146]	LGT	[1]

***Table 2:** summary table for the gaps present in the Lina's trajectories.*

After having found the gaps, we have to eliminate it.

To do this, we have decided to use the **linear regression method**. In detail, for each interval, we have used different degrees of the polynomial according to the best estimation that it produces. We have implemented this in **`GapFiller.m`**.

In **Figure 6**, we can see the trajectory without gaps of the marche 12 of Sirine.

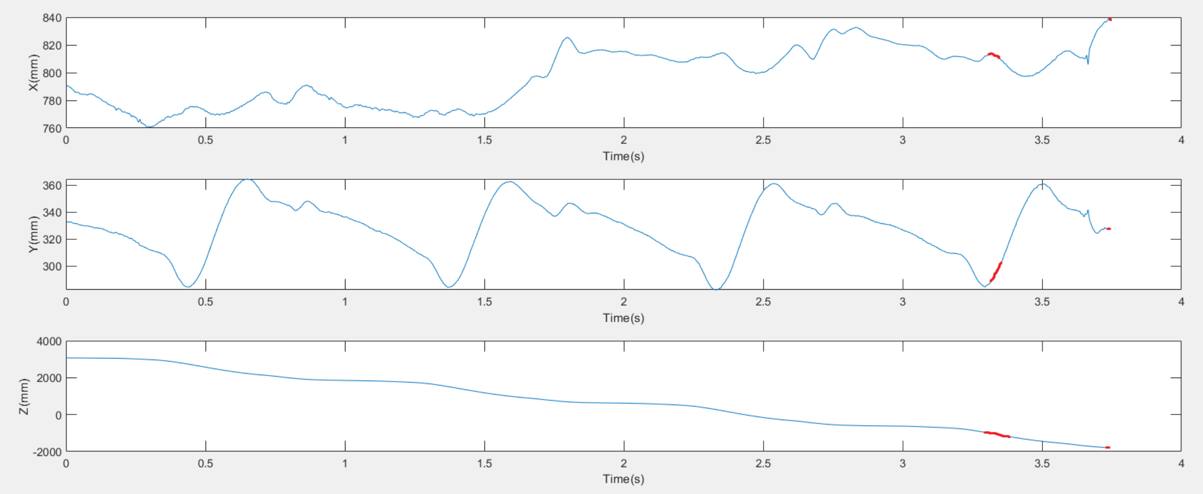


Figure 6: *Gap filling over the x, y, z LATT marker coordinates of Sirine. The blue line represents the signal with gaps, while the red part represents the reconstructed trajectory by using the linear regression method.*

3. FILTERING

Once we have performed gap filling, the next step is to use low-pass filters to turn the noise off from our signal of interest.

Using **`dataProcessing.m`** file, we designed a Butterworth filter of order 4. This kind of filter is used since it has the characteristic of having "maximally flat" response in the frequency band of interest, it does not produce ripples over the frequencies you want to preserve in the output signal. A fixed cutoff frequency is chosen to be used for all the different trials and all components.

We uploaded the .trc file in Matlab, previously manipulated through gap filling, and then we applied the filter.

In **Figure 7** is displayed LATT marker's filtered trajectory of the marche 12 of Sirine, previously filled in **Figure 6**. We have chosen the sampling frequency to 200 Hz for Sirine, 2000 Hz for Lina and a fixed cutoff frequency to 10 Hz.

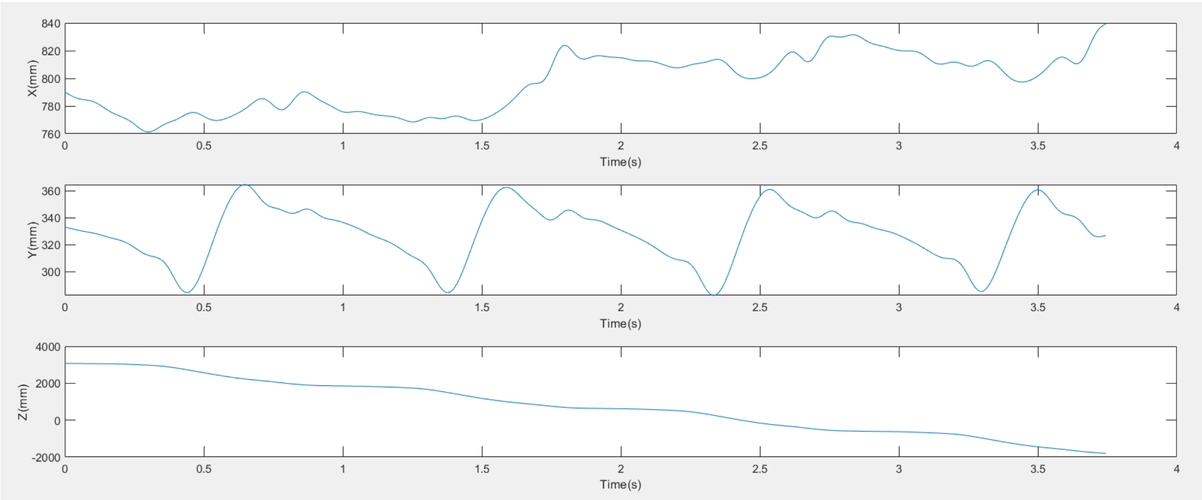


***Figure 7**: Filtered x, y, z LATT marker coordinates of Sirine (marche 12) using Butterworth filter of order 4.*

At the end of the filtering all the new coordinates are put in a .trc file that can be used in OpenSim and applied to the simplified Sirine/Lina model to run the manipulated data.

Further work done on the given code

At the beginning we have **`Sirine_model.m`** which contains the reading of the file and the plot of the model. So, we have decided to create 2 files for reading:

- .trc file → **`loadDataFromTRC.m`**;
- .mot file → **`loadDataFromMOT.m`**;

in which we have also improved the reading performance to make it faster.

We have also created 2 files to save the data after gap filling and filtering phases for:

- .trc file → **`saveDataOnFileTRC.m`**;
- .mot file → → **`saveDataOnFileMOT.m`**;

We have created the **`dataProcessing.m`** file in order to make the data processing phase totally automatic.

We have also created **`plotModel.m`** file to read the .trc file and to make the needed structure for plot; this file is able to plot every marche that has the marker that we need.

4. SCALING OF THE MUSCULOSKELETAL

Sirine and Lina are two true twins' sister. They have 7 years old. Sirine is 1.20 m and the mass is 20 Kg. She is healthy. Lina is 1.05 m and the mass is 17.6 Kg. She has spastic CP.

We have 48 markers placed to record the 3D motions. We need to adapt the location of the virtual markers on the body model to the experimental ones.

In the following table, we can see the final marker pairs chosen for each segment.
We have analyzed only the lower part of the body because this is our part of interest.

	x-scaling We consider the x-component		Sirine Values [m]		Lina Values [m]	
	Left	**Right**	**Left**	**Right**	**Left**	**Right**
Torso	$\|\|T10-STR\|\|$		0.992915		1.061081	
Pelvis	$\|\|RPSIS-RASIS\|\|$		1.292585		1.242569	
Femur	$\|\|LGT-LLFE\|\|$ $\|\|LGT-LMFE\|\|$	$\|\|RGT-RLFE\|\|$ $\|\|RGT-RMFE\|\|$	1.037906	1.025449	0.946882	0.972806
Tibia	z of $\|\|LLM-LSPH\|\|$	z of $\|\|RLM-RSPH\|\|$	1.028973	1.055212	0.941571	1.005329
Talus	$\|\|LCAL-LTT2\|\|$, $\|\|LSPH-LLM\|\|$	$\|\|RCAL-RTT2\|\|$, $\|\|RSPH-RLM\|\|$	1.042535	1.036238	0.904046	0.929530
Calcaneus	$\|\|LCAL-LTT2\|\|$, $\|\|LSPH-LLM\|\|$	$\|\|RCAL-RTT2\|\|$, $\|\|RSPH-RLM\|\|$	1.042535	1.036238	0.904046	0.929530
Toes	$\|\|LCAL-LTT2\|\|$, $\|\|LSPH-LLM\|\|$	$\|\|RCAL-RTT2\|\|$, $\|\|RSPH-RLM\|\|$	1.042535	1.036238	0.904046	0.929530

	y-scaling We consider the y-component		Sirine Values [m]		Lina Values [m]	
	Left	**Right**	**Left**	**Right**	**Left**	**Right**
Torso	$\|\|C7-LPSIS\|\|$	$\|\|C7-RPSIS\|\|$	1.052912		1.131802	1.131802
Pelvis	$\|\|LGT-LASIS\|\|$	$\|\|RGT-RASIS\|\|$	0.912367		1.076113	1.076113
Femur	$\|\|LGT-LLFE\|\|$ $\|\|LGT-LMFE\|\|$	$\|\|RGT-RLFE\|\|$ $\|\|RGT-RMFE\|\|$	1.037906	1.025449	0.946882	0.972806
Tibia	$\|\|LMFE-LSPH\|\|$	$\|\|RMFE-RSPH\|\|$	1.091638	1.099436	0.985154	0.958496
Talus	$\|\|LCAL-LTT2\|\|$, $\|\|LSPH-LLM\|\|$	$\|\|RCAL-RTT2\|\|$, $\|\|RSPH-RLM\|\|$	1.042535	1.036238	0.904046	0.929530
Calcaneus	$\|\|LCAL-LTT2\|\|$, $\|\|LSPH-LLM\|\|$	$\|\|RCAL-RTT2\|\|$, $\|\|RSPH-RLM\|\|$	1.042535	1.036238	0.904046	0.929530
Toes	$\|\|LCAL-LTT2\|\|$, $\|\|LSPH-LLM\|\|$	$\|\|RCAL-RTT2\|\|$, $\|\|RSPH-RLM\|\|$	1.042535	1.036238	0.904046	0.929530

	z-scaling We consider the z-component		Sirine Values [m]		Lina Values [m]	
	Left	**Right**	**Left**	**Right**	**Left**	**Right**
Torso	x of $\|\|C7-midPSIS\|\|$		0.992915		1.061081	
Pelvis	$\|\|RASIS-LASIS\|\|$		0.941311		0.753315	
Femur	$\|\|LGT-LLFE\|\|$ $\|\|LGT-LMFE\|\|$	$\|\|RGT-RLFE\|\|$ $\|\|RGT-RMFE\|\|$	1.037906	1.025449	0.946882	0.972806
Tibia	$\|\|LLM-LSPH\|\|$	$\|\|RLM-RSPH\|\|$	1.028973	1.055212	0.941571	1.005329
Talus	$\|\|LCAL-LTT2\|\|$, $\|\|LSPH-LLM\|\|$	$\|\|RCAL-RTT2\|\|$, $\|\|RSPH-RLM\|\|$	1.042535	1.036238	0.904046	0.929530
Calcaneus	$\|\|LCAL-LTT2\|\|$, $\|\|LSPH-LLM\|\|$	$\|\|RCAL-RTT2\|\|$, $\|\|RSPH-RLM\|\|$	1.042535	1.036238	0.904046	0.929530
Toes	$\|\|LCAL-LTT2\|\|$, $\|\|LSPH-LLM\|\|$	$\|\|RCAL-RTT2\|\|$, $\|\|RSPH-RLM\|\|$	1.042535	1.036238	0.904046	0.929530

In OpenSim we do the following steps:

1) We use the Scale Tool and in Settings we load the statique.trc file. The statique.trc file is obtained from the statique.c3d file by executing **`dataProcessing.m`**, in matlab. We do the same process for Lina, but in this case, we load the statique02.trc file, that is obtained in the same way as before.

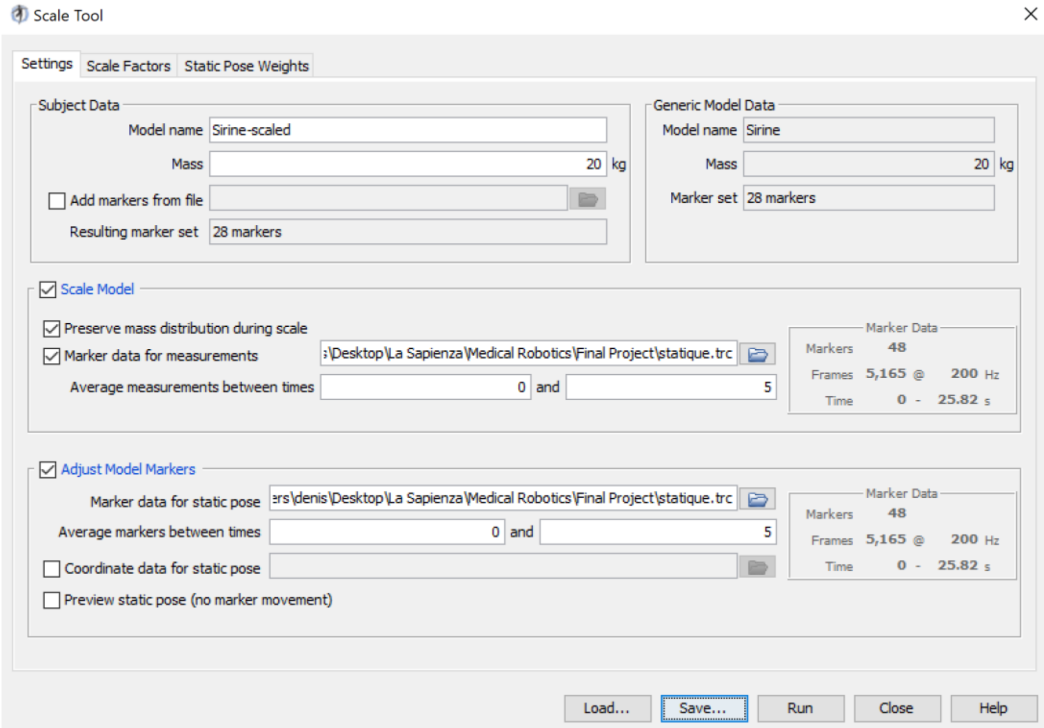
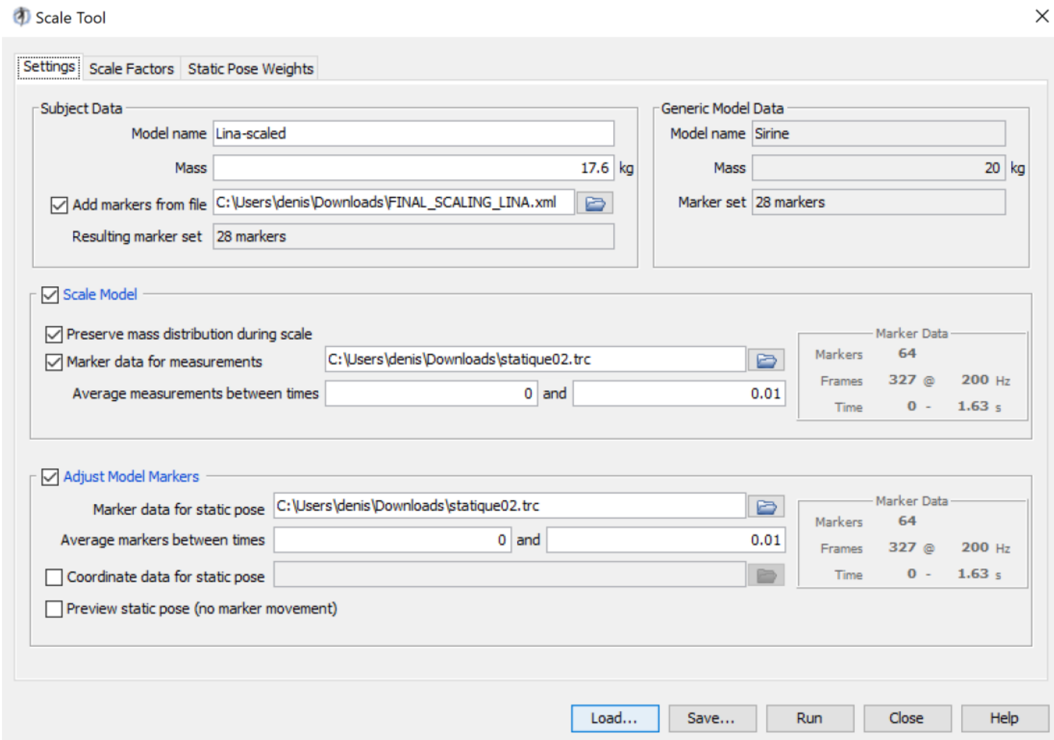
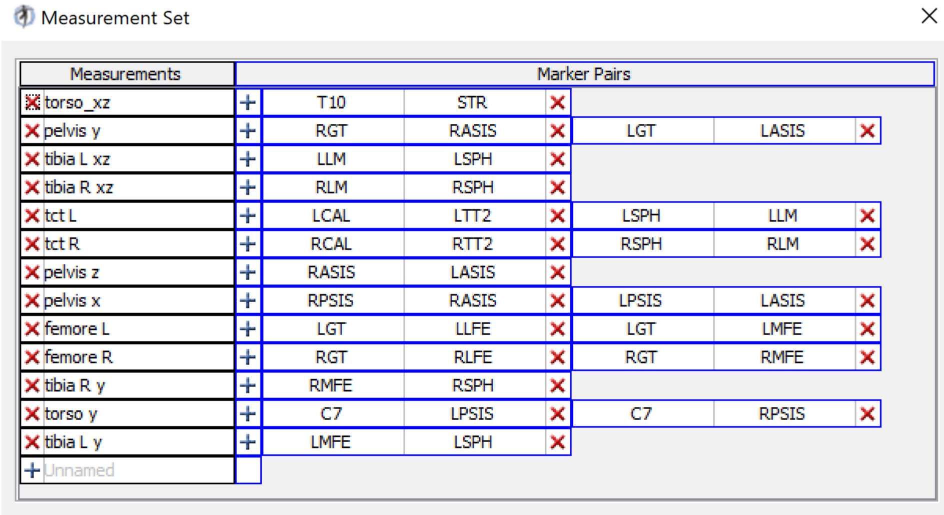


Figure 8: *Sirine Settings for scaling.*



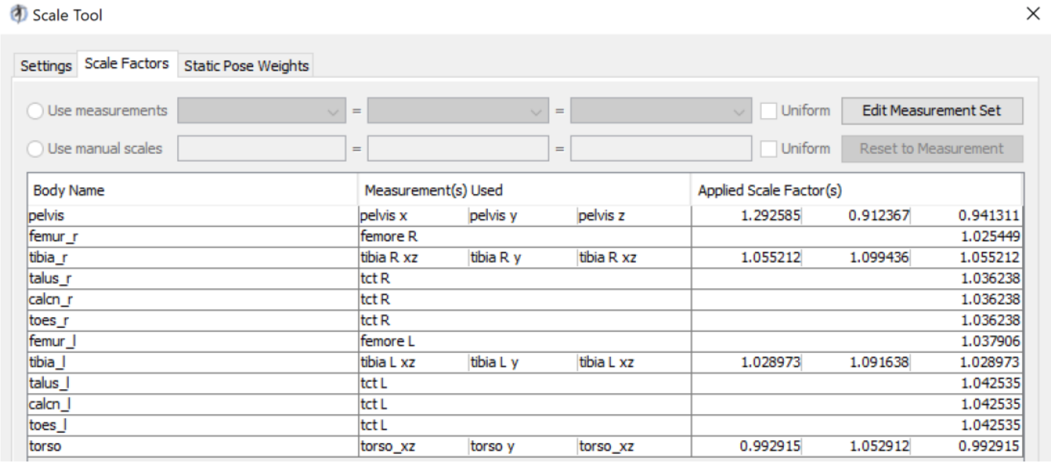
2) In Scale Factors we edit the measurement set, by following the previous tables. This is the same for both Sirine and Lina scaled model.


Measurement Set

Measurements	Marker Pairs			
torso_xz	T10	STR		
pelvis y	RGT	RASIS	LGT	LASIS
tibia L xz	LLM	LSPH		
tibia R xz	RLM	RSPH		
tct L	LCAL	LTT2	LSPH	LLM
tct R	RCAL	RTT2	RSPH	RLM
pelvis z	RASIS	LASIS		
pelvis x	RPSIS	RASIS	LPSIS	LASIS
femore L	LGT	LLFE	LGT	LMFE
femore R	RGT	RLFE	RGT	RMFE
tibia R y	RMFE	RSPH		
torso y	C7	LPSIS	C7	RPSIS
tibia L y	LMFE	LSPH		
Unnamed				

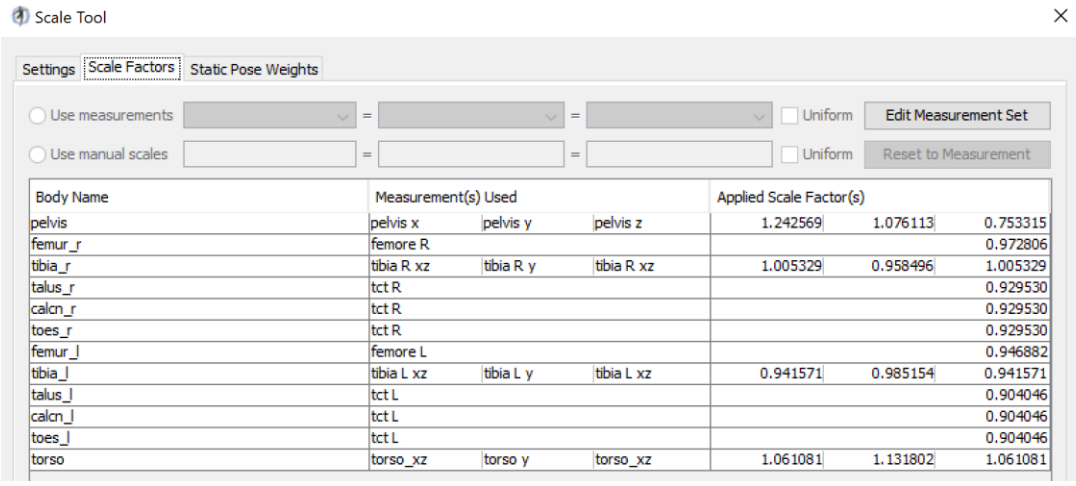
Figure 10: *Sirine and Lina Measurement set for scaling.*

3) By running this, we obtain the following scaling factors for Sirine and Lina resepectively.



Body Name	Measurement(s) Used			Applied Scale Factor(s)		
pelvis	pelvis x	pelvis y	pelvis z	1.292585	0.912367	0.941311
femur_r	femore R					1.025449
tibia_r	tibia R xz	tibia R y	tibia R xz	1.055212	1.099436	1.055212
talus_r	tct R					1.036238
calcn_r	tct R					1.036238
toes_r	tct R					1.036238
femur_l	femore L					1.037906
tibia_l	tibia L xz	tibia L y	tibia L xz	1.028973	1.091638	1.028973
talus_l	tct L					1.042535
calcn_l	tct L					1.042535
toes_l	tct L					1.042535
torso	torso_xz	torso y	torso_xz	0.992915	1.052912	0.992915

Figure 11: *Scaling factors of Sirine.*



Body Name	Measurement(s) Used			Applied Scale Factor(s)		
pelvis	pelvis x	pelvis y	pelvis z	1.242569	1.076113	0.753315
femur_r	femore R					0.972806
tibia_r	tibia R xz	tibia R y	tibia R xz	1.005329	0.958496	1.005329
talus_r	tct R					0.929530
calcn_r	tct R					0.929530
toes_r	tct R					0.929530
femur_l	femore L					0.946882
tibia_l	tibia L xz	tibia L y	tibia L xz	0.941571	0.985154	0.941571
talus_l	tct L					0.904046
calcn_l	tct L					0.904046
toes_l	tct L					0.904046
torso	torso_xz	torso y	torso_xz	1.061081	1.131802	1.061081

Figure 12: *Scaling factors of Lina.*

	scaled mass [Kg]	CoM [m]
pelvis	3.16042292829541	(-0.0389535 0 0)
femur_r	2.49608200944611	(0 -0.133551 0)
tibia_r	0.99492808072134	(0 -0.151816 0)
talus_r	0.0268355517389438	(0 0 0)
calcn_r	0.335444396736797	(0.0744395 0.0223319 0)
toes_r	0.0581258050665522	(0.0257561 0.00446637 -0.013027)
femur_l	2.49608200944611	(0 -0.135109 0)
tibia_l	0.99492808072134	(0 -0.1465 0)
talus_l	0.0268355517389438	(0 0 0)
calcn_l	0.335444396736797	(0.0739452 0.0221836 0)
toes_l	0.0581258050665522	(0.0255851 0.00443672 0.0129405)
torso	9.0167453842851	(0.00222114 0.175516 0)
tot.	**20.00000000000000**	

	scaled mass [Kg]	CoM [m]
pelvis	2.78117217689996	(-0.0374462 0 0)
femur_r	2.19655216831257	(0 -0.126695 0)
tibia_r	0.87553671103478	(0 -0.132354 0)
talus_r	0.0236152855302705	(0 0 0)
calcn_r	0.295191069128381	(0.066774 0.0200322 0)
toes_r	0.0511507084585659	(0.0231038 0.00400644 -0.0116855)
femur_l	2.19655216831257	(0 -0.12326 0)
tibia_l	0.87553671103478	(0 -0.13221 0)
talus_l	0.0236152855302705	(0 0 0)
calcn_l	0.295191069128381	(0.0641224 0.0192367 0)
toes_l	0.0511507084585659	(0.0221864 0.00384735 0.0112215)
torso	7.9347359381709	(0.00237363 0.188667 0)
tot.	**17.60000000000000**	

Table 3: *Sirine and Lina scaled mass and CoM for each link, respectively.*

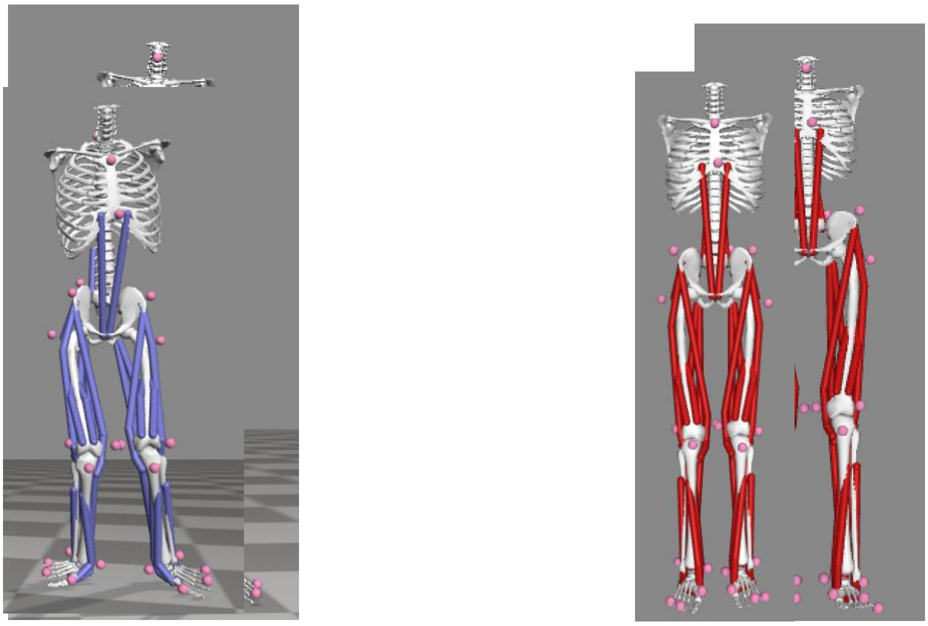


Figure 13: *Sirine (left) and Lina (right) models.* *) and Lina (right) static pose after scaling.*

So, we saved the two models as **`Sirine_scaled.osim`** and **`Lina_scaled.osim`**

5. KINEMATIC MODEL

In order to compute the kinematic model of the biped we have used a simplified model, in which we have considered the pelvis of the biped (1 joint), the left and right legs (3 joints each one).

In details, we have used the Denavit-Hartemberg table in which:

- $\boldsymbol{\alpha_i}$ represents the twist angle between joint axis, projected on a plane π orthogonal to the link axis, of the i-th link; in particular, it corresponds to the rotation that we have from z_i to z_{i+1} measured along x_{i+1};
- $\boldsymbol{a_i}$ represents the displacement between joint axes of the i-th joint; in particular, it corresponds to the distance that we have from z_i to z_{i+1} measured along x_{i+1};

- $\boldsymbol{d_i}$ represents the displacement between x_i to x_{i+1}, measured along z_i;
- $\boldsymbol{q_i}$ represents the angle between link axes (a variable if joint i-th is revolute); in particular, it corresponds to the rotation that we have from x_i to x_{i+1} measured along z_i.

DH table for the base (common)

α_i	a_i	d_i	q_i
$-\frac{\pi}{2}$	0	0	$q_{1,1}$
$-\frac{\pi}{2}$	0	0	$q_{1,2}-\frac{\pi}{2}$
0	d_1	0	$q_{1,3}$

DH table for the right leg / **DH table for the left leg**

α_i	a_i	d_i	q_i	α_i	a_i	d_i	q_i
$-\frac{\pi}{2}$	$d_{1,2}$	0	$-\frac{\pi}{2}$	$-\frac{\pi}{2}$	$-d_{1,2}$	0	$-\frac{\pi}{2}$
$-\frac{\pi}{2}$	0	0	$q_{2,1}-\frac{\pi}{2}$	$-\frac{\pi}{2}$	0	0	$q_{2,1}-\frac{\pi}{2}$
$-\frac{\pi}{2}$	0	0	$q_{2,2}-\frac{\pi}{2}$	$-\frac{\pi}{2}$	0	0	$q_{2,2}-\frac{\pi}{2}$
$-\frac{\pi}{2}$	a_2	0	$q_{2,3}$	$\frac{\pi}{2}$	a_2	0	$q_{2,3}$
$-\frac{\pi}{2}$	a_3	0	q_3	$-\frac{\pi}{2}$	a_3	0	q_3

Table 4: *DH table of the biped used to compute the complete kinematic model.*

Since we want to compute the end-effector point, from the DH-tables we derive the Denavit-Hartemberg matrix as follows:

$$p_n(q) = \left({}^0A_1(q_1) \left({}^1A_2(q_2) \dots \left({}^{n-1}A_n(q_n) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right) \right) \right) = \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix}$$

where n is the number of joints.

In order to plot the joint angles, we have done the **inverse kinematic** of the kinematic model, but instead of using the complete kinematic model, we have used a simplified version of this because it is very hard to find the joint angles by using the complete one.

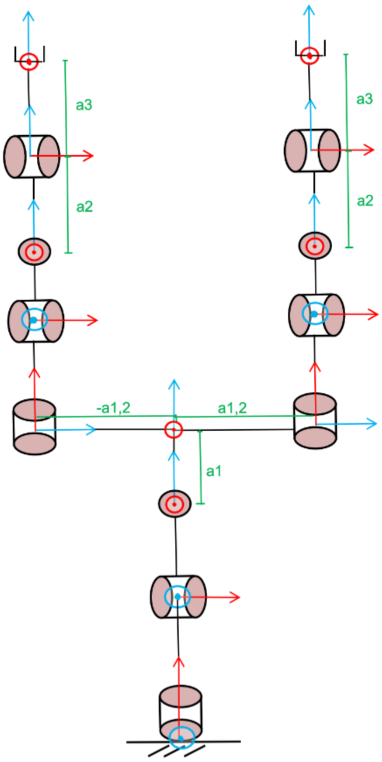


Figure 15: *Complex Biped model (7R).*

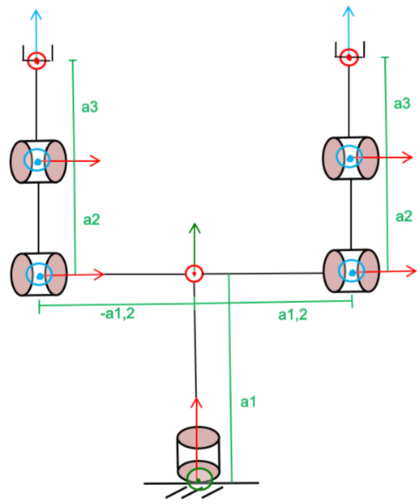


Figure 14: *Simplified Biped Model (3R)*

In **Table 6**, we report the simplified version that we have used in order to compute the inverse kinematic.

DH table for the base (common)

α_i	a_i	d_i	q_i
$\frac{\pi}{2}$	0	d_1	q_1

DH table for the right leg

α_i	a_i	d_i	q_i
0	0	$-d_2$	0
π	a_2	0	q_{2r}
0	a_3	0	q_{3r}

DH table for the left leg

α_i	a_i	d_i	q_i
0	0	d_2	0
π	a_2	0	q_{2l}
0	a_3	0	q_{3l}

Table 5: *Simplified DH table of the biped used to compute the inverse kinematic.*

The joints angles are defined as follow:

- q_1: is the pelvis angle;
- q_2: is the hip angle;
- q_3: is the knee angle.

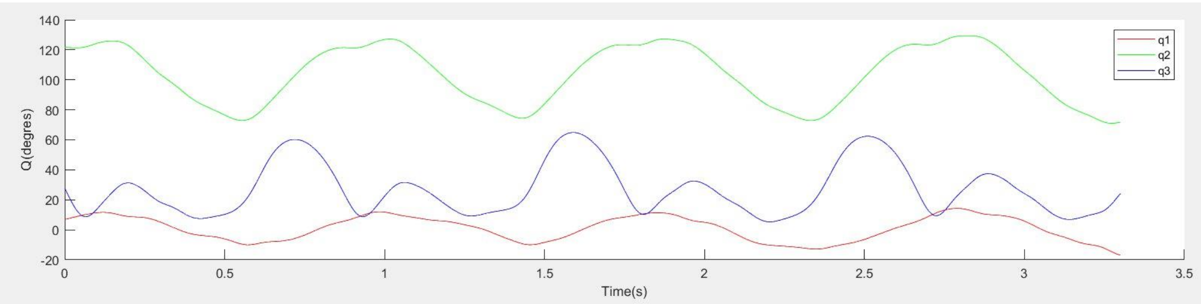


Figura 17: *Joint angles of Sirine's right leg of Marche 2.*

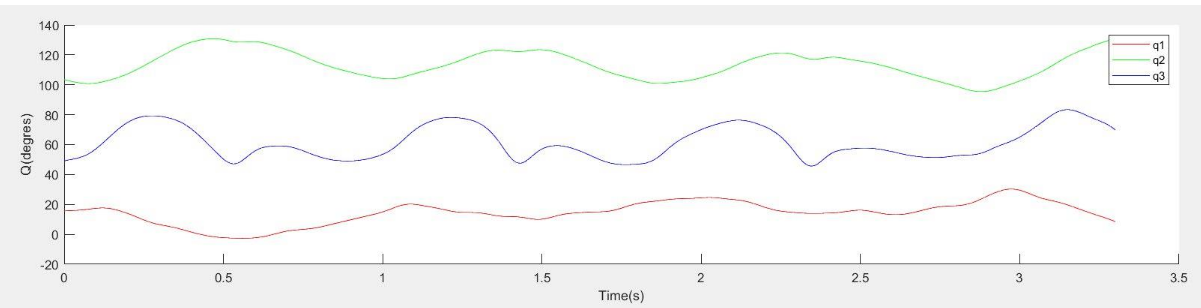


Figura 18: *Joint angles of Lina's left leg of Marche 2.*

6. DYNAMIC MODEL

In order to compute the dynamic model of the biped we have followed the moving frames approach (recursive algorithm).

In particular, the equation of the dynamic model that we want to obtain is the following:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u}$$

Where:
$\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix.
$\ddot{\boldsymbol{q}}$ is the acceleration.
$\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are the Coriolis and Centrifugal terms.
$\boldsymbol{g}(\boldsymbol{q})$ are the gravity terms.

We use the same approach for both Lina and Sirine scaled models.

6.1. INERTIA MATRIX

Given:

1. DH table of the biped model;
2. I_{ci}: inertia matrix of each link;
3. m_i: mass of each link;
4. ${}^{1}r_{1,c1}$, ${}^{2}r_{2,c2}$, ${}^{3}r_{3,c3}$: CoM of each link.

Goal: find the Inertia matrix $M(q)$. The inertia matrix depends only on the configuration.

Solution:

1. Initalization:
$${}^{0}\varpi_0 = 0 \qquad {}^{0}v_0 = 0 \qquad i = 0$$
2. Write all the rotation matrix that we have (matlab: DH_table file)
$${}^{0}R_1 \quad .... \quad {}^{n-1}R_n \quad n = nun\ joints$$
3. Assign σ_i , i-th joint.
$$\sigma_i = \begin{cases} 0 & revolute \\ 1 & prismatic \end{cases}$$
4. Follow these formulae for EACH link:
$${}^{i}\boldsymbol{\varpi_i} = {}^{i-1}R_i^{\ T}\left[{}^{i-1}\varpi_{i-1} + (1-\sigma_i)\,\dot{q}_i\, z\right] \quad z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$
$${}^{i}\boldsymbol{v_i} = {}^{i-1}R_i^{\ T}\left[{}^{i-1}v_{i-1} + \sigma_i \dot{q}_i\, z + {}^{i-1}\varpi_i \times {}^{i-1}r_{i-1,i}\right]$$
$$= {}^{i-1}R_i^{\ T}\,{}^{i-1}v_{i-1} + {}^{i-1}R_i^{\ T}\sigma_i\dot{q}_i\, z + {}^{i-1}R_i^{\ T}\,{}^{i-1}\varpi_i \times {}^{i-1}R_i^{\ T}\,{}^{i-1}r_{i-1,i}$$

${}^{i-1}r_{i-1,i}$ is which you can find in the ${}^{i-1}A_i$ (Denavit-Hartenberg matrix, that is a roto-translation matrix) [last column, first 3 elements, because we want the translation]

$${}^{i}\boldsymbol{v_i} = {}^{i-1}R_i^{\ T}\,{}^{i-1}v_{i-1} + {}^{i-1}R_i^{\ T}\sigma_i\dot{q}_i\, z + {}^{i}\varpi_i \times {}^{i}r_{i-1,i}$$
$${}^{i}\boldsymbol{v_{ci}} = {}^{i}v_i + {}^{i}\varpi_i \times {}^{i}r_{i,ci}$$

5. Compute the kinetic energy of each link:

$$T_i = \frac{1}{2} m_i \left\| {}^i v_{ci} \right\|^2 + \frac{1}{2} {}^i\varpi_i{}^T I_{ci} {}^i\varpi_i$$

Then, we compute the total kinetic energy for the whole model:

$$T = T_1 + \cdots + T_n$$

6. With matlab I find $M(q)$, given all the T_i
   $M(q)$ is always a symmetric matrix.
   $M(q)$ is $n \times n$, where n is the number of joints.

6.2. CORIOLIS AND CENTRIFUGAL TERMS

Once we have the Inertia matrix $M(q)$, in order to compute the Coriolis and Centrifugal terms we have identify each column of the inertia matrix with a different name as follows:

$$M(q) = [M_1 \quad M_2 \quad M_3 \dots \; M_n]$$

Where n identifies the number of joints.

Then, we apply the following formulas:

$$C_k(q) = \frac{1}{2}\left(\frac{\partial M_i}{\partial q} + \left(\frac{\partial M_i}{\partial q}\right)^T - \frac{\partial M}{\partial q_i}\right) \quad k = 1, \dots, n$$

$$c_k(q, \dot{q}) = \dot{q}^T \, C_k(q) \, \dot{q} \quad k = 1, \dots, n$$

We can rewrite $c_k(q, \dot{q})$ in a matrix form as follows:

$$c_k(q, \dot{q}) = \begin{pmatrix} c_i(q, \dot{q}) \\ \dots \\ c_n(q, \dot{q}) \end{pmatrix}$$

We have centrifugal term when we have a joint velocity square, otherwise we have a Coriolis term. In fact, the Coriolis and centrifugal terms are in function of both configuration and velocity. The Coriolis and Centrifugal terms depend on the configuration and on the velocity.

6.3. POTENTIAL ENERGY AND GRAVITY TERMS

Since our model is in a vertical plane, we have to compute the gravity terms.
We have to follow 2 steps to obtain the gravity vector:

1. We have to compute the potential energy for each link:

   $$U_{gi} = -m_i \, g_0 \, r_{c,i} \quad i = 1, \dots, n$$

   Then, we compute the total potential energy for the whole model:

   $$U_g = U_{g1} + \cdots + U_{gn}$$

   Where:
   m_i is the mass of the joint.
   g_0 is the gravity vector containing g_0 ($g_0 = 9.81 \, m/s^2$) in the gravity position and all

other values are equal to zero.
$r_{c,i}$ is the center of mass (CoM) offset w.r.t. RF0.

2. Now, we can compute the gravity vector:

$$g(q) = \left(\frac{\partial U_g}{\partial q_1} \ \frac{\partial U_g}{\partial q_2} \ \dots \ \frac{\partial U_g}{\partial q_n} \right)^T$$

The gravity vector depends only on the configuration of the robot.

7. EQUILIBRIUM ANALYSIS

7.1 TIME SEGMENTATION

In order to identify and better understand the gait deviations of children with CP we divided the walking cycle in two different phases: double stance (DS) and single stance (SS). These two phases will be further divided for the left leg (DSl followed by a SSl) and for the right one (DSr followed by a SSr).

The computation of DS and SS intervals are done using the trajectory of toe marker (RTT2, LTT2) on the y-axis as reference. The maximum height of the toe marker establishes the start of the DS phase, and its end is defined by the start of the foot's movement that it's found behind.

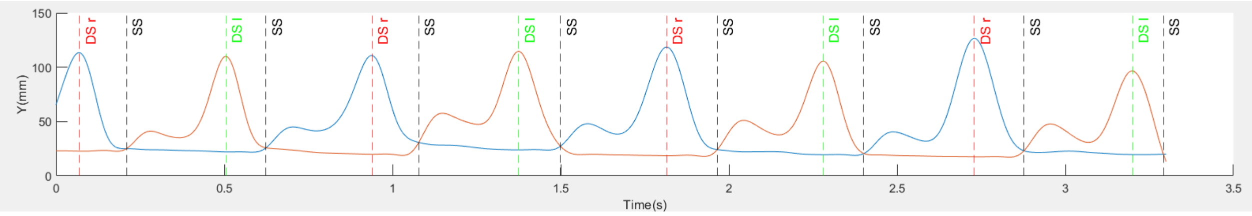


Figure 19: *plot of the time segmentation and trajectory of toes markers (RTT2, LTT2).*
(Sirine, `FILTERED_marche 2.trc`*)*

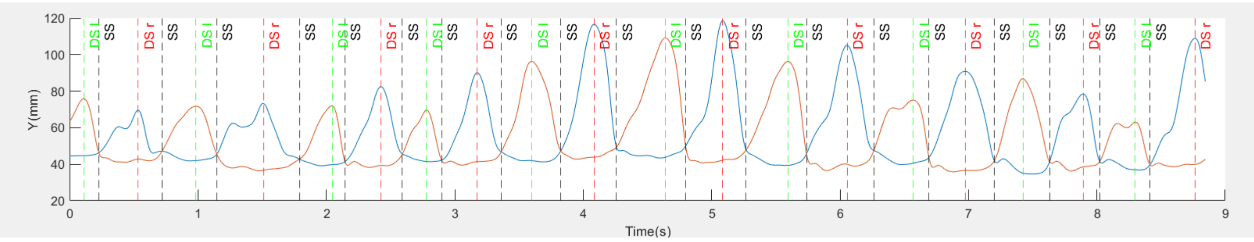


Figure 20: *plot of the time segmentation and trajectory of toes markers (RTT2, LTT2).*
(Lina, `FILTERED_marche 5.trc`*)*

As observed in [4] Lina, in order to decrease her instability, walks with a larger DS phase compared to Sirine, while reducing the SS phase as well. The DSl phase is clearly shorter than the right one for both twins. The left swing phase (SSr) of Lina lasts less than the right one because the left side is more affected by the spasticity as confirmed by medical observations. Notice also the higher standard deviations of Lina compared to the ones of Sirine which is due to the larger gait variability of Lina.

	Sirine	Lina
DS (sec)	**0,13±0,019**	**0,17±0,047**
DSl (sec)	0,11±0,015	0,15±0,042
DSr (sec)	0,14±0,006	0,19±0,042
SS (sec)	**0,31±0,012**	**0,29±0,048**
SSl (sec)	0,32±0,007	0,30±0,042
SSr (sec)	0,31±0,013	0,27±0,058

Table 6: *Time segmentation for Sirine and Lina. (Means and standard deviation values are computed on a single marche,* `FILTERED_marche 5.trc` *for Lina and* `FILTERED_marche 2.trc` *for Sirine).*

7.2 COMPUTATION OF $d_{CoM-\Delta}$

After having computed the CoM of the total body for Sirine and Lina, our work is to compute a descriptor that is computed as the distance between the center of mass of the body and the minimal moment axis [3].

We start our work by filtering the ground reaction forces and moments that are in the .mot file (Force Plate Data) by applying a 4th order, zero phase-shift, low-pass Butterworth filter with the cutoff frequency equal to 10 Hz (as we did for the .trc file).

Then, we have computed the moment field expressed at the CoM point. This is defined as:

$$M_{CoM} = M_A + F \times P_{(A,CoM)}$$

where $P_{(A,CoM)}$ is the position of CoM with respect to a point A.

In conclusion, we have computed:

$$d_{CoM-\Delta} = \frac{F \times M_{CoM}}{||F||^2}$$

That represent the distance between the center of mass of the body and the minimal moment axis.

Now, we show the result that we have obtained for Sirine and Lina.

<u>Note</u>: this is only one example done using one marche from Sirine and Lina of the given data. But we can do this analysis for all the Marches that we have without problems.

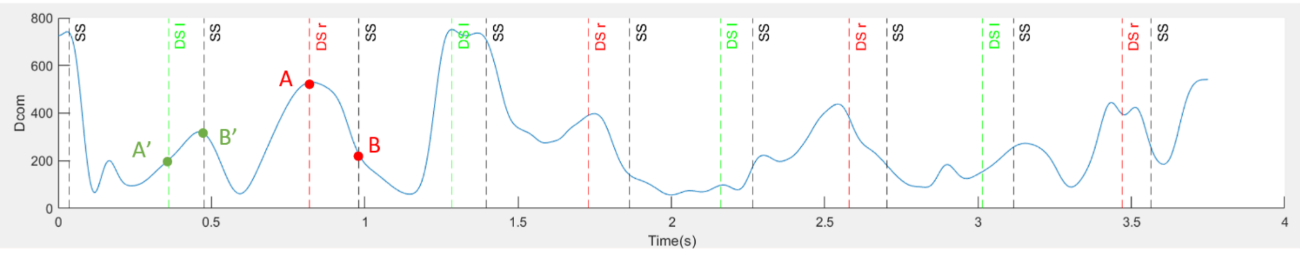


Figure 21: *plot of the distance between minimal moment axis and center of mass (CoM)* $d_{CoM-\Delta}$*, relative to the gait cycle. This plot is referred to the Sirine model (healthy subject).*

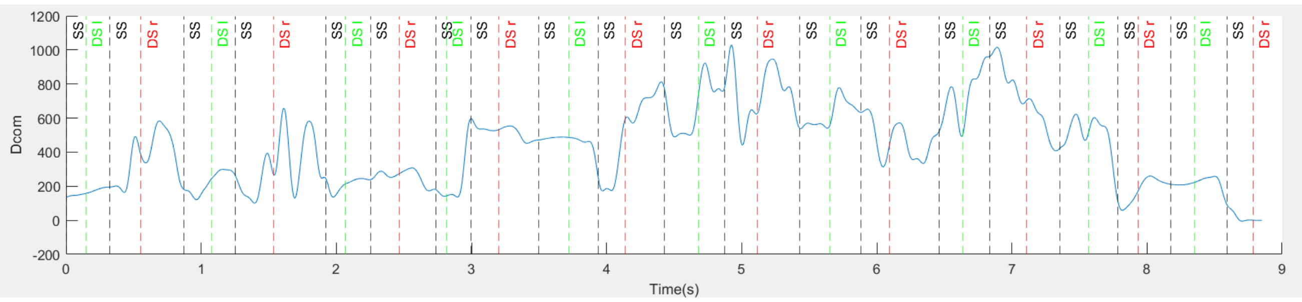


Figure 22: *plot of the distance between minimal moment axis and center of mass (CoM)* $d_{CoM-\Delta}$ *, relative to the gait cycle. The plot is referred to the lina model (subject that has a spastic CP).*

The results obtained for Sirine are only half compatible with those found in [3]. Only if the heel strike is made with the right foot, the outcome's evolution is similar to the healthy subject's gait: MMA is farthest away behind the subject's CoM (A). Afterwards we can notice a drop in the distance. Thus, the MMA moves toward the CoM and tries to cross it. In the Toe Off (B) instant the MMA is now in front of the body.
If the heel strike is made with the left foot (A') the results don't follow this trend.

The results obtained for Lina show variation of $d_{CoM-\Delta}$ very different from the physiological one found in [3] and from Sirine's. Because it depends on the whole-body dynamics, the variation profile of $d_{CoM-\Delta}$ at the scale of the walking cycle is representative of the rhythm of locomotion. Her pathological condition causes his posture to be irregular during the gait: torso bent over, knee flexed and feet spread apart. So, the distance's profile has a larger variability.

8. REFERENCES

[1] OpenSim Documentation
https://simtk-confluence.stanford.edu/display/OpenSim/C3D+%28.c3d%29+Files

[2] R. Dumas, L. Chèze, J.-P. Verriest: Adjustments to McConville et al. and Young et al. body segment inertial parameters

[3] Nahime Al Abiad, Hélène Pillet and Bruno Watier: A Mechanical Descriptor of Instability in Human Locmotion: Experimental Findings in Control Subjects and People with Transfemoral Amputation.

[4] Marco Marchitto (2019-2020). Motion analysis for children with cerebral palsy: a comparative study of twins.