AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
International Credit Hours Eng. Programs I-CHEP
Mechatronics Program

MCT348

Introduction to Biomechanics

Major Task

Supervised by:

Dr. Mohamed Ibrahim

Eng. Hamdy Osama

Submitted by:

Team 3	
Youssef Mohamed	2101009
Mohamed Elsayed	2002088
Abdelrahman Osama	21P0251

Table of Contents

List of Figures

Abstract

This study investigates the application of OpenSim, an open-source musculoskeletal simulation software, in the analysis of human gait biomechanics. The primary objective is to simulate a gait cycle using motion capture data to extract meaningful insights into joint kinematics, muscle force generation, and neuromuscular coordination. Through a systematic approach involving model scaling, inverse kinematics, inverse dynamics, and electromyography (EMG) signal processing, the study highlights how computational modeling can replicate and interpret complex human movements. Furthermore, the report demonstrates the integration of Python-based tools, such as Pyomeca, for processing EMG data to compare simulated muscle activity with physiological measurements. By bridging motion analysis and muscle-level simulations, this project offers valuable contributions to the fields of rehabilitation, clinical gait assessment, and performance optimization in sports biomechanics.

---

1. Introduction

Biomechanics is the science of movement of a living body, including how muscles, bones, tendons, and ligaments work together to produce motion. Gait analysis is a specific subdiscipline within biomechanics that focuses on the study of human locomotion. Accurate understanding of gait is crucial in various domains, such as physical therapy, sports science, prosthetics design, and the diagnosis of neuromuscular disorders.

Recent advances in motion capture technology and biomechanical modeling have enabled researchers to simulate human movement with high precision. OpenSim, a widely used open-source platform developed by Stanford University, allows users to build, scale, and simulate musculoskeletal models for dynamic motion studies. It enables researchers to evaluate joint torques, muscle activation patterns, and residual forces based on empirical data.

The primary objective of this project is to utilize OpenSim to model and analyze a subject's gait cycle using actual motion capture data. Starting from a static anatomical model, the simulation progresses through model scaling, marker adjustment, inverse kinematics (IK), inverse dynamics (ID), and finally to dynamic muscle control simulations using Computed Muscle Control (CMC). The project also integrates EMG signal analysis using the Pyomeca Python library, enhancing the correlation between modeled and real-world muscle activity.

By combining motion data, musculoskeletal modeling, and signal processing, this study not only simulates realistic human gait but also evaluates the physiological plausibility of muscle behavior during different phases of the walking cycle. The insights obtained can be extended to clinical gait evaluation, rehabilitation strategies, and performance enhancement.

---

2. Materials and Methods

The project began with motion capture data collected and saved in the `.c3d` format. This data was converted to `.trc` and `.mot` formats using C3DTools, which are compatible with OpenSim.

The OpenSim software was used for:

- Importing motion data and models
- Scaling the musculoskeletal model to the subject's body
- Running Inverse Kinematics (IK) to calculate joint angles
- Using Inverse Dynamics (ID) to compute joint forces
- Simulating muscle forces with Computed Muscle Control (CMC)

All simulation tools were applied sequentially using a standard lower-limb OpenSim model.

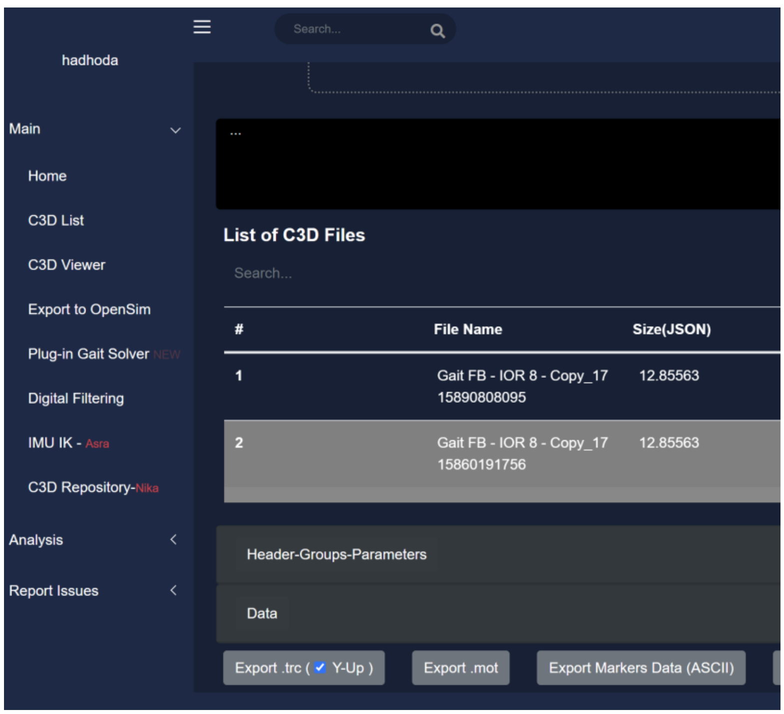


Figure 1 c3d tools used to import the .trc and .mot

3. Simulation Workflow

3.1 Model Setup

A musculoskeletal model (e.g., Gait2392) was selected from the OpenSim model library for its anatomical accuracy and compatibility with gait analysis.

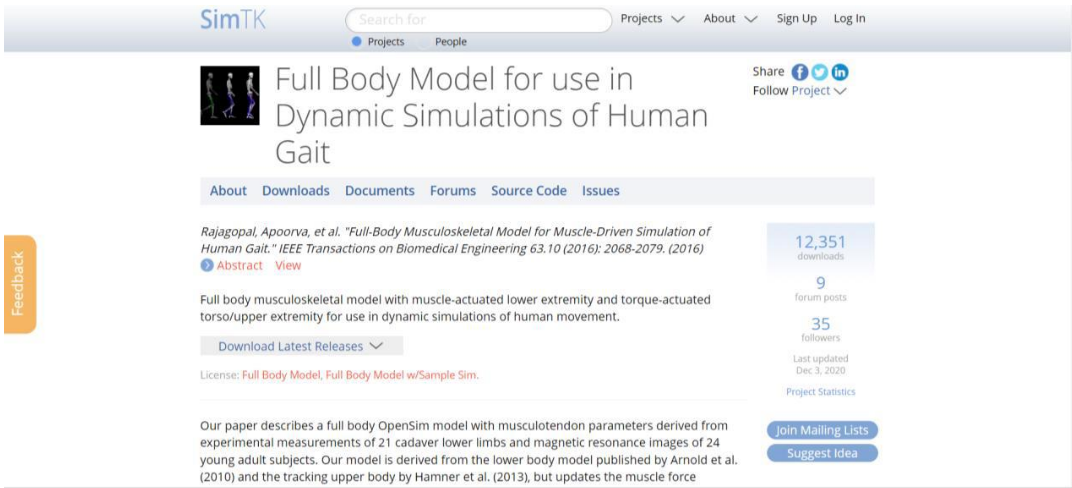


Figure 2 Model used

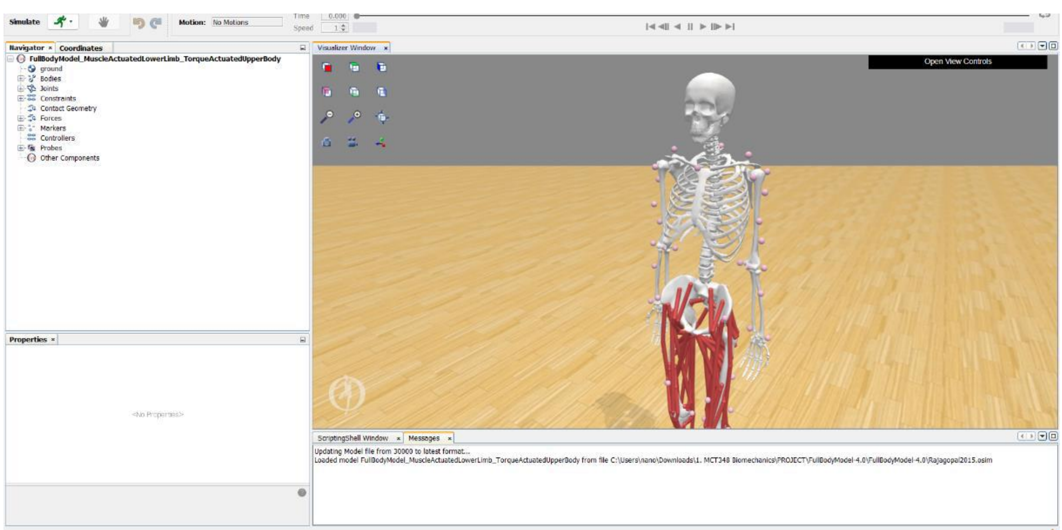


Figure 3 Loaded Model in Opensim

3.2 Marker Editing

Markers from the OpenSim model were modified to match the experimental setup, ensuring that marker tracking during the simulation accurately reflects the actual subject motion.

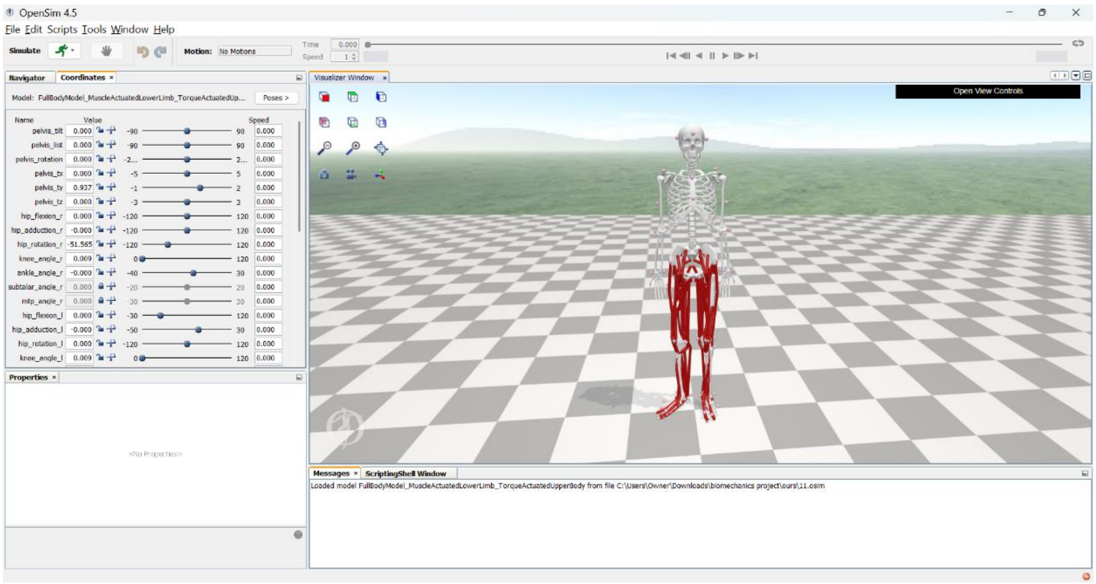
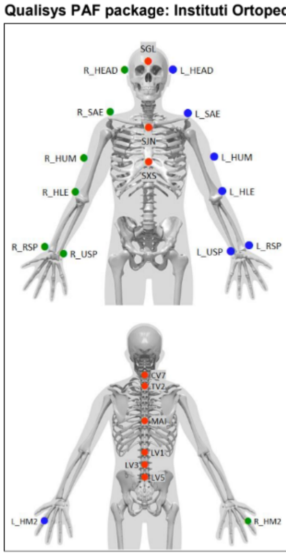


Figure 4 Markers adjusted manually

QUALISYS

Qualisys PAF package: Instituti Ortopedici Rizzoli (IOR) upper body marker set [1]



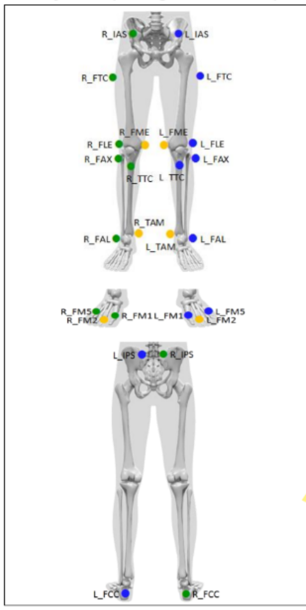
Name	Ref. [2]	Location	Static (23)	Dyn. (23)
L_HEAD		Just above the ear	X	X
R_HEAD		Just above the ear	X	X
SGL	SGL	Glabella	X	X
SJN	SJN	Sternum - Jugular Notch	X	X
SXS	SXS	Sternum - Xiphisternal joint	X	X
CV7	CV7	7th Cervical Vertebrae	X	X
TV2	TV2	2nd Thoracic Vertebrae	X	X
MAI	MAI	Midpoint between inferior angles of most caudal points of the scapulae	X	X
LV1	LV1	1st Lumbar Vertebrae	X	X
LV3	LV3	3rd Lumbar Vertebrae	X	X
LV5	LV5	5th Lumbar Vertebrae	X	X
L_SAE	SAE	Scapula - Acromial Edge	X	X
L_HUM			X	X
L_HLE	HLE	Humerus - Lateral Epicondyle	X	X
L_RSP	RSP	Radius - Styloid Process	X	X
L_USP	UPS	Ulna - Styloid Process	X	X
L_HM2	HM2	Basis of Forefinger	X	X
R_SAE	SAE	Scapula - Acromial Edge	X	X
R_HUM			X	X
R_HLE	HLE	Humerus - Lateral Epicondyle	X	X
R_RSP	RSP	Radius - Styloid Process	X	X
R_USP	USP	Ulna - Styloid Process	X	X
R_HM2	HM2	Basis of Forefinger	X	X

[1] Leardini, A., Biagi, F., Merlo, A., Belvedere, C., & Benedetti, M.G. (2011). Multi-segment trunk kinematics during locomotion and elementary exercises. Clinical Biomechanics, 26, 562-571. doi: 10.1016/j.clinbiomech.2011.01.015

[2] Sint Jan, S. Van (2007). Color Atlas of Skeletal Landmark Definitions. Guidelines for Reproducible Manual and Virtual Palpations. Edinburgh: Churchill Livingstone.

Figure 5 Upper limbs Markers' References

Qualisys PAF package: Istituti Ortopedici Rizzoli (IOR) lower body marker set [1]



Name	Ref. [2]	Location	Static (26)	Dyn. (20)
L_IAS	IAS	Anterior superior iliac spine	X	X
L_IPS	IPS	Posterior superior iliac spine	X	X
R_IPS	IPS	Posterior superior iliac spine	X	X
R_IAS	IAS	Anterior superior iliac spine	X	X
L_FTC	FTC	Lateral side of greater trochanter, 1/3 from proximal end	X	X
L_FLE	FLE	Femur lateral epicondyle	X	X
L_FME	FME	Femur medial epicondyle	X	
L_FAX	FAX	Proximal tip of the head of the fibula	X	X
L_TTC	TTC	Most anterior border of the tibial tuberosity	X	X
L_FAL	FAL	Lateral prominence of the lateral malleolus	X	X
L_TAM	TAM	Medial prominence of the medial malleolus	X	
L_FCC	FCC	Aspect of the Achilles tendon insertion on the calcaneus	X	X
L_FM1	FM1	Dorsal margin of the first metatarsal head	X	X
L_FM2	FM2	Dorsal margin of the second metatarsal head	X	
L_FM5	FM5	Dorsal margin of the fifth metatarsal head	X	X
R_FTC	FTC	Lateral side of greater trochanter, 1/3 from proximal end	X	X
R_FLE	FLE	Femur lateral epicondyle	X	X
R_FME	FME	Femur medial epicondyle	X	
R_FAX	FAX	Proximal tip of the head of the fibula	X	X
R_TTC	TTC	Most anterior border of the tibial tuberosity	X	X
R_FAL	FAL	Lateral prominence of the lateral malleolus	X	X
R_TAM	TAM	Medial prominence of the medial malleolus	X	
R_FCC	FCC	Aspect of the Achilles tendon insertion on the calcaneus	X	X
R_FM1	FM1	Dorsal margin of the first metatarsal head	X	X
R_FM2	FM2	Dorsal margin of the second metatarsal head	X	
R_FM5	FM5	Dorsal margin of the fifth metatarsal head	X	X

[1] Leardini, A., Sawacha, Z., Paolini, G., Ingrosso, S., Nativo, R., & Benedetti, M.G. (2007). A new anatomically based protocol for gait analysis in children. *Gait & Posture*, *26*, 560-571. doi: 10.1016/j.gaitpost.2006.12.018 http://www.c-motion.com/v3dwiki/index.php?title=Tutorial:_Building_the_IOR_Gait_Model

[2] Sint Jan, S. Van (2007). Color Atlas of Skeletal Landmark Definitions. Guidelines for Reproducible Manual and Virtual Palpations. Edinburgh: Churchill Livingstone.

Figure 6 Lower limb Markers' References

3.3 Model Scaling

The model was scaled based on a static trial TRC file. This involved adjusting limb lengths and segment proportions to match the subject's anthropometry.

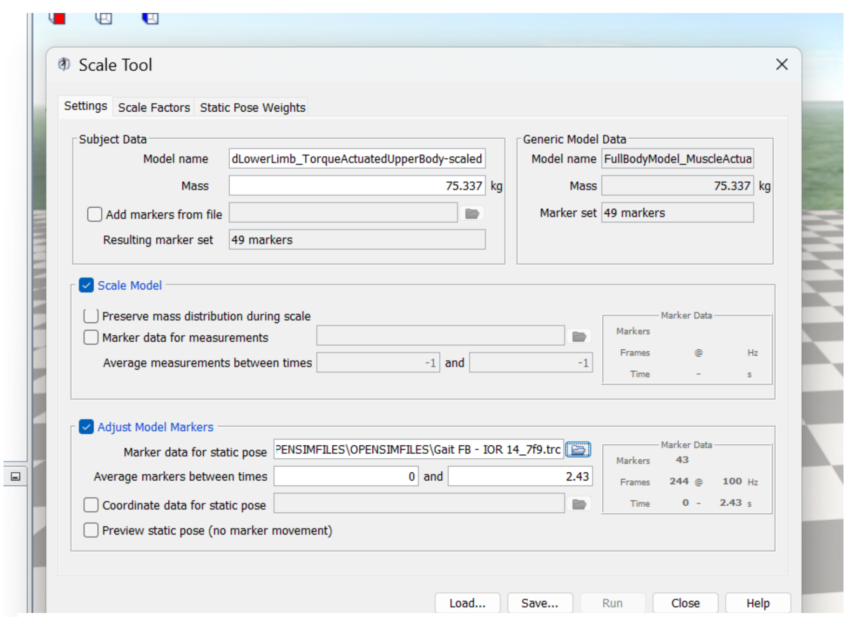


Figure 7 Using the scaling tool with the .trc file of static trial

Adjusting Scale factors from scale tools shown below

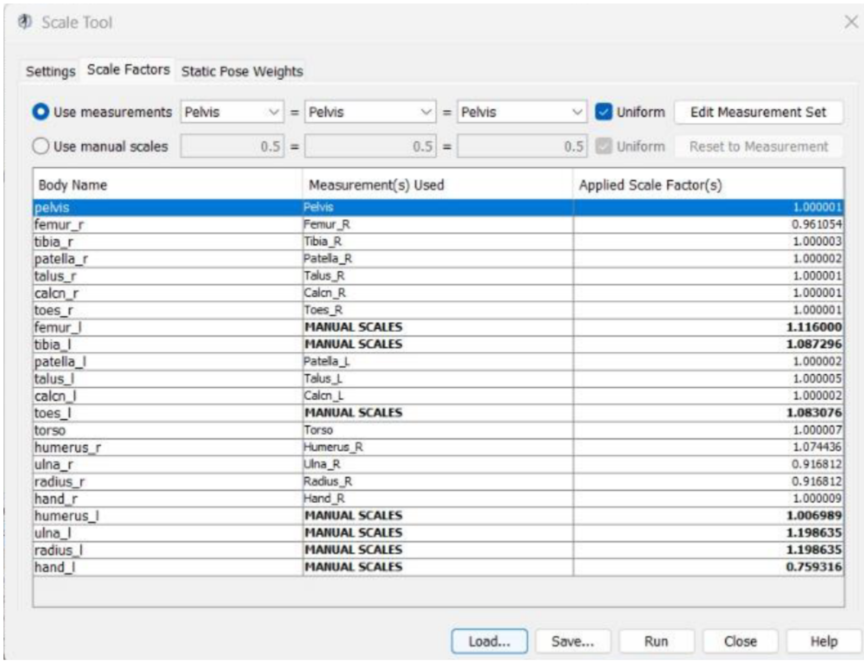


Figure 8 Scale factor Adjustments

3.4 Inverse Kinematics (IK)

IK was used to calculate joint angles by minimizing the difference between the model's and the experimental markers throughout the gait cycle.

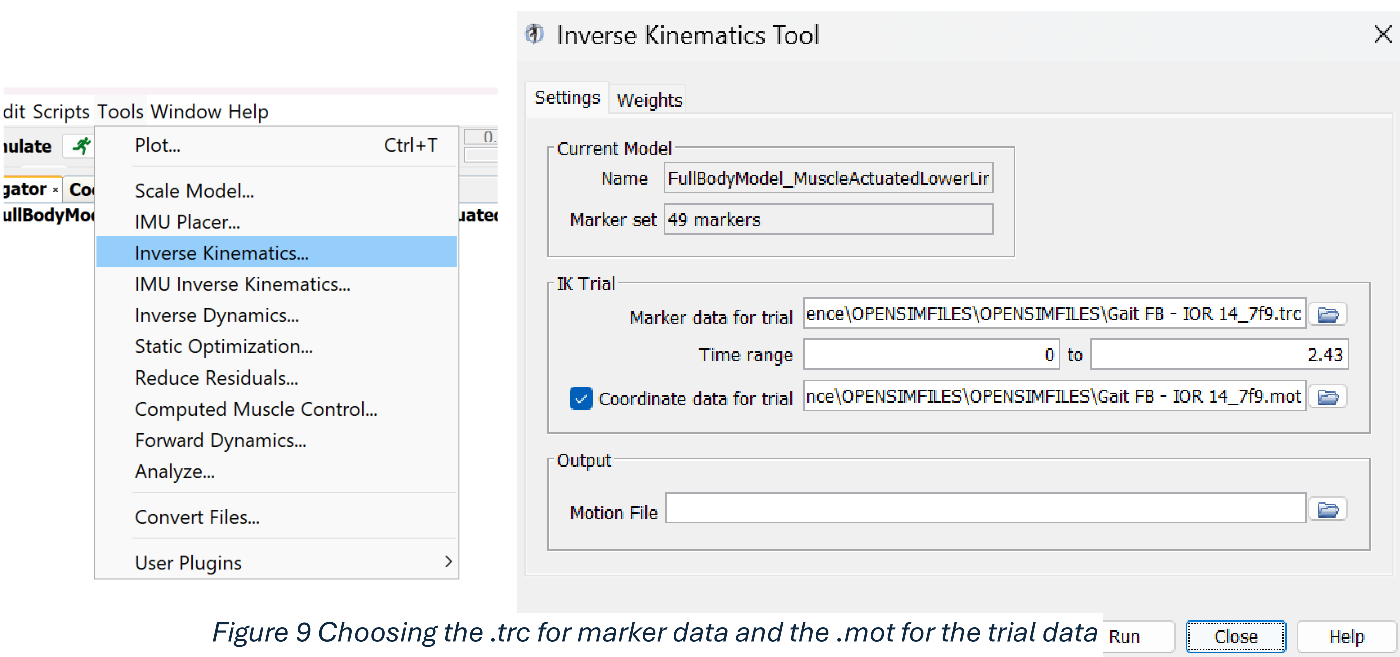
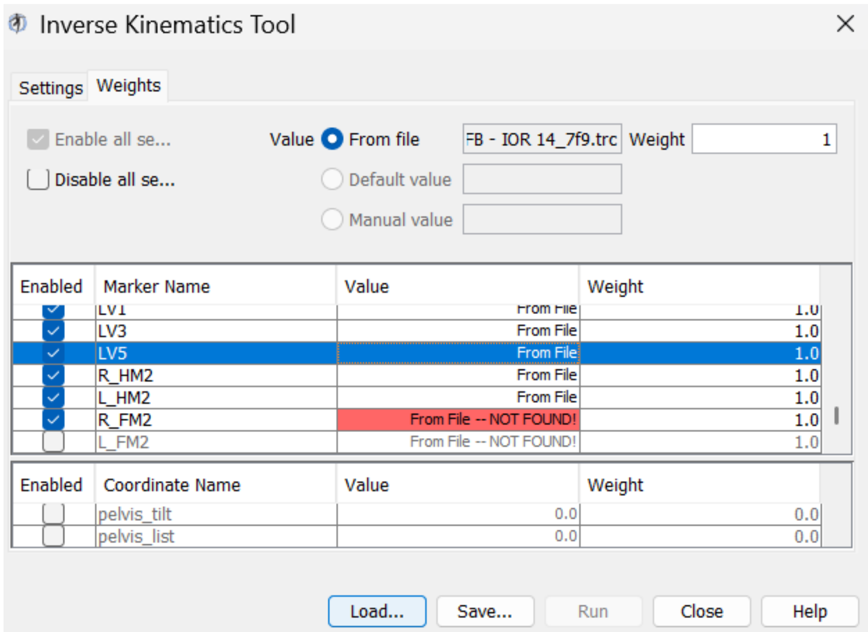


Figure 9 Choosing the .trc for marker data and the .mot for the trial data



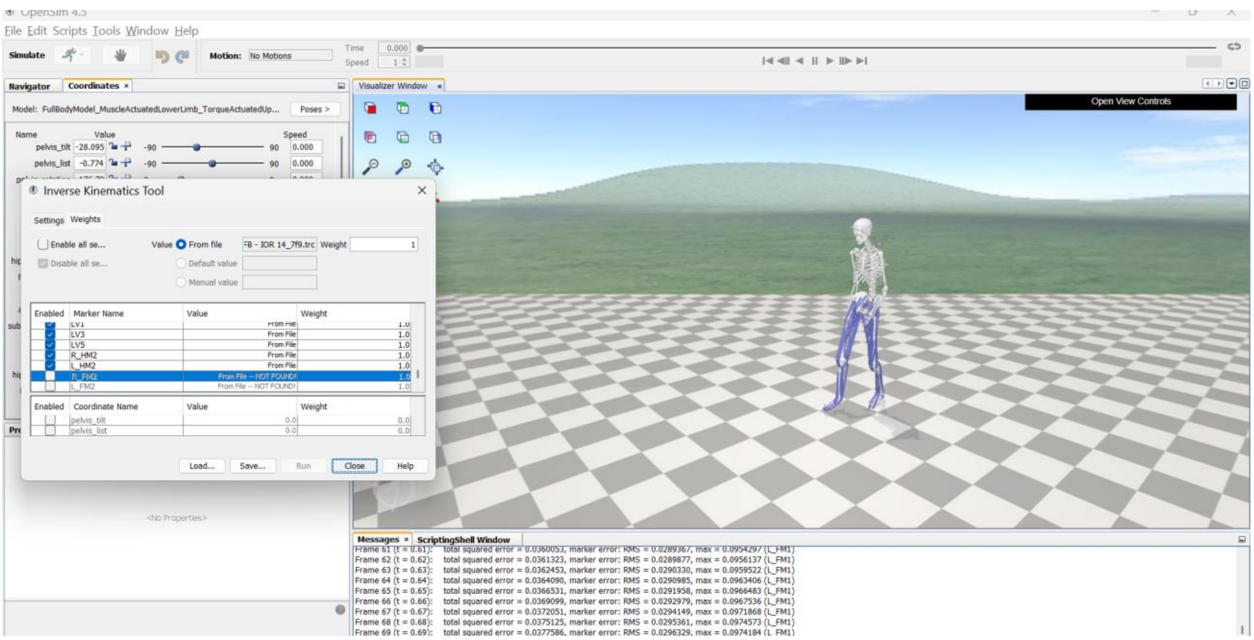


Figure 10 Removing unwanted markers

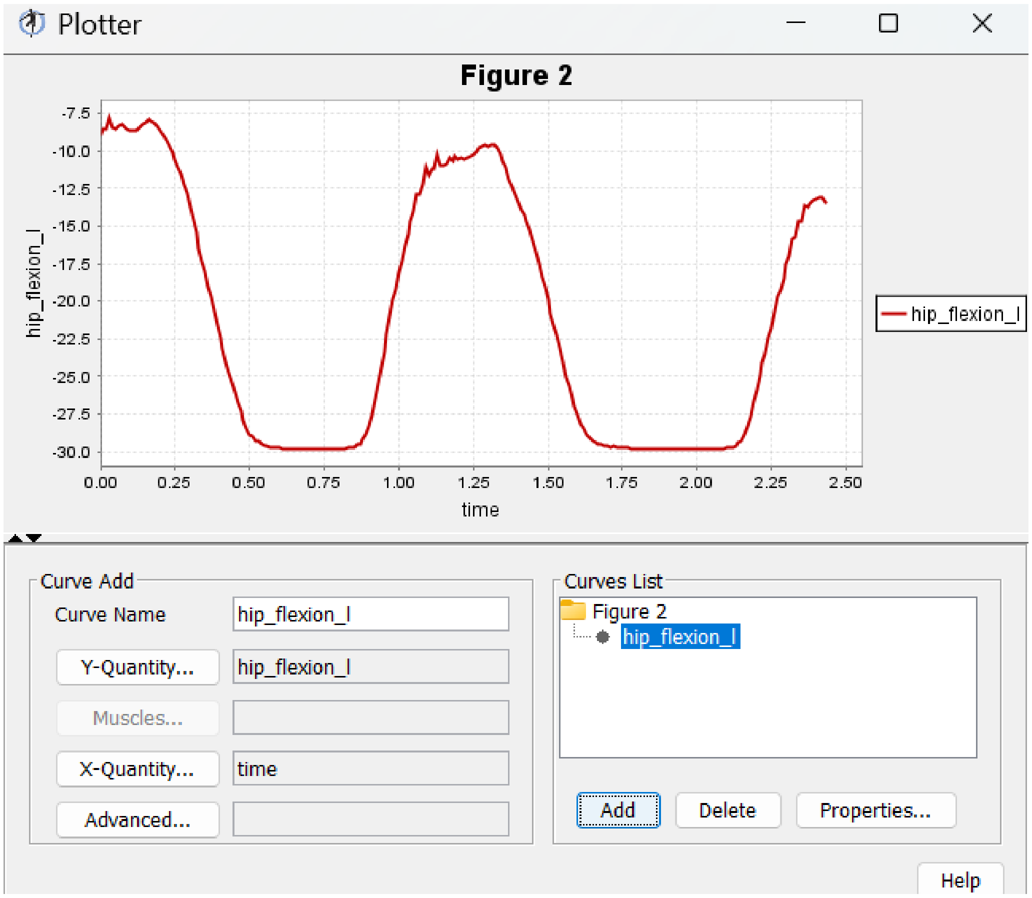


Figure 11 Plotting the hip flexion against time to confirm validity of steps

3.5 Inverse Dynamics (ID)

The Inverse Dynamics (ID) Tool was then used to compute the joint moments and forces from the motion data and the model's kinematic information. This required inputting the ground reaction forces recorded during the gait cycle and applying them to the model to solve for the net forces and moments at each joint.

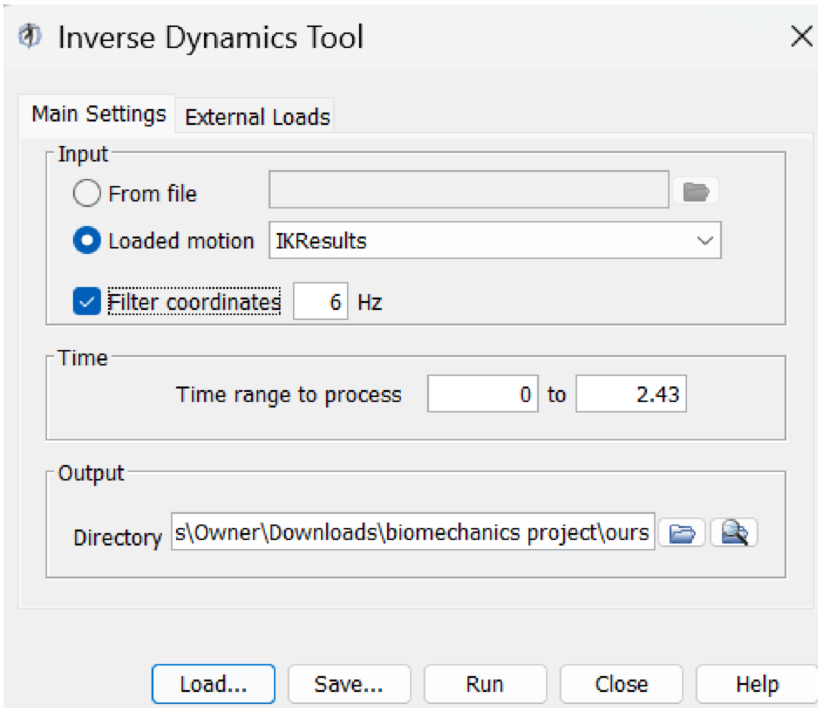


Figure 12 Loading the IK results in the ID tool, and adding force platforms

3.6 Residual Reduction Algorithm (RRA)

RRA was applied to improve dynamic consistency by adjusting model kinematics and reducing non-physical residual forces. This step adjusted the model's mass distribution and joint kinematics slightly to reduce non-physical residual forces and moments, enhancing the accuracy of subsequent analyses.

$$F + F_{res} = MA$$

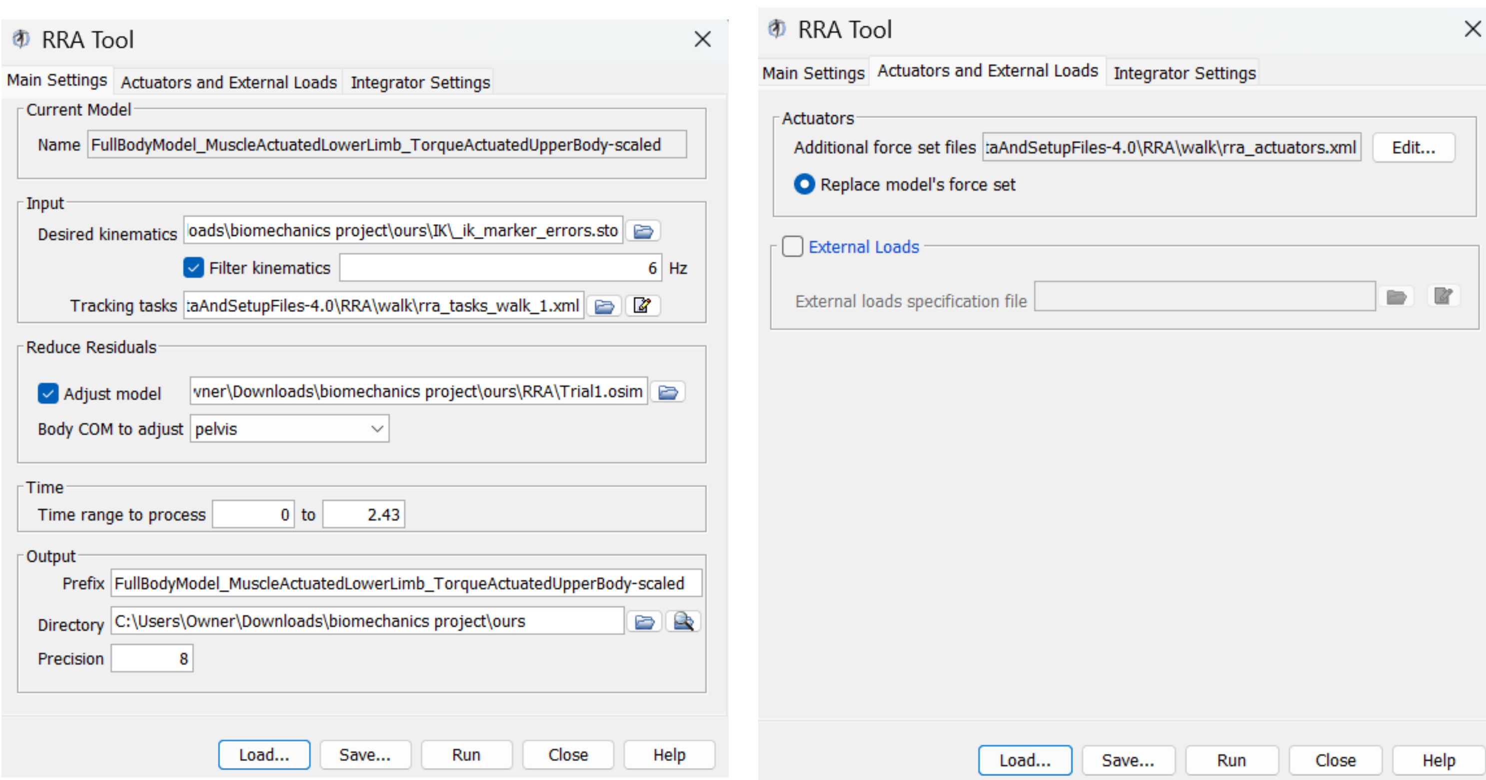
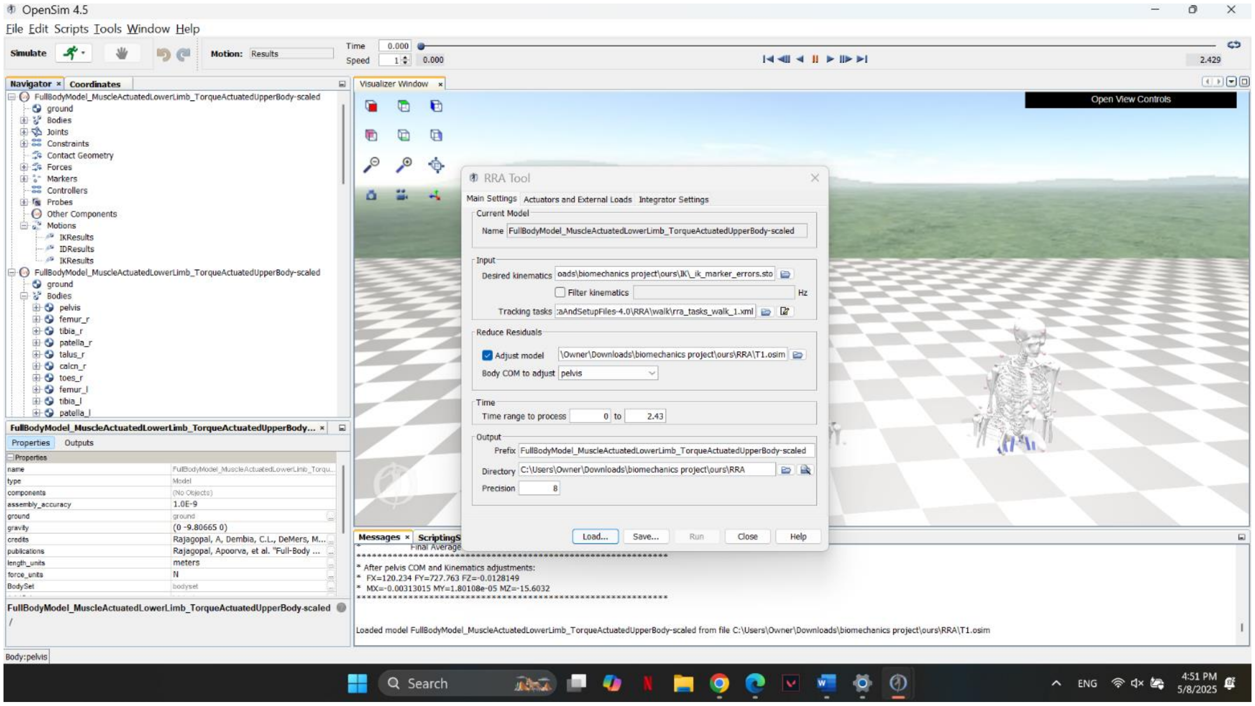


Figure 13 In RRA tools, add the downloaded Tracking tasks, Actuator file, and choosing the pelvis COM to adjust



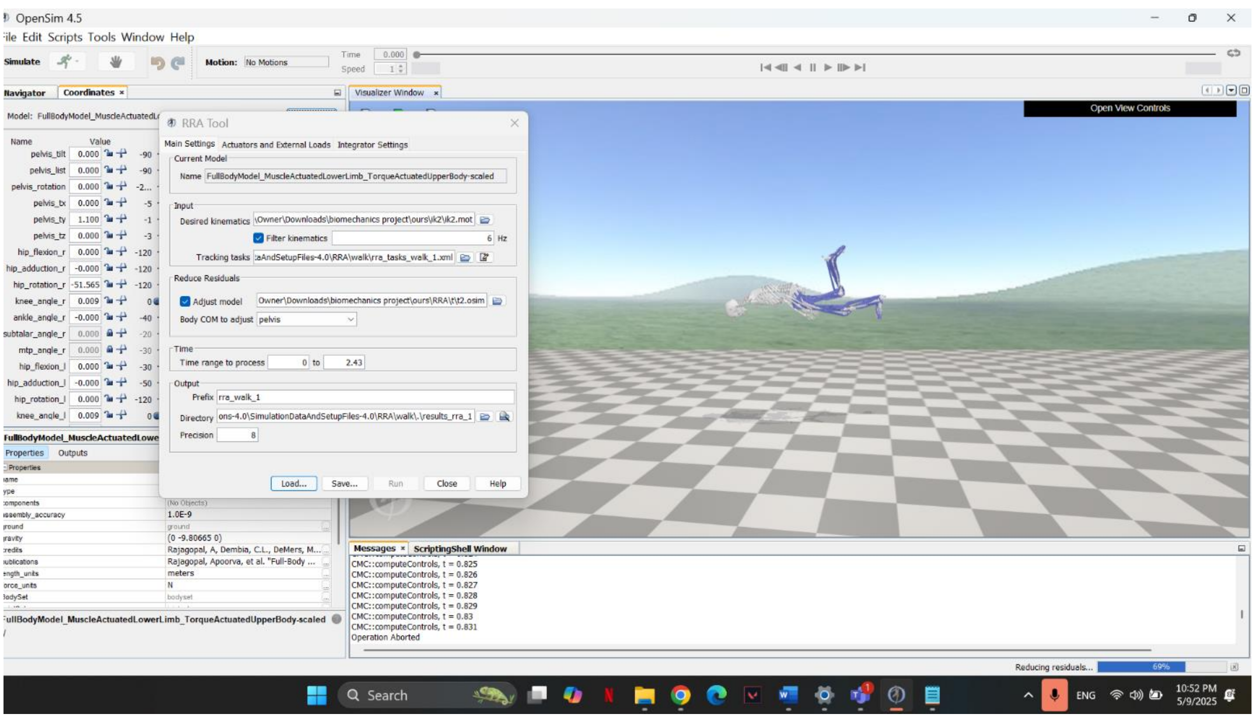


Figure 14 Replaying for multiple trials to get adjusted mass value for CMC

3.7 Static Optimization

Static Optimization was performed to estimate the muscle forces required to produce the joint moments computed previously in the Inverse Dynamics. This tool estimated the muscle forces by minimizing the sum of squared muscle activations required to reproduce the joint torques.

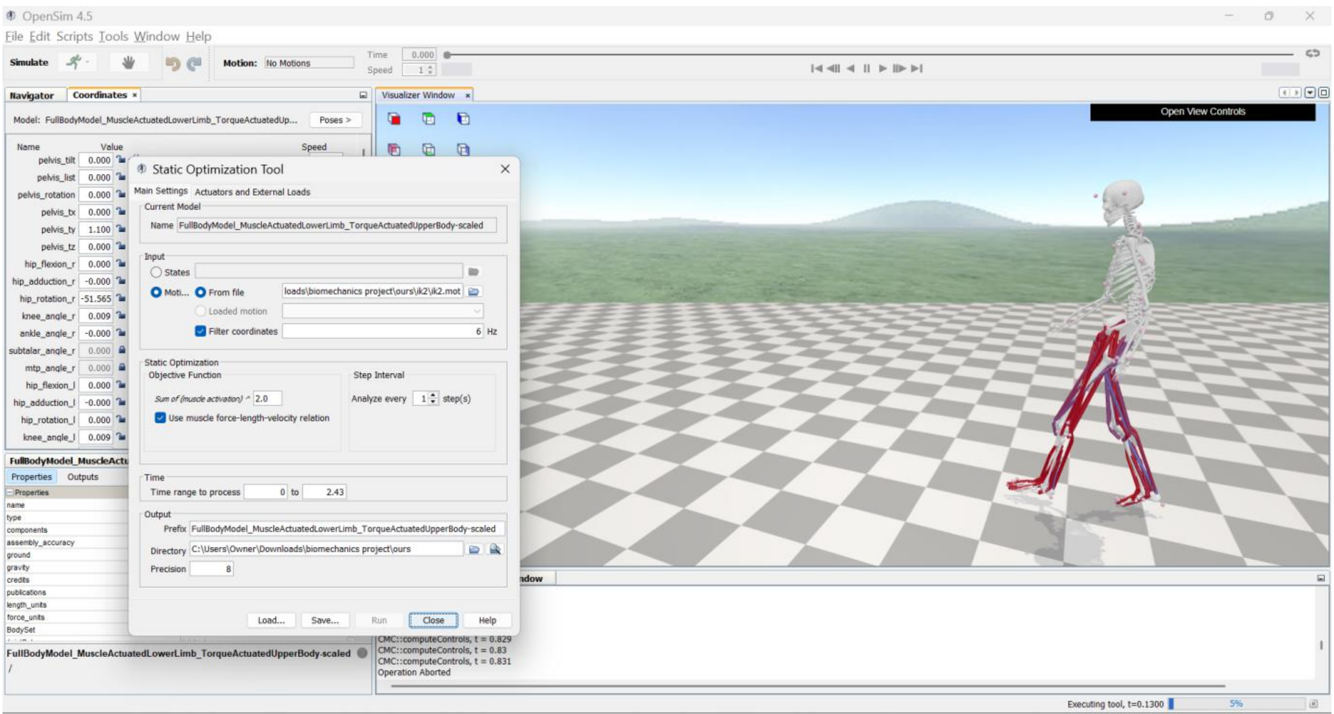


Figure 15 Adding the IK file and the external load xml done previously in ID

3.8 Computed Muscle Control (CMC)

Computed Muscle Control (CMC) Tool was used to generate a forward dynamic simulation of the gait cycle. CMC calculates muscle excitations that drive the model to follow the experimentally measured kinematics, allowing for detailed analysis of muscle coordination and function.

For our Model, the time range to process should not be more than 2, and the CMC look ahead window was ideally at 0.01. These steps deduced was important so that the window does not crash during processing

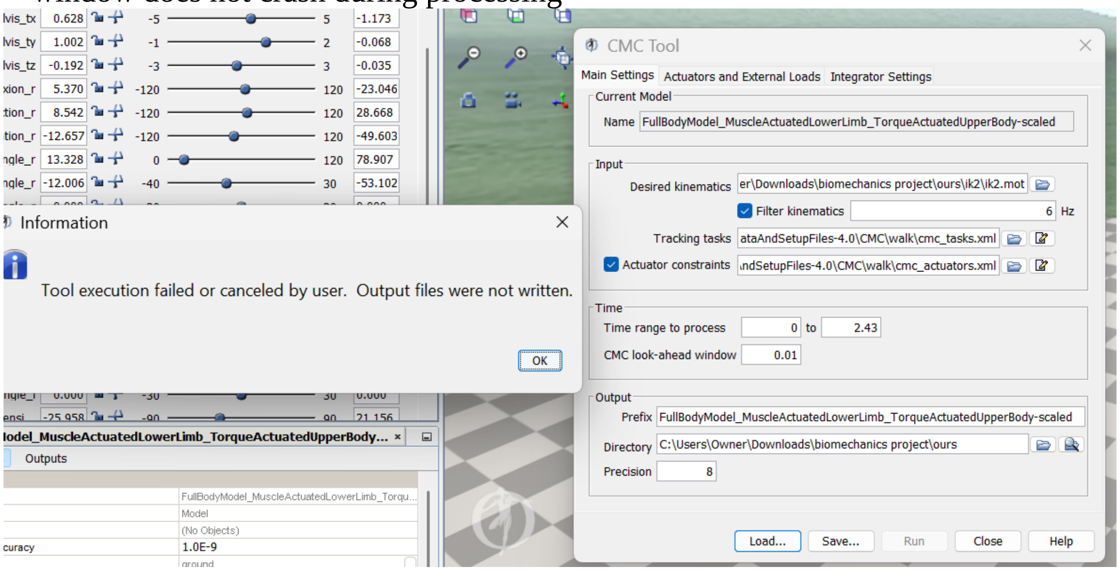


Figure 16 Adding the IK file, filtering the coordinates, and add the Tracking task

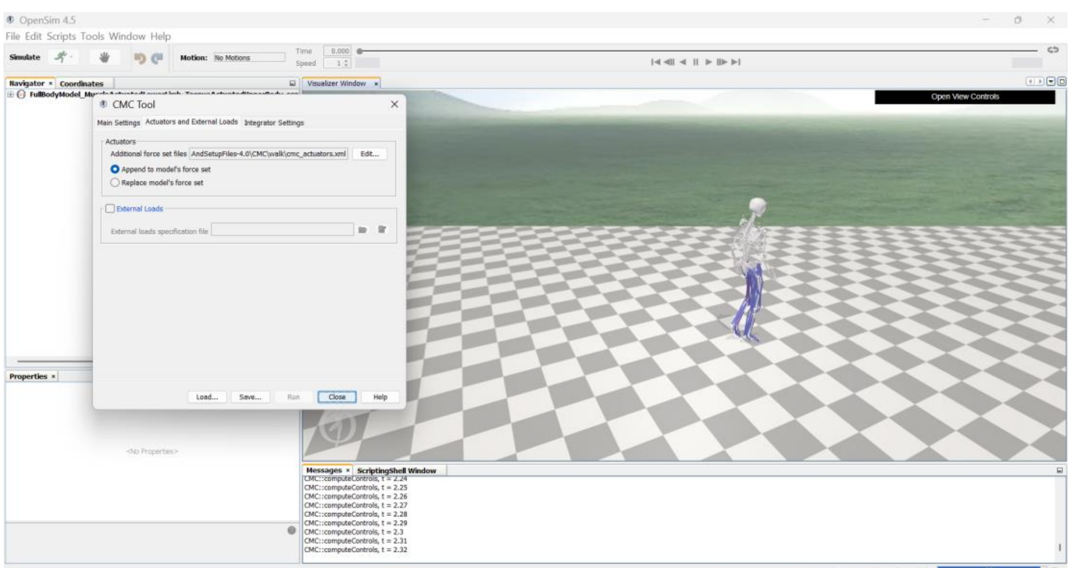


Figure 17 Use the external forces xml used previously for the ID

4. EMG Data Analysis

Electromyography (EMG) data was processed using Pyomeca, a Python library for biomechanical data analysis.

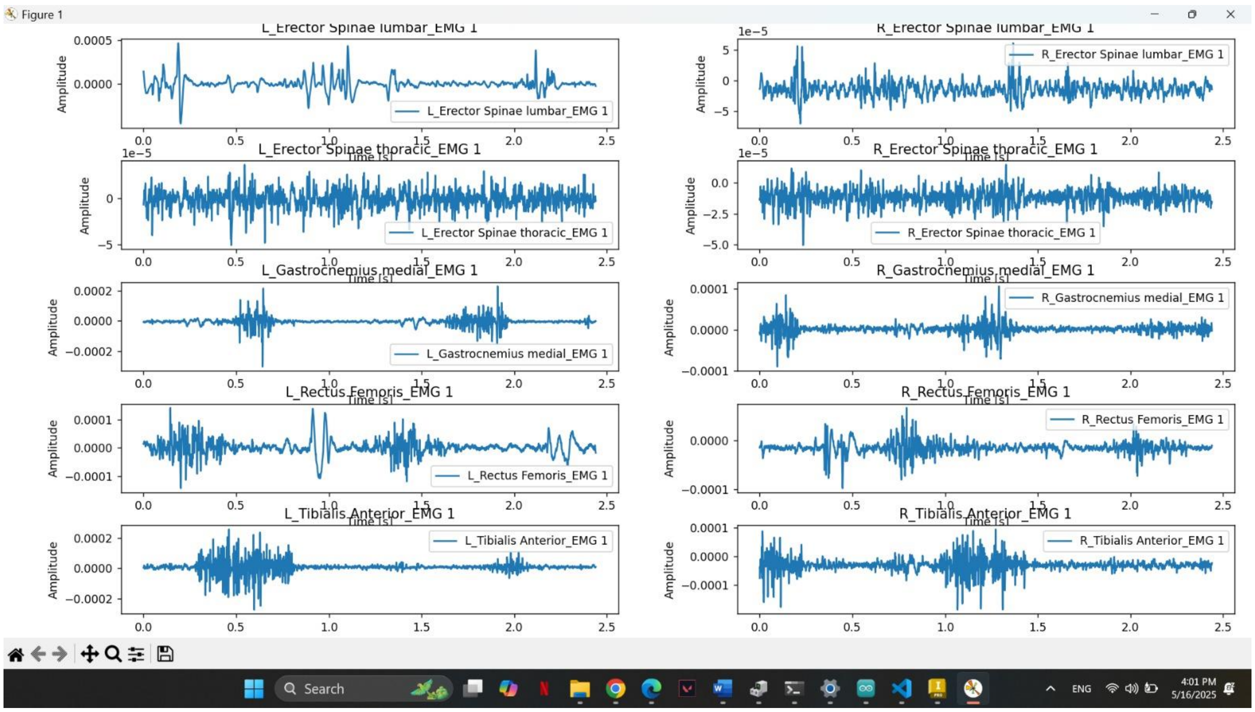


Figure 18 EMG Raw Data

Using Pyomeca, the data was processed using these steps:

1. **Band-pass filtering** (20–450 Hz)
2. **Centering** the signal to remove DC offset
3. **Rectification** (taking the absolute value)
4. **Low-pass filtering** (6 Hz cutoff)
5. **Normalization** to scale the peak signal to 1

Here is the script used for processing the raw EMG data using the pyomeca library:

```python
from pyomeca import Analogs
import matplotlib.pyplot as plt
from scipy.signal import butter, filtfilt
import numpy as np

# File path
file_path = r"C:\Users\Owner\Downloads\C3D gait data (1)\C3D gait data\Gait FB - IOR 14.c3d"

# Load analog data
analogs = Analogs.from_c3d(file_path)

# Get all channel names as strings
channel_names = analogs.coords["channel"].values.astype(str)

# Identify EMG channels by checking name
emg_indices = [i for i, name in enumerate(channel_names) if "emg" in name.lower()]
emg_names = [channel_names[i] for i in emg_indices]

# Extract EMG data using index selection
emg = analogs.isel(channel=emg_indices)

# Filter functions
def bandpass_filter(data, lowcut=20, highcut=450, fs=1000, order=4):
    nyq = 0.5 * fs
    b, a = butter(order, [lowcut / nyq, highcut / nyq], btype='band')
    return filtfilt(b, a, data)

def envelope(data, lowpass=10, fs=1000, order=4):
    b, a = butter(order, lowpass / nyq, btype='low')
    rectified = np.abs(data)
    return filtfilt(b, a, rectified)

# Sampling rate
fs = int(emg.rate)

# Prepare arrays for filtered data
filtered_data = []
enveloped_data = []

# Process each channel
for idx in range(len(emg_indices)):
    raw = emg.isel(channel=idx).values
    filtered = bandpass_filter(raw, fs=fs)
    smooth = envelope(filtered, fs=fs)
    filtered_data.append(filtered)
    enveloped_data.append(smooth)

# Stack processed data to match shape (time, channels)
filtered_emg = np.stack(filtered_data, axis=1)
enveloped_emg = np.stack(enveloped_data, axis=1)

# Plot first 4 channels
plt.figure(figsize=(12, 6))
for i in range(min(4, len(emg_names))):
```

```python
# Stack processed data to match shape (time, channels)
filtered_emg = np.stack(filtered_data, axis=1)
enveloped_emg = np.stack(enveloped_data, axis=1)

# Plot first 4 channels
plt.figure(figsize=(12, 6))
for i in range(min(4, len(emg_names))):
    plt.subplot(2, 2, i + 1)
    plt.plot(emg.time, enveloped_emg[:, i], label=emg_names[i])
    plt.title(emg_names[i])
    plt.xlabel("Time [s]")
    plt.ylabel("Amplitude")
    plt.legend()
plt.tight_layout()
plt.show()
```

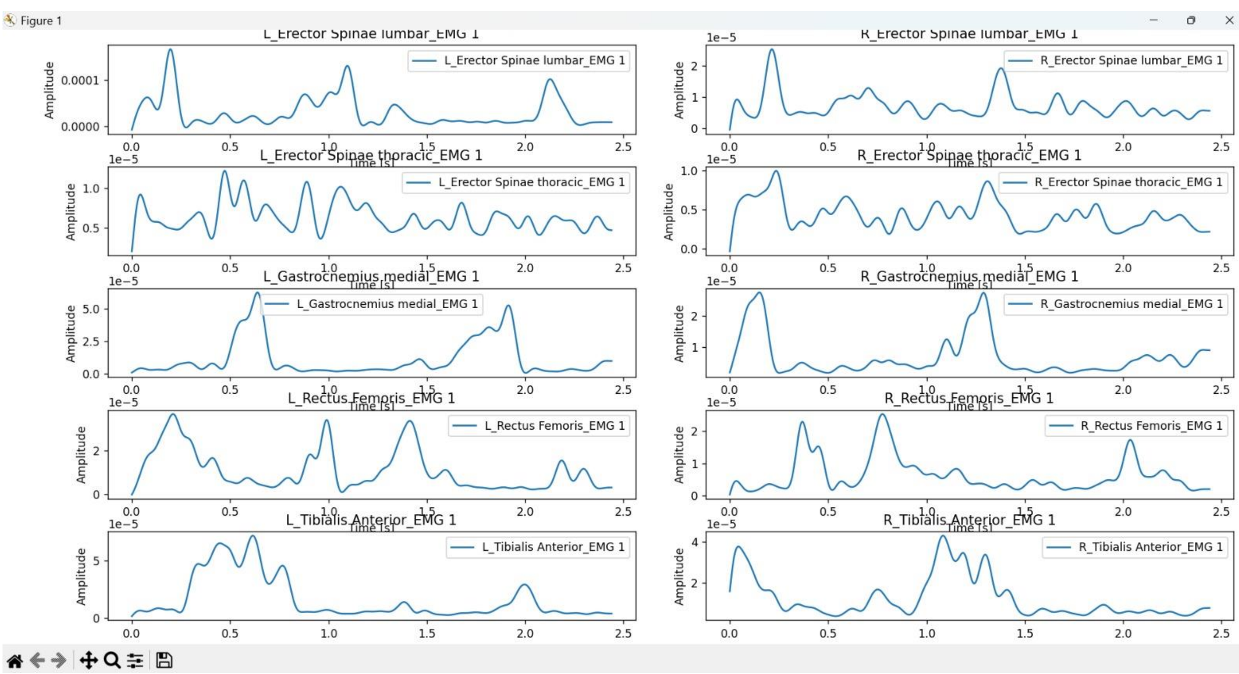


Figure 19 Processed EMG Data

5. Results and Discussion

The gait cycle was divided into two major phases: swing and stance. The most active muscles were identified in each phase.

5.1 Swing Phase

- Tibialis Anterior
- Psoas
- Adductor Magnus
- Gluteus Medius

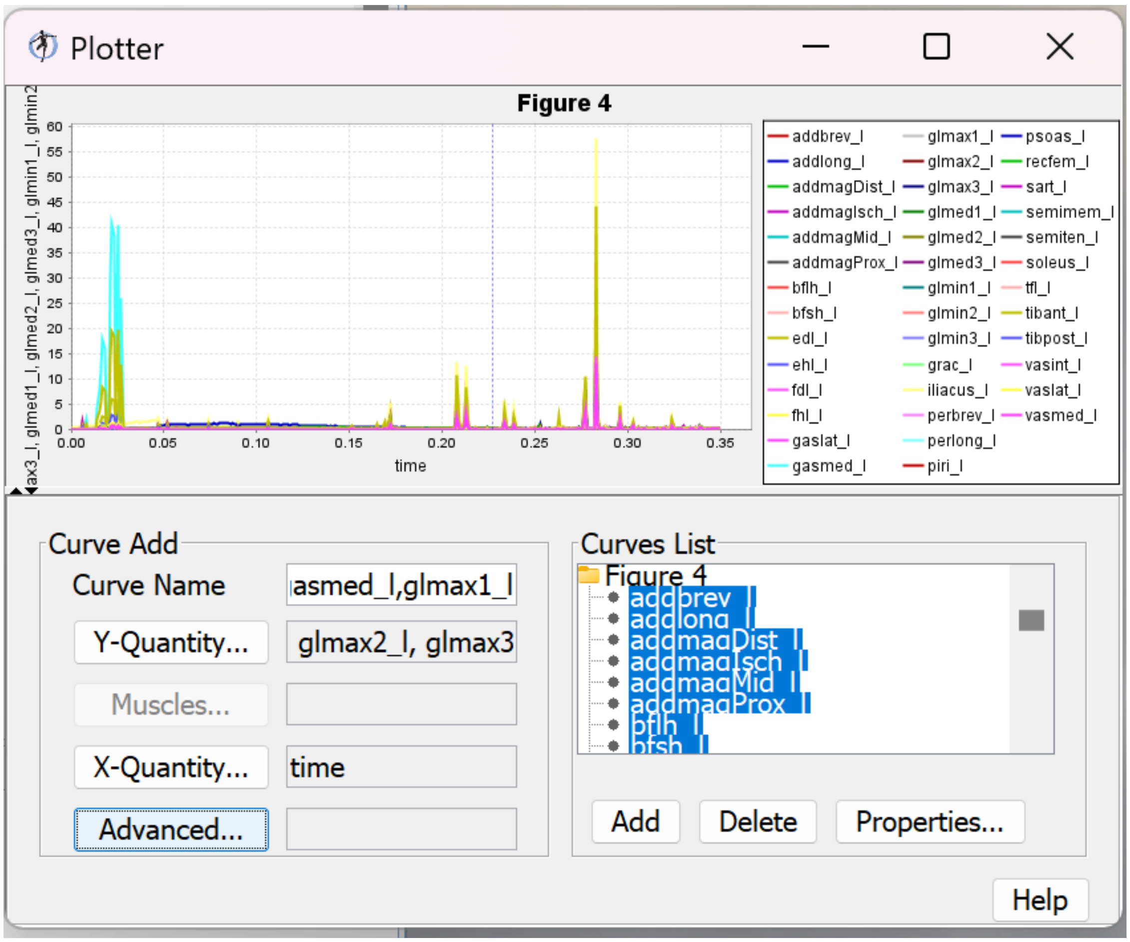


Figure 20 Highly active muscles during Swing phase

Figure 22Tibialis Anterior

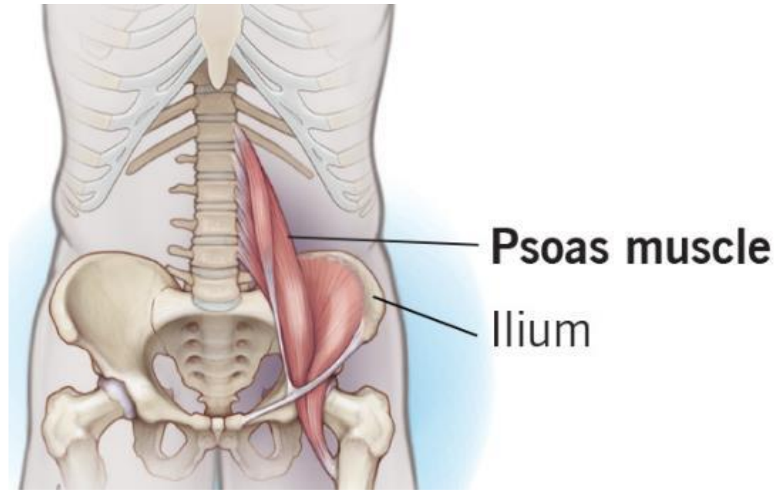


Figure 21 Psoas

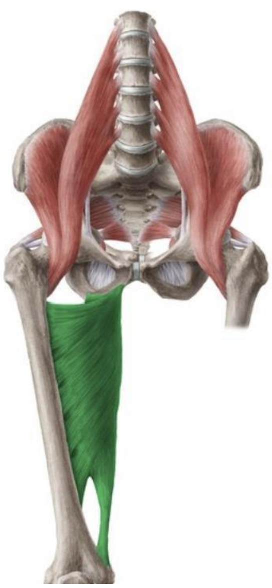


Figure 24 Adductor Magnus

Figure 23 Gluteus Medius

5.2 Stance Phase

- Gastrocnemius Medialis
- Vastus Lateralis
- Tibialis Anterior
- Extensor Hallucis Longus

The simulated muscle activations aligned well with known physiological muscle timing, validating the simulation pipeline.

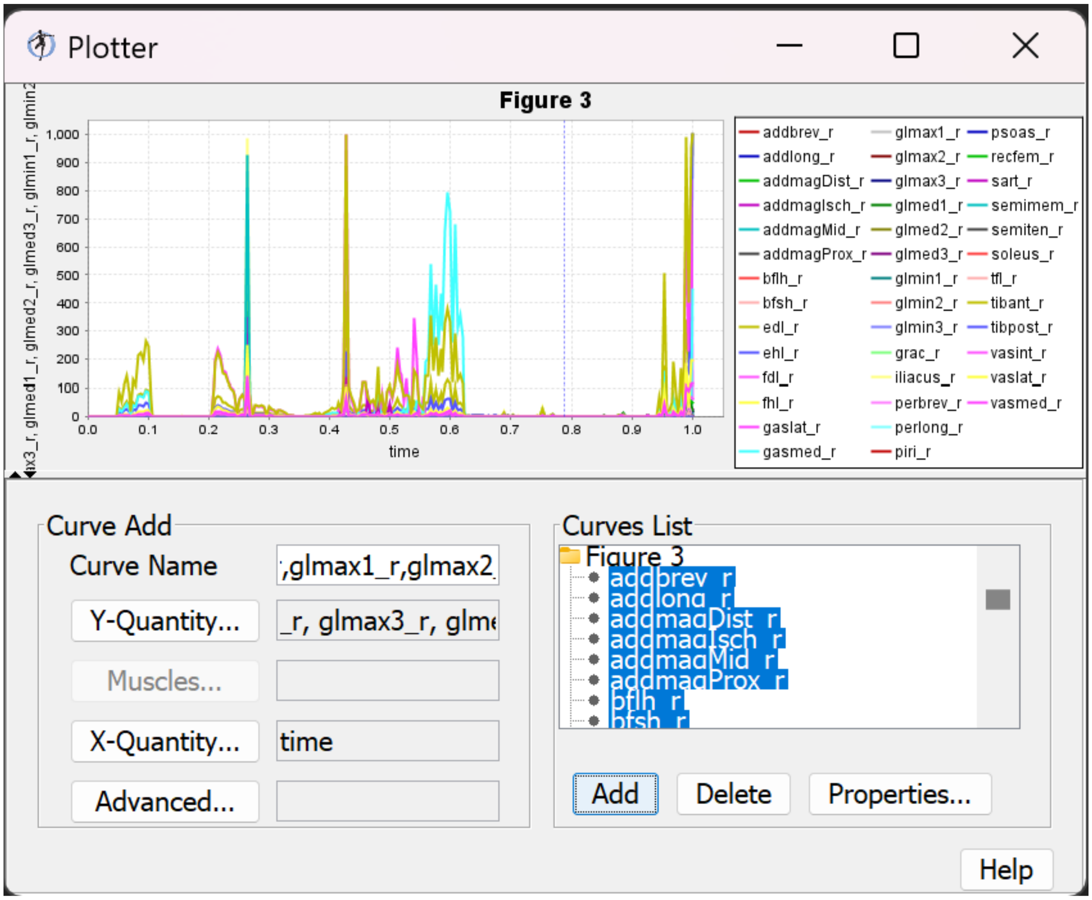


Figure 25 Highly Active Muscles during Stance phase

Figure 27 Gastrocnemius Medialis

Figure 26 Vastus Lateralis

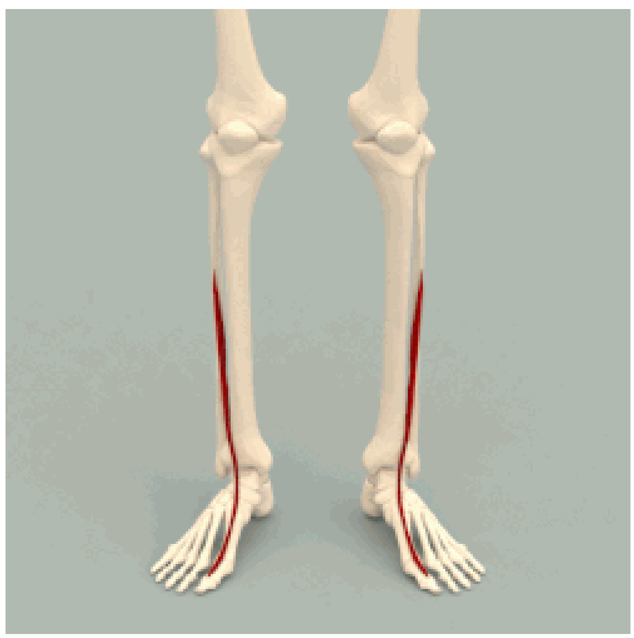


Figure 28 External Hallucis Longus

6. Conclusion

This project successfully demonstrated the application of musculoskeletal modeling and simulation tools in analyzing human gait. By employing OpenSim alongside motion capture data, we created a comprehensive workflow that simulates the biomechanics of walking with remarkable fidelity. The step-by-step methodology—from model scaling and inverse kinematics to static optimization and computed muscle control—allowed us to estimate joint angles, muscle forces, and activation timing throughout the gait cycle.

One of the most valuable aspects of this study was the integration of EMG signal processing through Pyomeca. This allowed for a direct comparison between the model-generated muscle activations and actual physiological EMG signals, adding a layer of validation to the simulation. The observed correlation between simulation and measured data reinforces the credibility of OpenSim as a powerful tool for musculoskeletal research.

Beyond its academic contributions, this project also offers practical implications. The techniques and insights presented can support clinical gait assessment, aiding in the diagnosis and treatment of musculoskeletal disorders. They can also guide rehabilitation strategies by identifying abnormal movement patterns and evaluating the effectiveness of therapeutic interventions. In sports science, the same approach can be used to optimize athlete performance and reduce injury risk by analyzing loading conditions and muscular effort.

In summary, this work highlights how modern computational tools can advance our understanding of complex human movement. As simulation technologies continue to improve, their integration with clinical and performance settings will become increasingly valuable, helping bridge the gap between theoretical biomechanics and real-world human function.

7.Drive Link for Explanation Video

https://drive.google.com/file/d/15aIENcFbvlVvpJTSzDEUqsiG5AwBjU3G/view

8. References

1. Delp, S.L., et al. (2007). *OpenSim: Open-source software to create and analyze dynamic simulations of movement*. IEEE Transactions on Biomedical Engineering.
2. Seth, A., et al. (2018). *OpenSim: Simulating musculoskeletal dynamics and neuromuscular control*. PLOS Computational Biology.
3. OpenSim Documentation: https://opensim.stanford.edu
4. Pyomeca GitHub Repository: https://github.com/pyomeca/pyomeca
5. NIH PMC Article: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4397580/
6. YouTube Tutorials: https://www.youtube.com/@OpenSimVideos