SAPIENZA
UNIVERSITÀ DI ROMA

Adult-Child Musculoskeletal Model and Motion Analysis

Final Project in *Medical Robotics*

September 2021

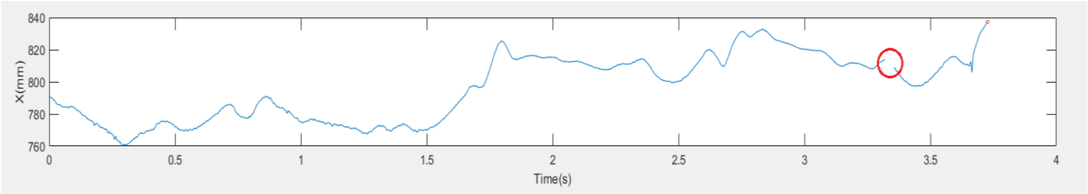
Catia Carocci, Alessandro Lambertini, Denise Landini

Project goal and Overview:

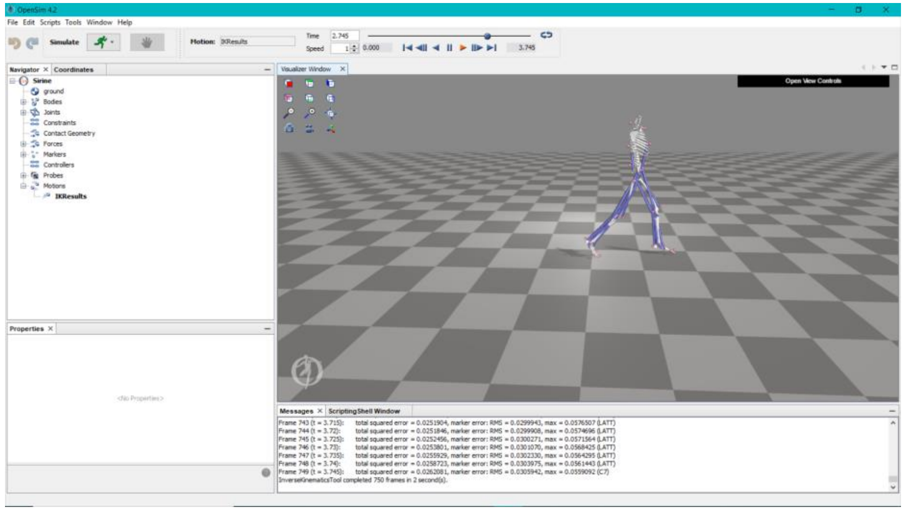
SAPIENZA
UNIVERSITÀ DI ROMA

FIRST PART:

1) **Processing** of the raw data from MOCAP system of two twins relating to 18 walking trials for Sirine (healthy subject) and 6 walking trials for Lina (affected by spastic CP)
2) **Visualization** of the manipulated data with an OpenSim model for Sirine and Lina.



From raw data to OpenSim



Project goal and Overview:

SAPIENZA UNIVERSITÀ DI ROMA

SECOND PART:

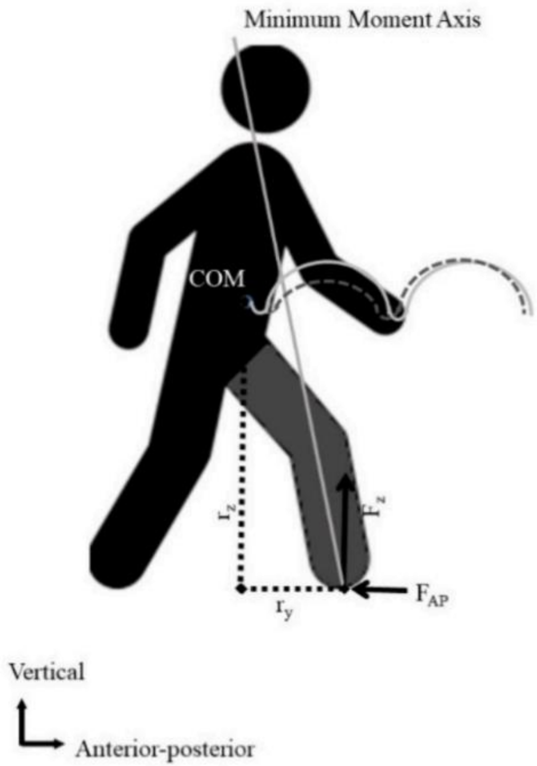
Use the processed data in order to do the **equilibrium analysis** of the twins:

1) Scaling of the muscoloskeletals models;

2) Kinematic and Dynamic analysis;

3) Computation of the distance between Minimum Moment Axis (Δ) and the Center of Mass (CoM) [1]:

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



Input Data

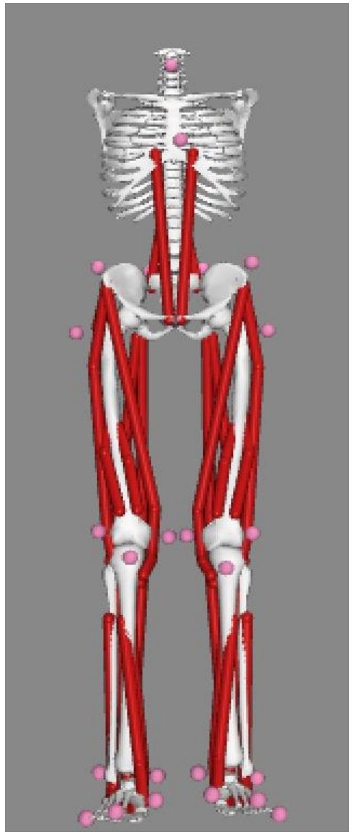
Adult-Child Musculoskeletal Model and Motion Analysis

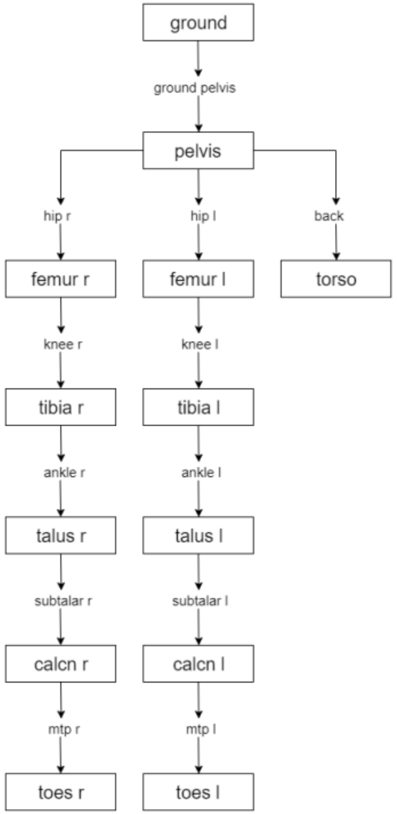
Input Data – Sirine

- 7 years old, 1.20 m height, 20 kg weight;
- OpenSim simplified model (upper body approximated only to the torso);
- 18 walking trials stored in .c3d files

↳ Coverted using Matlab through **c3dExport.m** function [2] into:

- *.mot* files (Force Plate Data)
- *.trc* files (Marker Data)

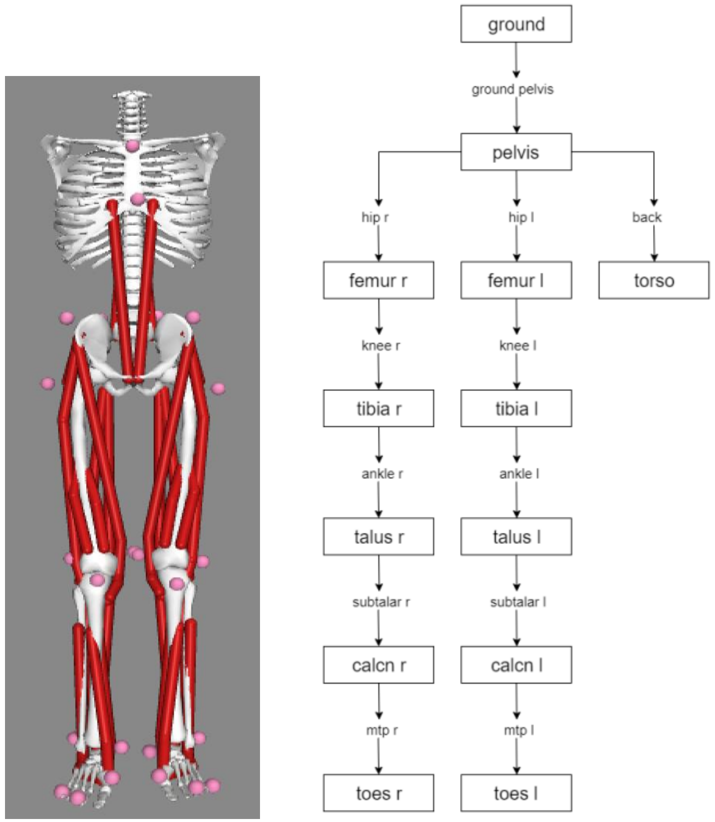




Input Data – Lina

- 7 years old, 1.05 m height, 17.6 kg weight;
- OpenSim simplified model (upper body approximated only to the torso);
- 6 walking trials stored in .c3d files

  ↳ Coverted using Matlab through **c3dExport.m** function [2] into:
  - *.mot* files (Force Plate Data)
  - *.trc* files (Marker Data)



Input Data: Marker data – Sirine/Lina

SAPIENZA UNIVERSITÀ DI ROMA

.trc files have information about**:**

- Sampling frequency [Hz] (*DataRate*);
- Total number of frames (*NumFrames*);
- Total number of markers (*NumMarkers*);
- Units of measurement of recorded values (*Units*);
- (x, y, z) coordinates for markers, recorded in camera frame.

```
DataRate	CameraRate	NumFrames	NumMarkers	Units	OrigDataRate	OrigDataStartFrame	OrigNumFrames
200.000000	200.000000	677	48	mm	200.000000	0	677
Frame#	Time	LASIS			RASIS			LPSIS			RPSIS			RGT			RMFE			RLFE
		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

1	0	807.4160766601563	728.630187988281	2939.827392578125	1015.848510742188	733.7728881835935	2959.91
2	0.005	807.1309814453125	727.8632812499998	2932.158447265625	1015.347351074219	732.2096557617185	295
3	0.01	806.552490234375	727.225891113281	2924.619873046875	1017.967895507813	730.3627319335935	294
4	0.015	806.8902587890625	726.4095458984373	2916.244140625	1016.125305175781	729.5965576171873	2936.69
5	0.02	805.5819702148438	726.4669189453123	2908.81298828125	1017.543395996094	728.0111083984373	292
6	0.025	805.8358764648438	726.1632080078123	2899.45849609375	1017.225830078125	726.672668457031	291
7	0.03	805.3377685546875	726.0731811523435	2891.212890625	1015.249572753906	726.2040405273435	2911.30
8	0.035	805.3782958984375	726.2785034179685	2882.498291015625	1017.725646972656	724.9804077148435	290
9	0.04	805.6094970703125	726.547180175781	2873.031494140625	1015.696228027344	724.2023925781248	289
```

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

Input Data: Marker data – Sirine/Lina

SAPIENZA UNIVERSITÀ DI ROMA

`loadDataFromTRC.m` function was used to read the data from the file .trc:

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 loadDataFromTRC.m of marche 18.trc of Sirine

The data matrix (*m_cols*) first two columns represent the frames and the time instants in which they were acquires and the remaining ones are precisely the x, y and z components of each marker.

Input Data: Force Plate Data– Sirine/Lina

SAPIENZA
UNIVERSITÀ DI ROMA

.mot files 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 .

```
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)	-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 by using Notepad++

Input Data: Force Plate Data– Sirine/Lina

`loadDataFromMOT.m` function was used to read the data from the file .mot:

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.

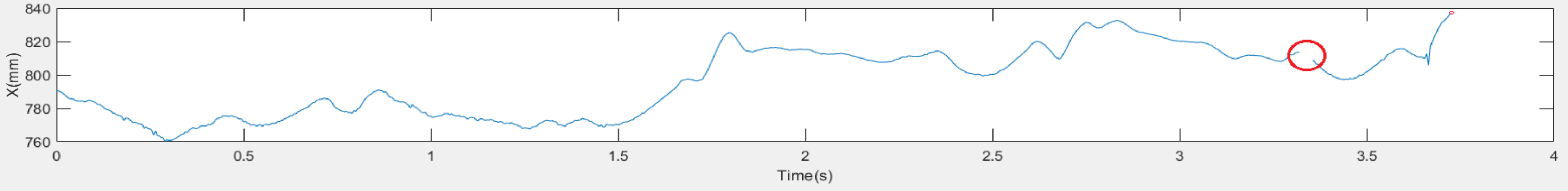
Gap filling analysis

Adult-Child Musculoskeletal Model and Motion Analysis

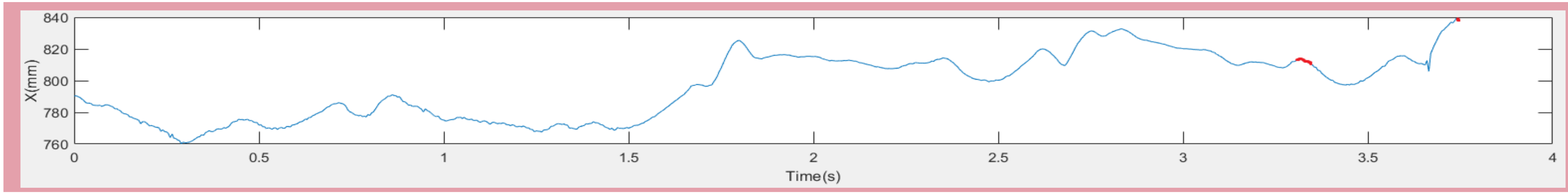
Gap filling: introduction – Sirine/Lina

- Raw data contain NaN values in 3D markers
- **Gap filling** allows us to eliminate this NaN values
- Gap filling is done **before filtering** because if we filter the entire signal, we will reduce the distortion of the filtered signal.

Trajectory of x-component of LATT marker in marche 12 of Sirine with Gap



Trajectory of x-component of LATT marker in marche 12 of Sirine without Gap



Gap filling: introduction – Sirine/Lina

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 trajectories of Sirine

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]

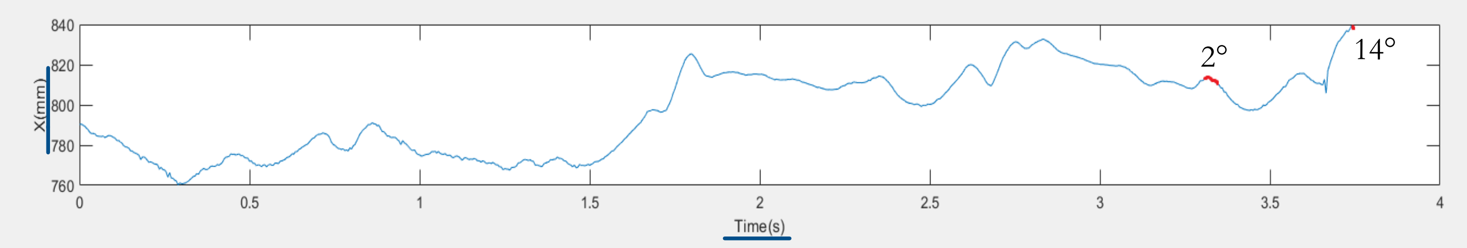
***Table 2**:* summary table for the gaps present in the trajectories of Lina

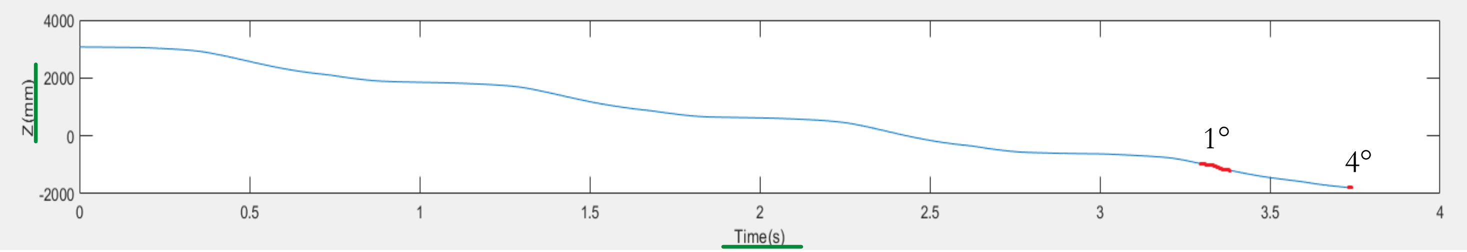
Gap filling: implementation – Sirine/Lina

- **Implementation:**
  - **GapFiller.m:** file that include the identification and the elimination of the gap.

- **Solution:** use a **linear regression method**.
  - For each interval, we have used different degrees of the polynomial according to the best estimation that it produces.
  - Example:

```
MC(660:680, 42) = gapfilling(MC(660:680, :), 42, 2);
MC(:, 42) = gapfilling(MC, 42, 14);
MC(660:680, 44) = gapfilling(MC(660:680, :), 44, 1);
MC(:, 44) = gapfilling(MC, 44, 4);
```





Filtering analysis

Adult-Child Musculoskeletal Model and Motion Analysis

Filtering analysis – Sirine/Lina

Raw data contains high frequency noise, **filtering** allow us to remove it.

- **Implementation:**
  - **dataProcessing.m:** script that also design filter, filter the data and then put them in a new *.trc* file.

- **Solution:** use a **low-pass filter**.
  - **Butterworth filter of order 4**
    - NO ripples in the passband
    - Rolls off towards zero in the stopband
  - Fixed cutoff frequency: 10 Hz
  - Sampling frequency:
    - **Sirine**: 200 Hz
    - **Lina**: 2000 Hz

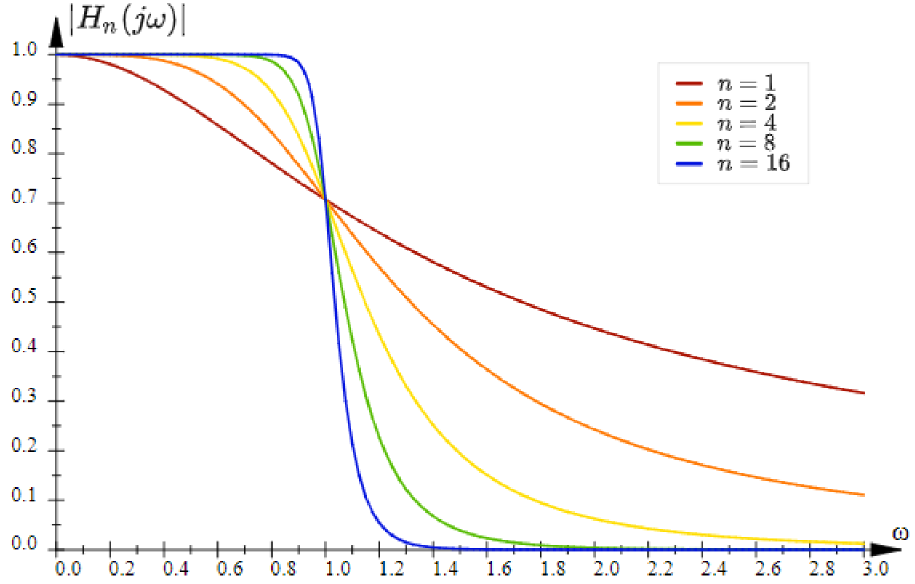
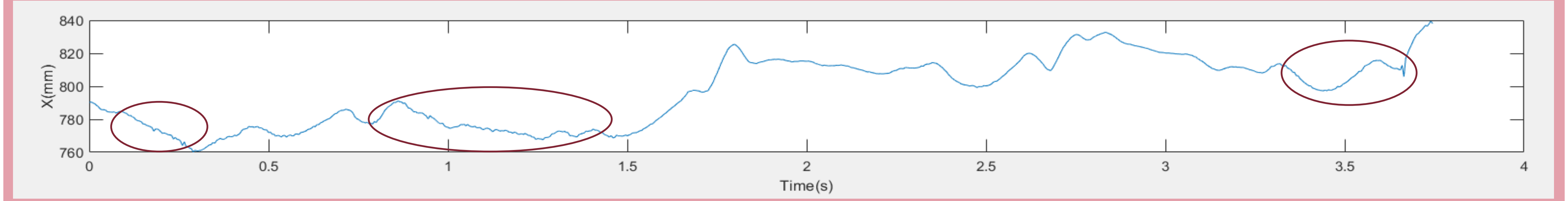


Figure 5: Frequency response of n-order Butterworth low-pass filter

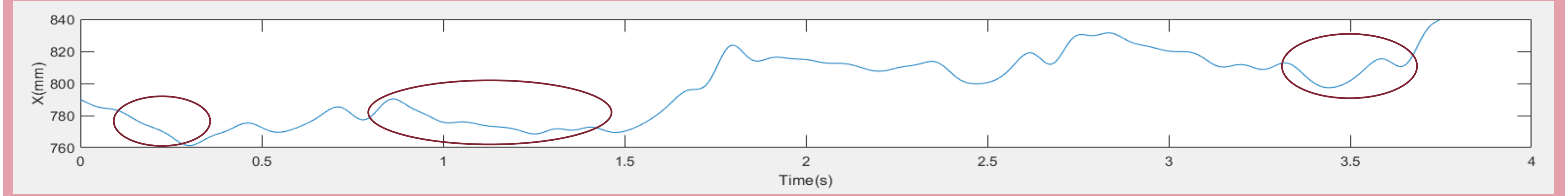
Filtering analysis: example – Sirine/Lina

SAPIENZA
UNIVERSITÀ DI ROMA

Trajectory of x-component of LATT marker in marche 12 of Sirine without filtering



Trajectory of x-component of LATT marker in marche 12 of Sirine with filtering



Scaling phase

Adult-Child Musculoskeletal Model and Motion Analysis

Scaling phase – Sirine/Lina

Sirine

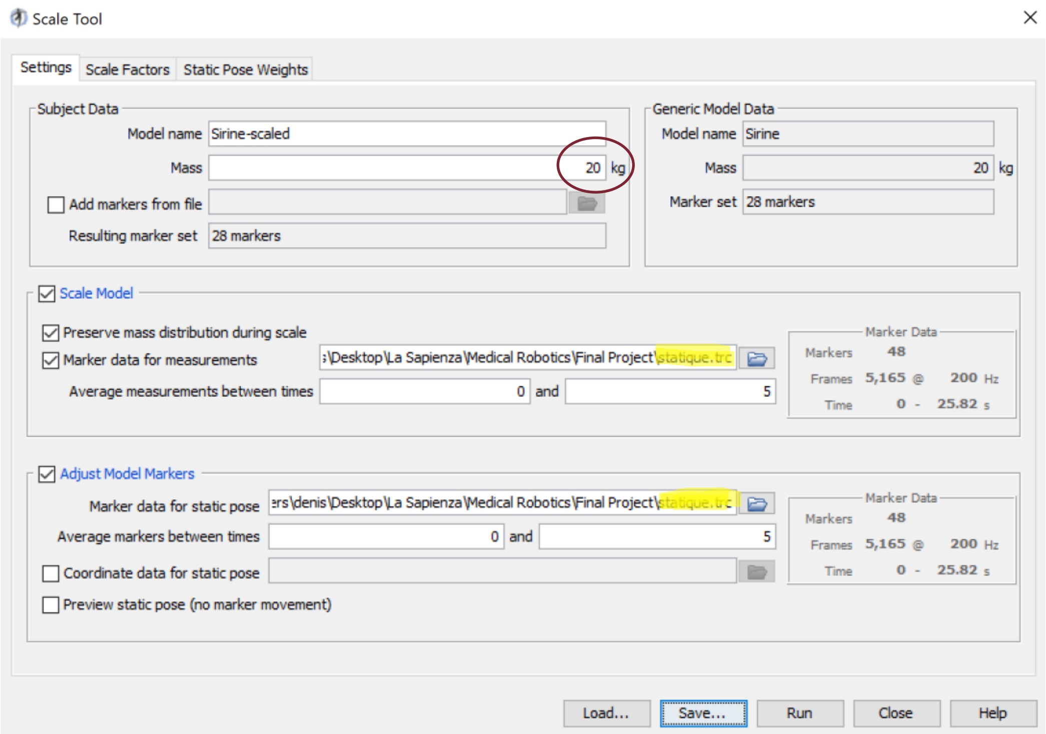


Figure 6: Setting of Scale Tool for Sirine

Lina

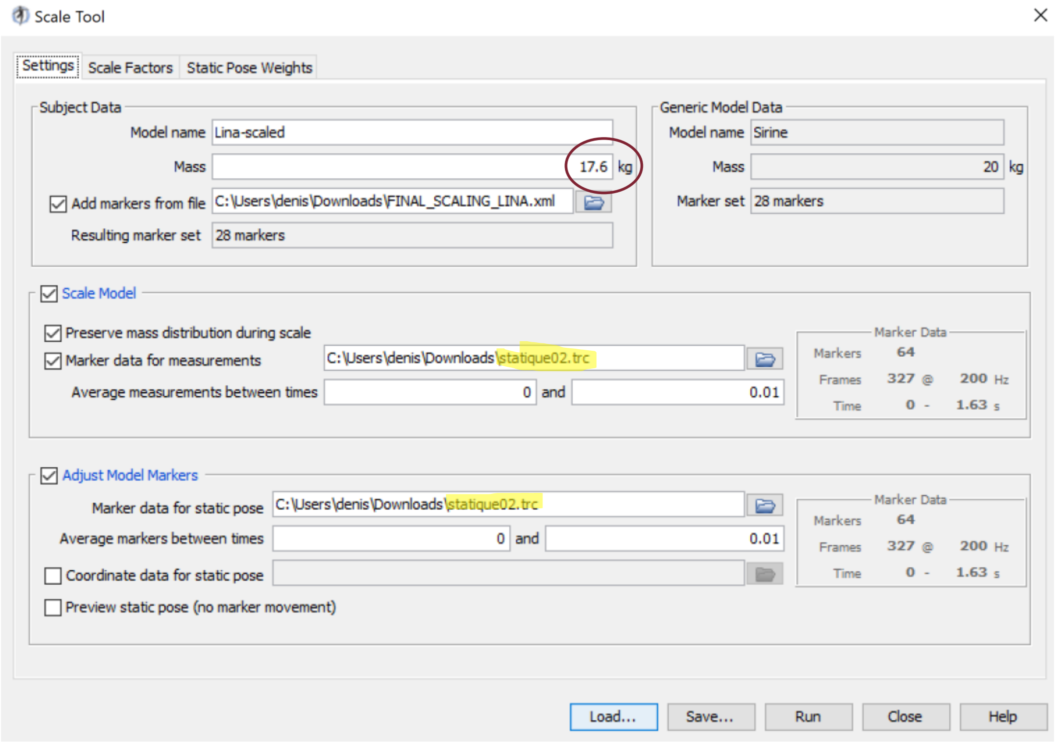


Figure 7: Setting of Scale Tool for Lina

Scaling phase – Sirine/Lina

Why?
In order to adapt the location of the virtual markers on the body model to experimental ones.

We scale the model by using the **Scale Tool** in OpenSim

- For each segment of the body, we choose the final markers pairs.

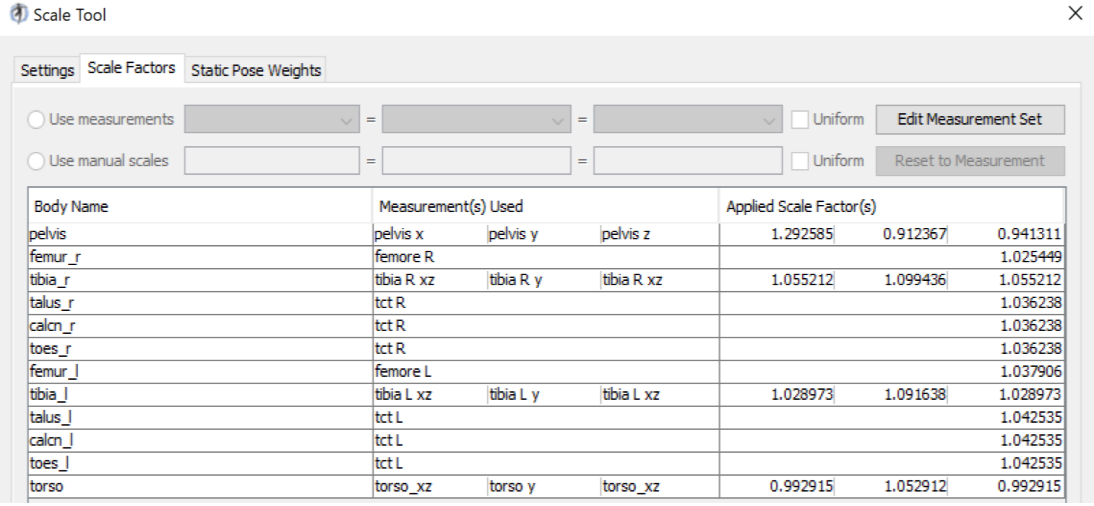
Measurement Set

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 8: Measurment Set for both Sirine and Lina

Scaling phase – Sirine

With these final markers pairs, we are able to obtain the **scaling factors.**


Scale Tool

Settings | Scale Factors | Static Pose Weights

Use measurements = = Uniform Edit Measurement Set

Use manual scales = = Uniform Reset to Measurement

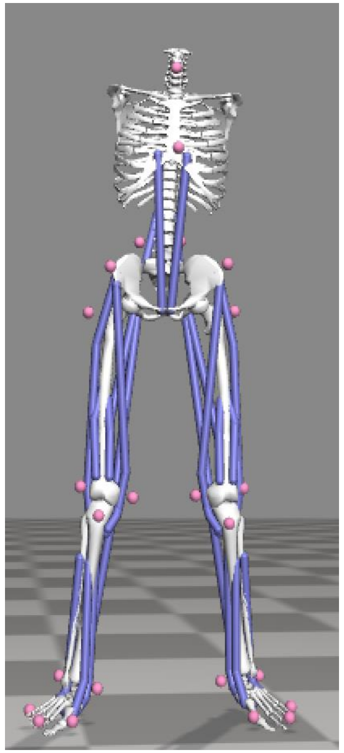
Body Name	Measurement(s) Used			Applied Scale Factor(s)		
pelvis	pelvis x	pelvis y	pelvis z	1.292585	0.912367	0.941311
femur_r	femore R					1.025449
tibia_r	tibia R xz	tibia R y	tibia R xz	1.055212	1.099436	1.055212
talus_r	tct R					1.036238
calcn_r	tct R					1.036238
toes_r	tct R					1.036238
femur_l	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 9: Scaling factors for Sirine

	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**	

Figure 10: Scales masses and CoM for Sirine

With scaling tool, OpenSim scale also the masses, CoM, forces and inertia of each link.

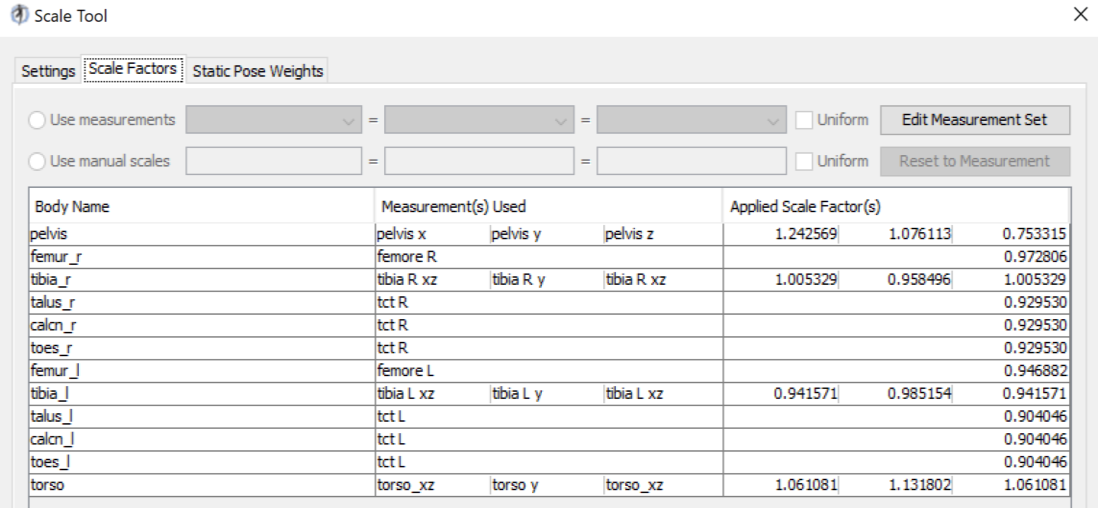


Scaling phase – Lina

With these final markers pairs, we are able to obtain the **scaling factors.**

	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**	

Figure 12: Scales masses and CoM for Lina


Scale Tool

Settings | Scale Factors | Static Pose Weights

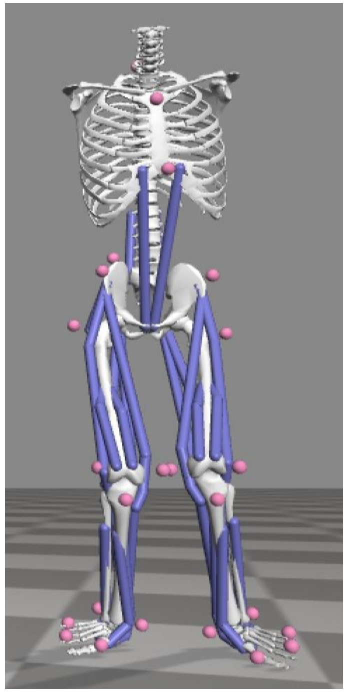
Use measurements = = Uniform Edit Measurement Set

Use manual scales = = Uniform Reset to Measurement

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 11: Scaling factors for Sirine

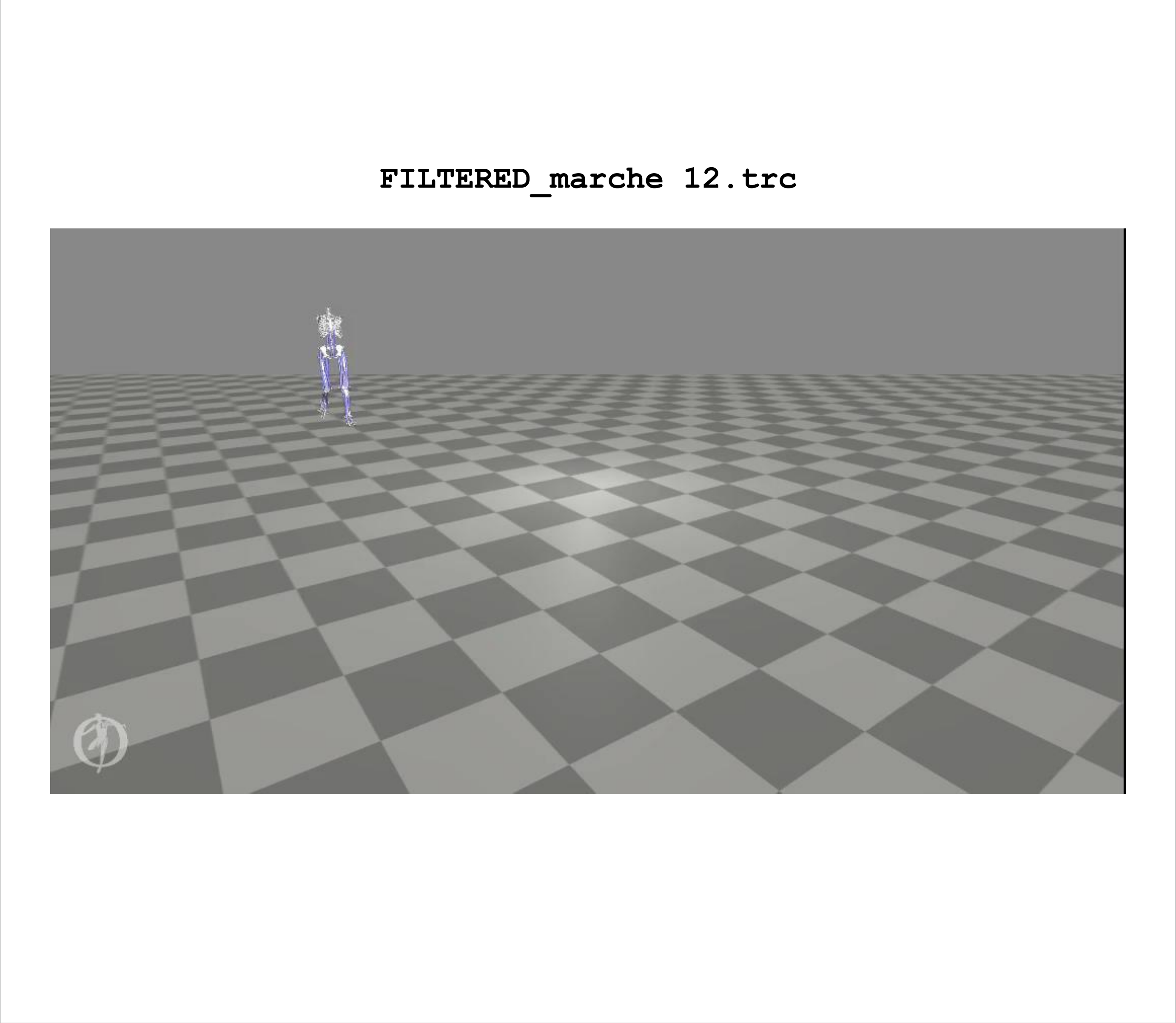
With scaling tool, OpenSim scale also the masses, CoM, forces and inertia of each link.



Visualization in OpenSim - Sirine

- Filled and filtered maker data can be associated to the OpenSim simplified and scaled model.

- The inverse Kinematics tool is used to display trials.

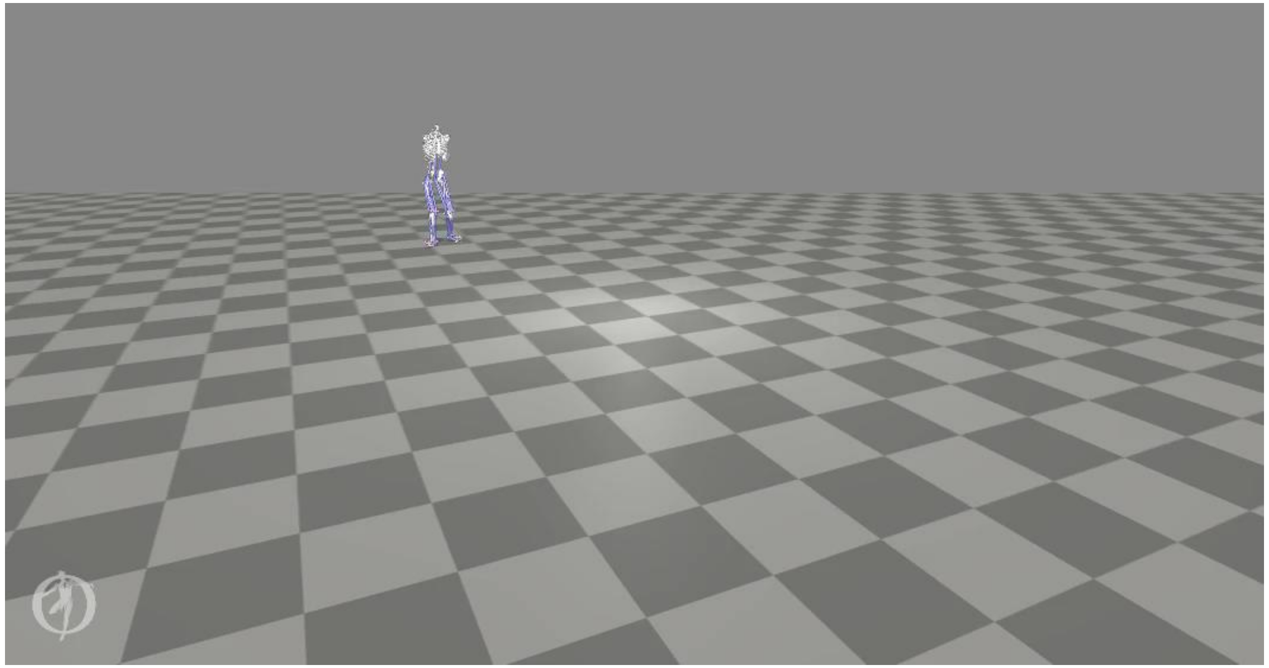


Visualization in OpenSim - Lina

- Filled and filtered maker data can be associated to the OpenSim simplified and scaled model.

- The inverse Kinematics tool is used to display trials.

FILTERED_marche 2.trc



Kinematic analysis

Adult-Child Musculoskeletal Model and Motion Analysis

SAPIENZA
UNIVERSITÀ DI ROMA

Kinematic Model – Sirine/Lina

In order to compute the **direct kinematic** to obtain the kinematic model, we have computed the DH tables of the biped model.

Implementation: `KD_model_complex.m`

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	a_1	0	$q_{1,3}$

DH table for the right leg

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

DH table for the left leg

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

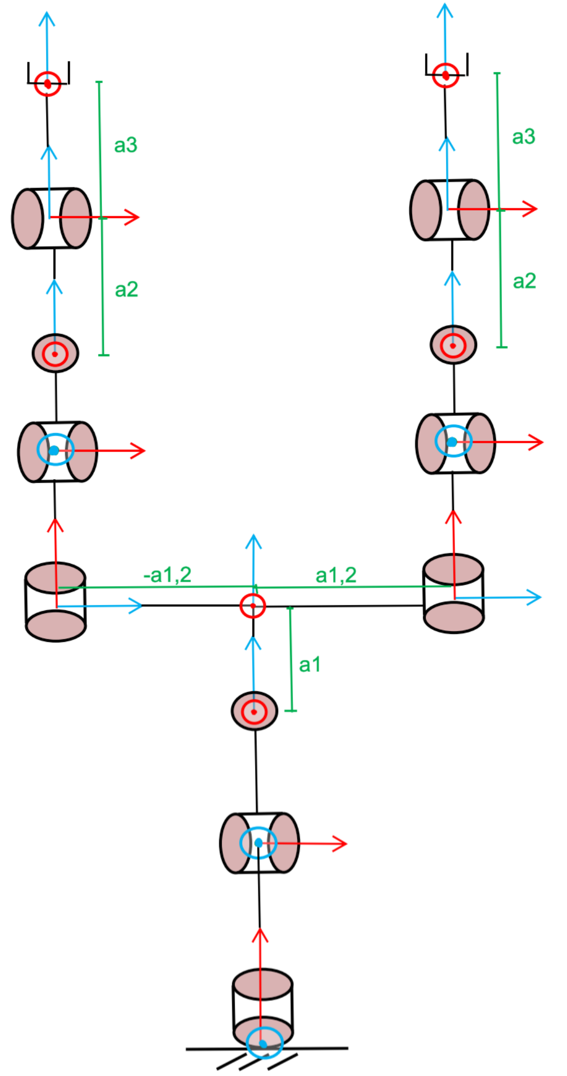


Figure 13: Complex Biped model (7R)

Kinematic Simplified Model – Sirine/Lina

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}

In order to compute the **inverse kinematic** to obtain the plot of the joint angles we have used a simplified version of the kinematic model.

Implementation: `KD_model.m`

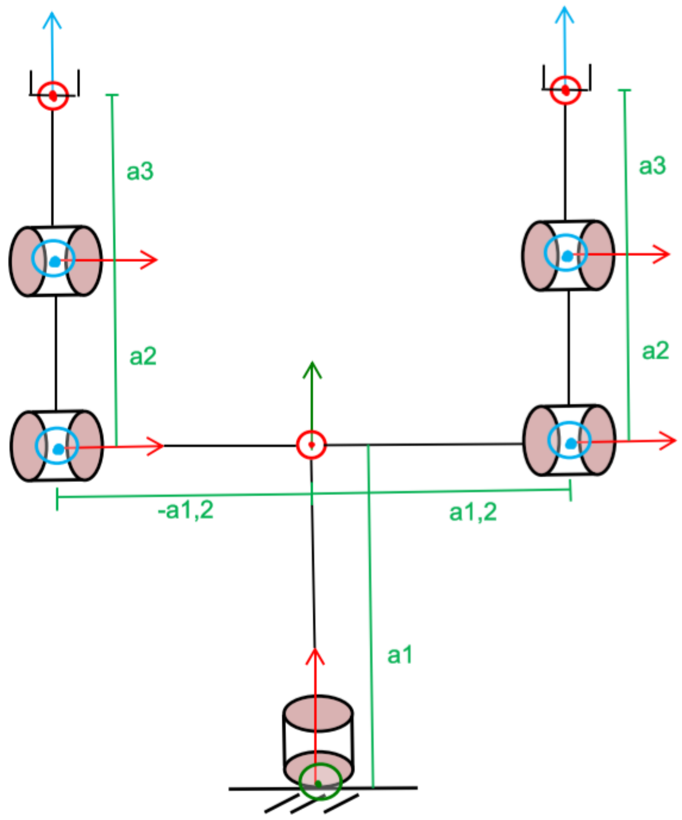


Figure 14: Simplified Biped model (3R)

Results of Inverse kinematic – Sirine/Lina

These are the **results** that we have obtained by computing the **inverse kinematic** using the simplified kinematic model. The joints angles are:

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

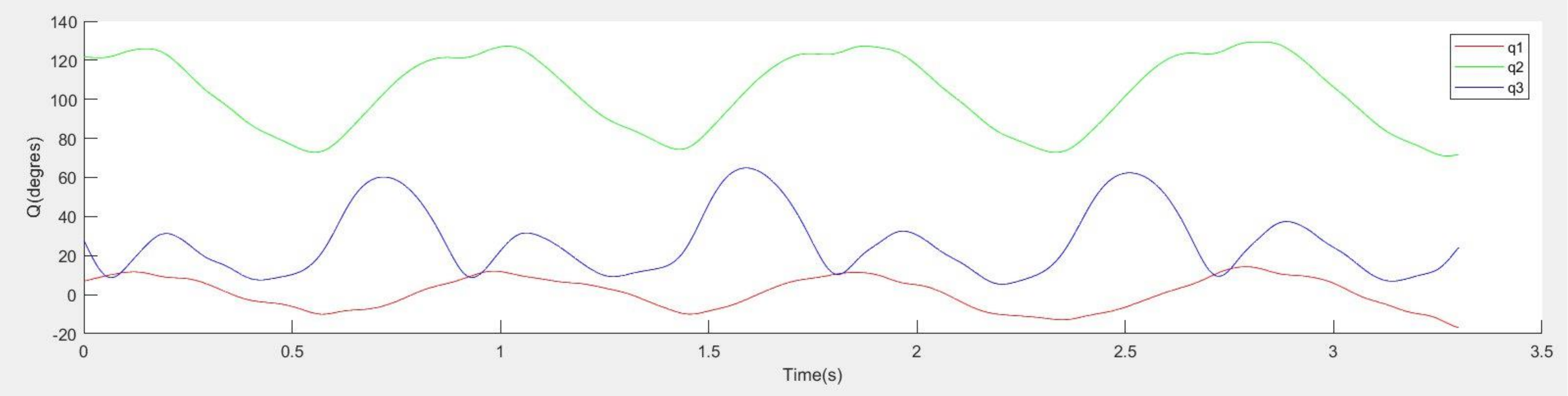


Figure 15: Joint angles of Sirine's right leg of Marche 2

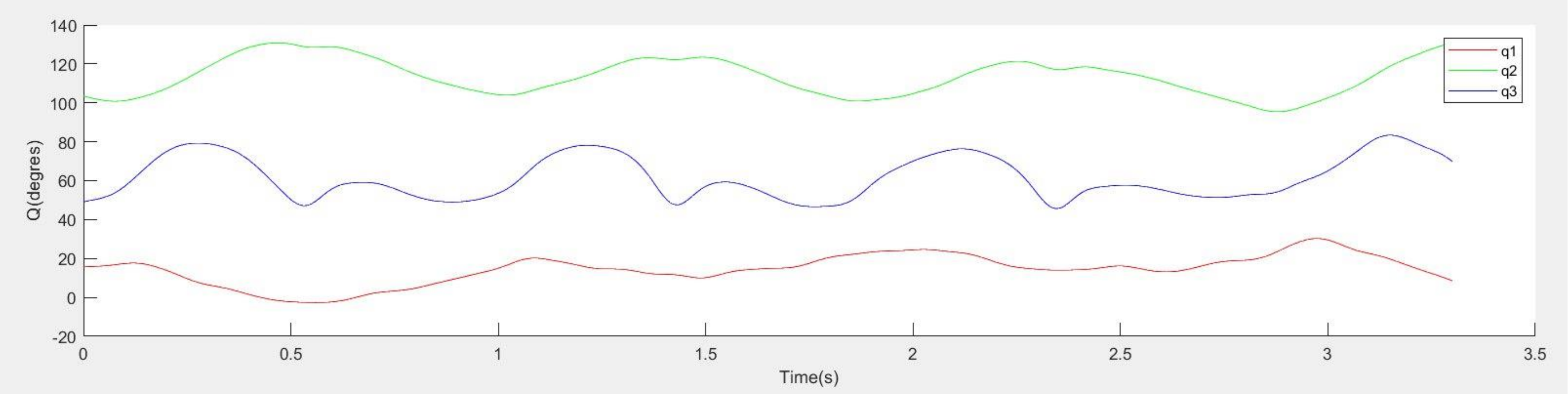


Figure 16: Joint angles of Lina's left leg of Marche 2

Dynamic analysis

Adult-Child Musculoskeletal Model and Motion Analysis

Sapienza Università di Roma

Dynamic Model – Sirine/Lina

In order to do the **dynamic model,** we have computed:

- **Inertia Matrix $\boldsymbol{M}(\boldsymbol{q})$:** that depends only on the **configuration** of the humanoid.
- **Coriolis and Centrifugal Terms $\mathbf{c}(\boldsymbol{q},\dot{\boldsymbol{q}})$:** that depends on the **configuration** and on the **velocity** of the humanoid.
- **Gravity $\boldsymbol{g}(\boldsymbol{q})$:** that depends only on the **configuration** of the humanoid. We have computed this because we are in a vertical plane.

The formula that we have used is the following:

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

We have decided to use the moving frame approach (recursive algorithm). We use the same approach for both Sirina and Lina.

Implementation: `ComputeDynamicModel3R.m`

Dynamic Model: Inertia Matrix

Given:

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

Goal: find the inertia matrix $M(q)$.

The inertia matrix depends only on the configuration of the model.

Solution:

1. Initialization: ${}^0\varpi_0 = 0 \quad {}^0v_0 = 0 \quad i = 0$
2. Write all the rotation matrix that we have ${}^0R_1 \quad \ldots. \; {}^{n-1}R_n \quad n = n° \; joints$
3. Assign $\sigma_i = 0$ because we have all joints that are revolute
4. Follow these formulas for each link:

$$\boldsymbol{{}^i\varpi_i} = {}^{i-1}R_i^{\,T}\left[{}^{i-1}\varpi_{i-1} + (1-\sigma_i)\,\dot{q}_i\, z\right] \quad z = \begin{pmatrix}0\\0\\1\end{pmatrix}$$

$$\boldsymbol{{}^iv_{ci}} = {}^iv_i + {}^i\varpi_i \times {}^ir_{i,ci}$$

$$\begin{aligned}\boldsymbol{{}^iv_i} &= {}^{i-1}R_i^{\,T}\left[{}^{i-1}v_{i-1} + \sigma_i\dot{q}_i\, z + {}^{i-1}\varpi_i \times {}^{i-1}r_{i-1,i}\right]\\ &= {}^{i-1}R_i^{\,T}\,{}^{i-1}v_{i-1} + {}^{i-1}R_i^{\,T}\sigma_i\dot{q}_i\, z + {}^{i-1}R_i^{\,T}\,{}^{i-1}\varpi_i \times {}^{i-1}R_i^{\,T}\,{}^{i-1}r_{i-1,i}\end{aligned}$$

${}^{i-1}r_{i-1,i}$ is which we can find in the ${}^{i-1}A_i$ (last column, first 3 elements)

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

5. Compute the kinetic energy of EACH link:

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

Kinetic energy of whole model : $T = T_1 + \cdots + T_n$

6. Compute $M(q)$ with Matlab

Dynamic Model: Coriolis and Centrifugal terms

Given:

- Inertia Matrix $M(q)$

Goal: find the Coriolis and Centrifugal terms $C_k(q)$.
The Coriolis and centrifugal terms depend on the configuration and on the velocity.

Solution:

1. We divide $M(q) = [M_1 \quad M_2 \quad M_3 \dots M_n]$ where $n = n° \, joints$

2. We apply the following formulas:

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

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

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

Dynamic Model: Potential energy, gravity terms

Since we are in a **vertical plane**, we have to compute the potential energy.

Given:

- $r_{c,i}$: CoM offset w.r.t RF0
- m_i: mass of each link
- g_i: gravity vector containing g_0 in the gravity position and all other values are 0.

Goal: find potential energy U_{gi} and gravity term $g(q)$.
$g(q)$ depends only on the configuration.

Solution:

1. We compute the potential energy of the whole model by using the following formula:

$$U_{gi} = -m_i \, g_i \, r_{c,i} \quad i = 1, \dots, n \quad \text{where } n = n° \, joints$$

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

Where:

- m_i is the mass of the joint.
- g_i is the gravity term ($g_0 = 9.81 \frac{m}{s^2}$)
- $r_{c,i}$ is CoM offset w.r.t. RF0.

2. I compute the gravity vector by using the potential energy:

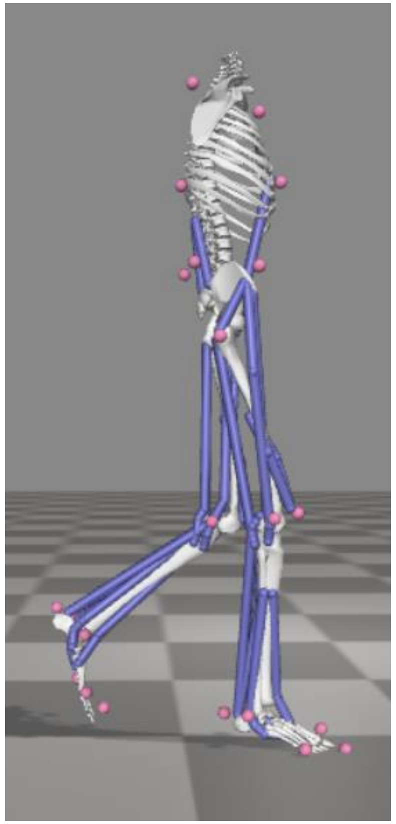
$$g(q) = \left(\frac{\partial U_g}{\partial q_1} \; \frac{\partial U_g}{\partial q_2} \; \dots \; \frac{\partial U_g}{\partial q_n} \right)^T$$

Equilibrium analysis

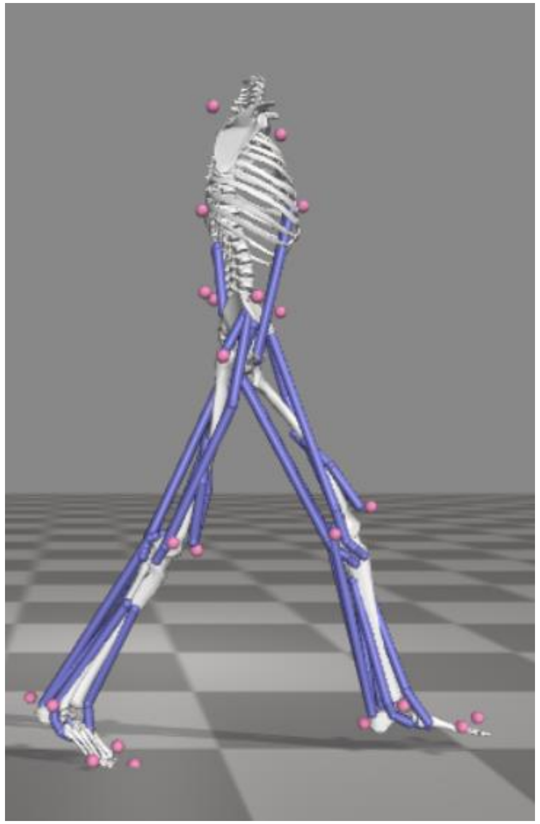
Adult-Child Musculoskeletal Model and Motion Analysis

Sapienza Università di Roma

Equilibrium analysis: Time Segmentation



SINGLE SUPPORT (SS) PHASE: Only one limb in in contact with the ground.



DOUBLE SUPPORT (DS) PHASE: Both feet are in contact with the ground.

We are going to use the trajectory of RTT2 and LLT2 markers as reference:

- **End of the SS phase/Start of the DS phase:** the maximum height of the toe marker (Heel Strike).
- **Start of the SS phase/End of the DS phase:** start of the toe's movement that it's found behind (Toe Off).

These intervals will be further divided for:

- the left limb (SSl, DSl)
- the right limb (SSr, DSr).

Equilibrium analysis: Time Segmentation

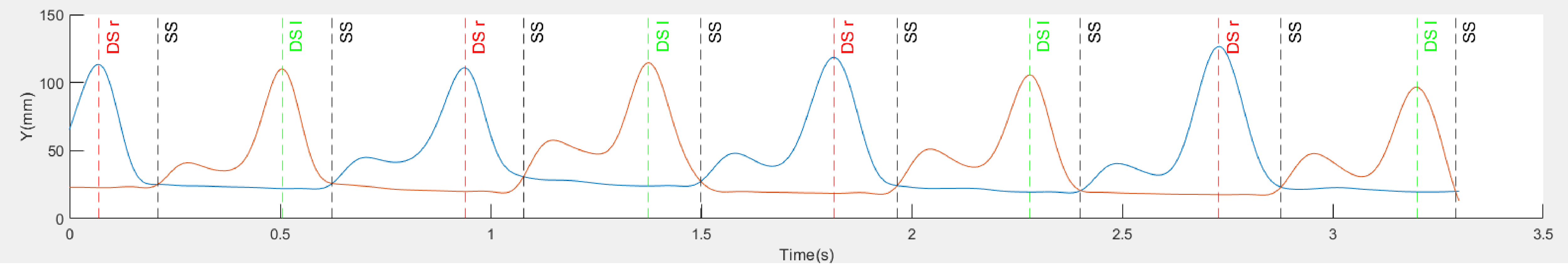


Figure 17: plot of the time segmentation and trajectory of toes markers (RTT2, LTT2). (Sirine, FILTERED_marche 2.trc)

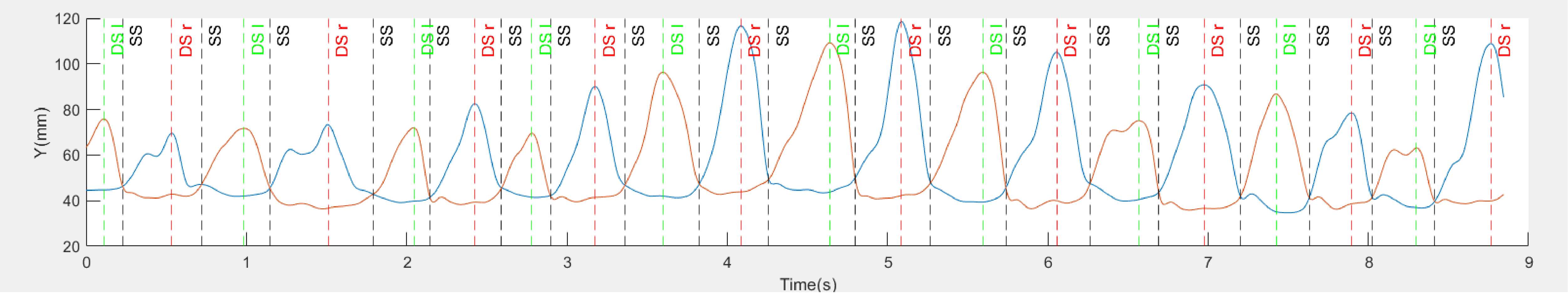


Figure 18: plot of the time segmentation and trajectory of toes markers (RTT2, LTT2). (Lina, FILTERED_marche 5.trc)

Equilibrium analysis: Time Segmentation

	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

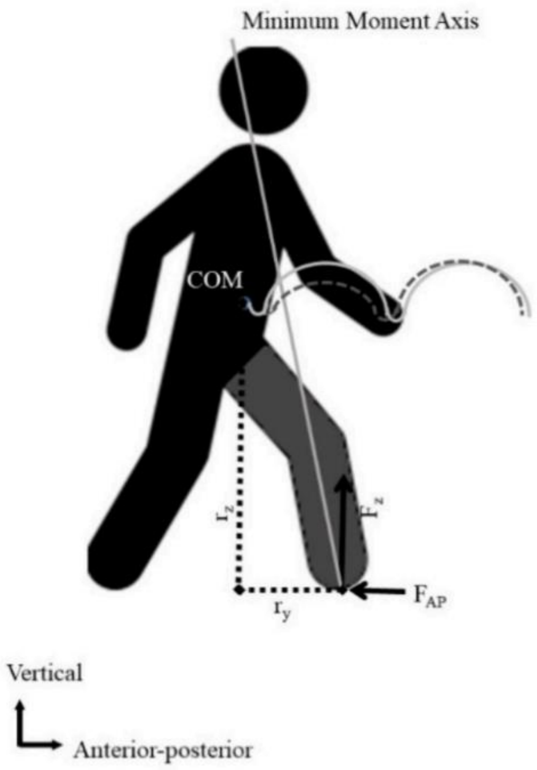
Table 3: Time segmentation for Sirine and Lina.
(means and standard deviation values are computed on a single marche, FILTERED_marche 5.trc for Lina and FILTERED_marche 2.trc for Sirine).

As seen in [3]:

- **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.
- **Higher standard deviations of Lina** compared to the ones of Sirine which is due to the larger gait variability of Lina.

Equilibrium analysis: Computation of $d_{CoM-\Delta}$

Computation of the distance between Minimum Moment Axis (Δ) and the Center of Mass (CoM) [1]:



- **Filter** Force Plate Data (4^{th} order, zero phase-shift, low-pass Butterworth filter with the cutoff frequency equal to 10 Hz);
- **Compute** the moment field expressed at the CoM point:

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

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

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

"Several experiments involving multi-contact locomotion were previewed in the paper of Bailly et al. They conclude that the distance increases as the walking terrain increases in difficulty, implying that this **distance may be used to quantify instability and risks of falling during gait** [3]."[1]

Equilibrium analysis: Computation of $d_{CoM-\Delta}$

Sirine:

- **Heel strike with the right foot**, the outcome's evolution is similar to the healthy subject's gait found in [1]:
  - Δ is farthest away behind the subject's CoM (A).
  - Afterwards we can notice a drop in the distance. Δ moves toward the CoM and tries to cross it.
  - In the Toe Off (B) instant the Δ 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.

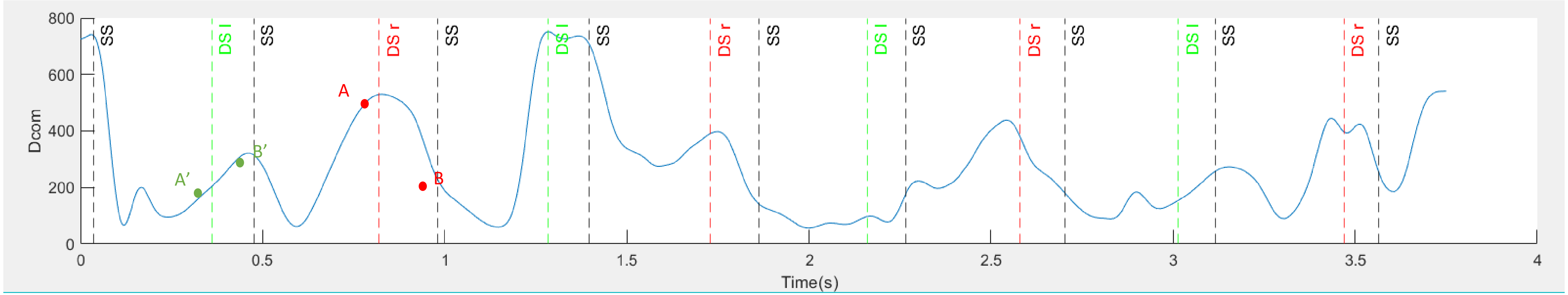


Figure 19: plot of the distance between minimal moment axis and center of mass (CoM) dCoM-Δ, relative to the gait cycle.
This plot is referred to the Sirine model (healthy subject), Marche 6.

Equilibrium analysis: Computation of $d_{CoM-\Delta}$

Lina:

- **Variations** of $d_{CoM-\Delta}$ are very different from the physiological one found in [3] and from Sirine's.

The distance depends on the whole-body dynamics, her pathological condition causes his posture to be irregular during the gait (torso bent over, knee flexed and feet spread apart). Profile of $d_{CoM-\Delta}$ seems coherent with a non-healthy one.

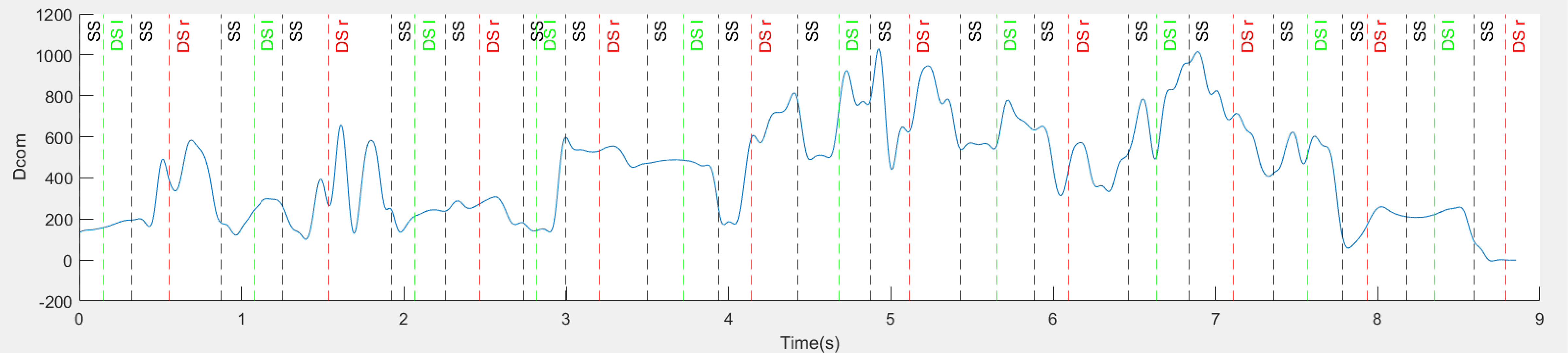


Figure 20: plot of the distance between minimal moment axis and center of mass (CoM) *dCoM-Δ* , relative to the gait cycle. The plot is referred to the lina model (subject that has a spastic CP), Marche 5.

Sources

- [1] Al Abiad N, Pillet H, Watier B. *A Mechanical Descriptor of Instability in Human Locomotion: Experimental Findings in Control Subjects and People with Transfemoral Amputation*. *Applied Sciences*. 2020; 10(3):840. https://doi.org/10.3390/app10030840
- [2] OpenSim Documentation https://simtk-confluence.stanford.edu/display/OpenSim/C3D+%28.c3d%29+Files
- [3]M. Marchitto. *Motion Analysis for Children with Cerebral Palsy: Study on Twins.* Master thesis at Sapienza Università di Roma.
- [4] F. Bailly, J. Carpentier, B. Pinet, P. Soueres and B. Watier, *"A Mechanical Descriptor of Human Locomotion and its Application to Multi-Contact Walking in Humanoids," 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, 2018, pp. 350-356, doi: 10.1109/BIOROB.2018.8488125.

Sapienza
Università di Roma

Thanks for your attention!

Final Project in *Medical Robotics*
September 2021

Catia Carocci, Alessandro Lambertini, Denise Landini