UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

University of Padova
Department of Information Engineering
Master's Degree in Bioengineering

Ground reaction forces estimation using IMU-based kinematics and OpenSim Moco

SUPERVISOR
Prof Zimi Sawacha

CO-SUPERVISOR
Prof Ilse Jonkers
Dr Giacomo Di Raimondo
Dr Bryce A Killen

CANDIDATE
Ludovica Bottini
2021060

Academic Year 2022/2023
April, 3rd 2023

Abstract

The current thesis is the result of the candidate's work over a six-month period at KU Leuven under the supervision of the supervisor and co-supervisors, within a collaboration between the Human Movement Bioengineering Laboratory Research group at the University of Padova (Italy) and the Human Movement Biomechanics Research group at KU Leuven (Belgium).

At the clinical level, gait analysis can provide clinical and functional informaion that can have different application, such as identifying functional impairments associated with a pathological gait patterns. Three-dimensional motion capture (MoCap) is now widely accepted as a method for studying human movement: it consists of a series of precise measurements that are processed by biomechanical models to yield curves related to kinematics and indirect dynamics, such as joint angles and relative forces and moments. These results are generally considered reliable, and used for clinical decision making upon specific interventions to restore a more functional gait of the pathological individuals spanning from rehabilitation treatments to surgery. Despite this, it requires a controlled environment and highly skilled operators. In addition, it was observed that subjects may change their movement strategy when walking on a controlled environment (i.e., a treadmill) instead of an open field overground.

To overcome the need for a laboratory environment, several methods were developed to analyze motion, and one promising approach is represented by inertial measurement units (IMUs) wearable sensors: consisting of accelerometers, gyroscopes, and magnetic sensors. The use of these has increased in the last decade due to the low production costs, their small size, and their portability, allowing studies in everyday life situations. Inertial capture (InCap) systems have become an appealing alternative to 3D MoCap systems due to the ability of IMUs to estimate the orientation of 3D sensors and body segments.

Moreover, the gold standard to measure ground reaction forces (GRFs) is based on force plates (FP) often integrated with MoCap systems in motion analysis laboratories. However, with the increasing interest of using "wearable" approaches in motion analysis, several methods were developed to measure GRFs using movable devices such as wearable sensors: for instance, methods based on insoles that directly measure GRFs, or methods based on IMUs that measure the motion of body segments and estimate GRFs using musculoskeletal models, and finally machine learning based methods [1].

Musculoskeletal modeling and simulation provide an ideal framework for investigating *in silico* quantities such as musculoskeletal loading, muscle forces, and joint contact forces that cannot be measured *in vivo* [2]. OpenSim, an open-source software that allows modeling, analysis, and simulation of the musculoskeletal system, was used in this study.

The aim of this thesis is to develop a workflow to estimate the GRFs based on a novel musculoskeletal model combined with IMUs derived kinematics, and to evaluate its accuracy compared to the gold standard in terms of kinematics, dynamics, and GRFs.
The project was organized as follows: first, a customized model was developed that has spherical contact geometries placed on the feet that can record force values. Following a specific workflow that used OpenSim Moco package [3] it was possible to obtain values in terms of kinematics, dynamics, and GRFs to be compared with the reference ones. In particular, OpenSim Moco is a open-source tool developed to solve optimization problems by using the direct collocation algorithm [4]. As reference data, this project focuses on the InCap approach derived measurements; however, to validate the developed method, MoCap data were also used and results were evaluated and analyzed.
The thesis is divided into five chapters: the first provides a detailed description of gait analysis, including the procedure, guidelines, and instruments used to collect all the necessary data to study human locomotion. In this first chapter characteristics of musculoskeletal modeling can be found, with a focus on the steps required to obtain kinematics and dynamics information. Moreover, it gives an overview of available GRFs estimation methods and OpenSim Moco. The second chapter describes the experimental setup, including all procedures and methods used to achieve the objectives. Here it is present a detailed description of the musculoskeletal model used.

Finally, in the latest chapters, the obtained results are reported graphically and also in terms of Root Mean Square Error (RMSE), correlation coefficient, and maximum differences, as well as a general discussion of the project and conclusions.

Sommario

La presente tesi è il risultato del lavoro del candidato svolto per un periodo di sei mesi, con l'aiuto del relatore e dei supervisori, grazie alla collaborazione tra il gruppo di ricerca del Laboratorio di Bioingegneria del Movimento dell'Università di Padova (Italia) e il gruppo di ricerca Human Movement Biomechanics dell'Università KU Leuven (Belgio).

A livello clinico, l'analisi del cammino è un test utilizzato quale supporto alla decisione per i clinici che può avere svariate applicazioni, in particolare l'identificazione delle limitazioni funzionali aassocciate a un percorso patologico al fine di stabilire la necessità di effettuare un trattamento di riabilitazione piuttosto che un intervento chirurgico. La *Motion capture* tridimensionale (MoCap) è ormai un metodo ampiamente accettato per lo studio del movimento umano: consiste in una serie di misurazioni precise che vengono elaborate da modelli biomeccanici per ottenere curve relative alla cinematica e alla dinamica indiretta, come gli angoli delle articolazioni, le forze e i momenti articolari.
Questi risultati sono considerati generalmente affidabili, e in base a queste curve si decide come intervenire sul soggetto specifico per pianificare eventuali interventi di riabilitazione e chirurgici. Nonostante ciò, questo approccio richiede un ambiente controllato e operatori altamente qualificati. Inoltre, è stato osservato che i soggetti possono cambiare la loro strategia di movimento quando camminano in un ambiente controllato (ad esempio, un tapis roulant) anziché in un ambiente libero.

Per superare la necessità di un laboratorio controllato, sono stati sviluppati molteplici metodi al fine di analizzare il movimento, e uno degli approcci più promettenti è rappresentato dai sensori indossabili inerziali (inertial measurements units, IMUs): composti da accelerometri, giroscopi, e magnetometri, l'utilizzo di questi è aumentato nell'ultimo decennio grazie ai loro costi di produzione contenuti, la loro piccola dimensione, e la loro portabilità, permettendo studi in situazioni di vita quotidiana.

La *Inertial capture* (InCap) è diventata una interessante alternativa al 3D MoCap, grazie alla capacità degli IMUs di stimare l'orientazione di sensori 3D e segmenti.

Inoltre, il gold standard per misurare le forze di reazione al terreno (ground reaction forces, GRFs) è basato sull'utilizzo di pedane di forza (force plates, FP) spesso integrate con sistemi MoCap in laboratori di analisi del movimento. Tuttavia, con il crescente interesse nell'uso di approcci "indossabili" nell'analisi del movimento, sono stati sviluppati diversi metodi per misurare le GRFs utilizzando dispositivi mobili come i sensori indossabili: per citare qualche esempio, metodi basati su solette che direttamente misurano le GRFs, metodi basati su IMUs che misurano il movimento di segmenti di corpo e stimano le GRFs utilizzando modelli muscoloscheletrici, ed infine metodi basati sul machine learning [1].

La modellazione e simulazione muscoloscheletrica forniscono un quadro ideale per lo studio *in silico* di quantità come carichi muscoloscheletrici, forze muscolari, e forze di contatto articolari che non possono essere misurate *in vivo* [2]. In questo studio è stato utilizzato OpenSim, un software open-source che permette la modellazione, l'analisi, e la simulazione del sistema muscoloscheletrico.

Lo scopo di questa tesi è quello di sviluppare un flusso di lavoro atto a stimare le GRFs basato su un nuovo modello muscoloscheletrico in combinazione con una cinematica derivata da IMUs, e di comparare l'accuratezza di tale approccio con il gold standard in termini di cinematica, dinamica, e GRFs.
Il progetto è stato organizzato come segue: innanzitutto, è stato sviluppato un modello personalizzato che presenta delle geometrie sferiche di contatto posizionate sui piedi in grado di registrare valori di forza. Seguendo uno specifico workflow che utilizza il pacchetto OpenSim Moco [3] è stato possibile ottenere valori di cinematica, dinamica, e GRFs da confrontare con i corrispettivi valori di reference. In particolare, OpenSim Moco è un tool open-source sviluppato per risolvere problemi di ottimizzazione utilizzando l'algoritmo di collocazione diretta [4]. Come dati di reference, questo progetto si focalizza sull'approccio InCap; tuttavia, al fine di validare il metodo sviluppato, sono stati usati dati MoCap e sono stati analizzati i risultati ottenuti.

La tesi si divide in cinque capitoli: il primo fornisce una descrizione dettagliata dell'analisi del cammino includendo procedure, linee guida, e strumentazione usata per collezionare tutti i dati necessari al fine dello studio della locomozione umana. In questo primo capitolo è possibile trovare le caratteristiche della modellazione muscoloscheletrica, con un approfondimento sui passaggi necessari ad ottenere informazioni di cinematica e dinamica. Inoltre, il capitolo fornisce una panoramica sui metodi di stima delle GRFs disponibili e di OpenSim Moco. Il secondo capitolo descrive il setup sperimentale, includendo tutte le procedure e i metodi utilizzati. Qui è inoltre presente una descrizione dettagliata del modello muscoloscheletrico utilizzato.

Infine, negli ultimi capitoli, sono riportati i risultati ottenuti graficamente ma anche in termini di Root Mean Square Error (RMSE), coefficiente di correlazione, e massime differenze, assieme alla discussione generale del progetto e alle conclusioni.

Contents

Chapter 1

Introduction

1.1 Human motion analysis

Motion analysis aims to study the movement of any object; it involves the use of mathematical and computational methods to analyze, model, and understand the motion of objects in the real world. This can encompass a wide range of applications, from analyzing the motion of vehicles, robots, or industrial machinery to understanding the motion of celestial bodies in space. Motion analysis can also be used in fields such as engineering, physics, and computer science to study and optimize the motion of various systems. The techniques used in motion analysis can include mathematical modeling, numerical simulation, and experimental measurement. The goal is to gain a better understanding of the dynamics of motion and how different factors, such as forces, masses, and control inputs, affect the motion of objects.

Human motion analysis, a branch of motion analysis, refers to the study of the movement of the human body. It involves the use of techniques from various fields such as biomechanics, computer vision, and machine learning, to analyze, model, and understand human motion [5]. The goal of human motion analysis is to gain a better understanding of how the human body moves, how it is controlled, and how it responds to different stimuli. This knowledge can be used in a variety of applications, such as sports training, rehabilitation, animation, and human-computer interaction. By analyzing human motion, it is possible to identify areas for improvement, track progress, and ultimately optimize performance in a wide range of activities.

The main objectives of human motion analysis are:

- To obtain the motion of selected point, subsequently the movement of body parts, joints, and the entire human body;
- To derive the muscular activity;
- Finally, know the loads acting on tissues during specific motor tasks.

Moreover, this approach aims to offer descriptive details regarding both physiological and pathological cases; by this, is possible to designing therapeutic routes, assessing functional limitations, tracking the effectiveness of rehabilitation therapy, and documenting the usage of aids and prosthesis.

1.1.1 Basic human body movements

To understand in detail how motion analysis works, it is necessary to introduce the basic movements that the human body can make.
The main axes and planes that describe human body movements are depicted in [Fig. 1].

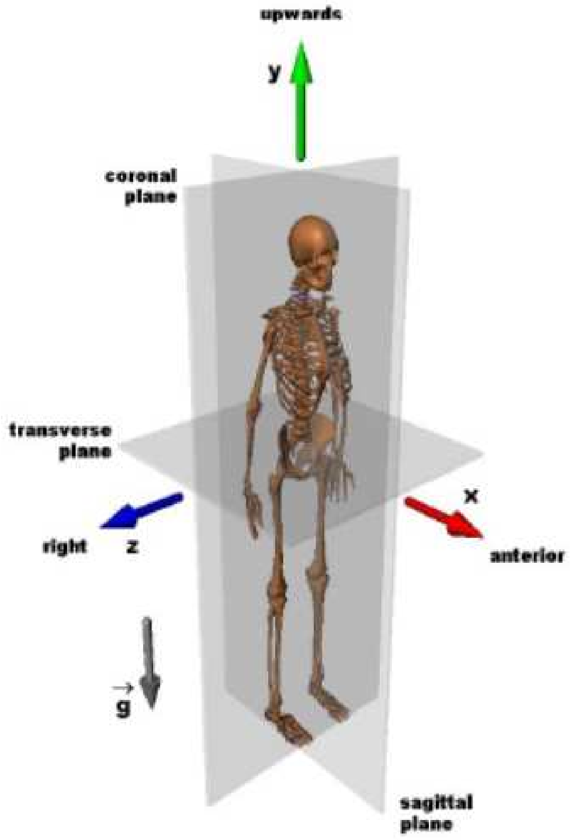


Figure 1: Anatomical coordinate axes and planes [2].

The **main axes** are:

- X, the **antero-posterior** axis;
- Y, the **vertical** axis;
- Z, the **medio-lateral** axis.

With these axes, it is possible to identify the **main planes**:

- (X, Y), the **sagittal** plane, which divides the human body into two *symmetrical* parts (right and left);
- (Y, Z), the **frontal or coronal** plane, which divides the body into two *asymmetrical* parts (anterior and posterior);
- (X, Z), the **transverse** plane, which divides the body into two *asymmetrical* parts (upper and lower).

The principal movements of the human body are performed around the principal axes and lie in the principal planes; in particular, these movements are classified into:

- **Flexion / extension**, around the Z axis in the sagittal plane;
- **Adduction / abduction**, around the X axis in the frontal plane;
- **Internal / external rotation**, around the Y axis in the transverse plane.

Other movements are possible and result from the combination of the latter.
In order to better understand the movements previously introduced, [Fig. 2] represents these applied to the hip joint.

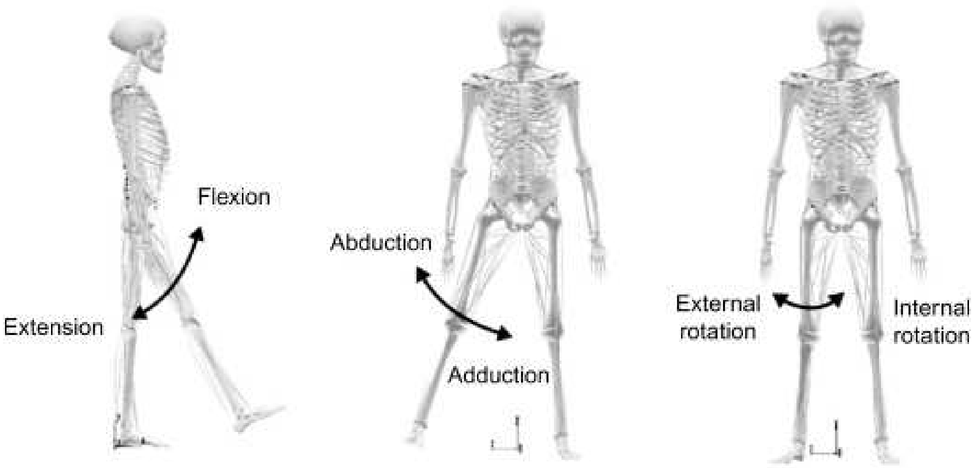


Figure 2: Hip joint movements [6].

1.1.2 Kinematics of the musculoskeletal system

The information obtained from the human motion analysis refers to a model of the **musculoskeletal system**: the anatomical segment is *kinematically* defined by the movement of its bone component [2].

By assumption, the various anatomical segments composing the human musculoskeletal system are considered as **rigid body**: thereby, during the motion the constituent particles of the segment under investigation do not deform or change their relative positions under the influence of applied forces [2].

Therefore, a system of reference defined using significant geometric points is always maintained in the same relationship with respect to the body itself. A rigid body can be defined by the position of its three points, where two of them are required to determine the axis around which the body could spin, while the third point precludes rotation as long it is not parallel to the other two.

To determine the position of any other point of the rigid body it is necessary to know its relative position with respect to the three points chosen to establish the rigid body's position.

This relative position is expressed with respect to a so called **local reference system**, a reference system integrated in the rigid body which can be defined as follows:

1. The **origin** is one of the chosen three points;
2. The **first axis** corresponds to the line connecting the origin and one of the other two points;
3. The **second axis** is directed perpendicular to the plane identified by the chosen three points;
4. The **third axis** is perpendicular to the two other axes.

The local reference system just defined has twelve (12) parameters that locate it in space: the three (3) **coordinates** of the cartesian axis sysyem origin and the nine (9) **director cosines** of the (x, y, z) axes (three per axis) [2]. In particular, the director cosine uniquely describe the orientation of the local system with respect to the global one, but they are not independent from each other since the axes of the reference systems are orthogonal to each other.

Moreover, by applying *the fundamental equation of trigometry*, the rigid body has six (6) degrees of freedom (DOFs) where three (3) of them are the free coordinates necessary to define the translation and three (3) for the orientation in space (i.e., the rotation with respect to the global reference system) [Fig. 3] [2].

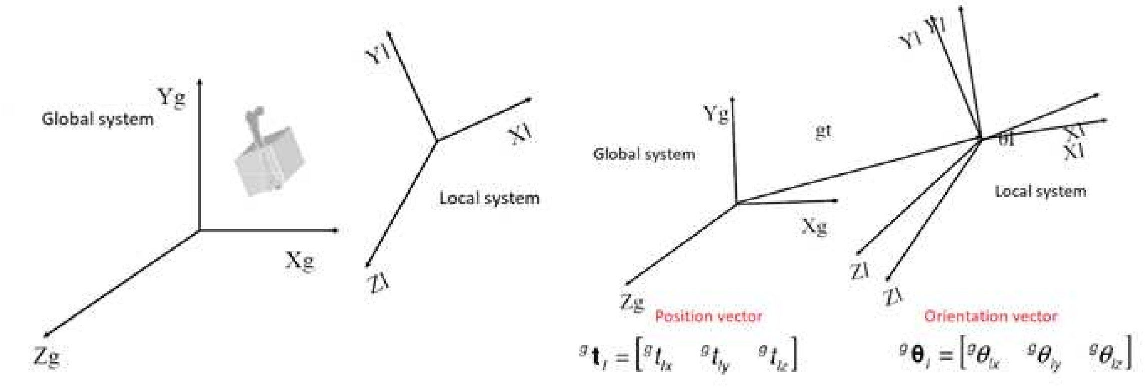

Global system
Yg
Xg
Zg
Yl
Xl
Zl
Local system
Yg
Global system
gt
Xg
Zg
Yl
Yl
θl
Xl
Xl
Local system
Zl
Zl
Position vector
Orientation vector
${}^{g}\mathbf{t}_{l} = [{}^{g}t_{lx} \quad {}^{g}t_{ly} \quad {}^{g}t_{lz}]$
${}^{g}\boldsymbol{\theta}_{l} = [{}^{g}\theta_{lx} \quad {}^{g}\theta_{ly} \quad {}^{g}\theta_{lz}]$

Figure 3: The global (**GLO**) and the local (**LOC**) system of a rigid body and its six (6) degrees of freedom [2].

Joint kinematics After defining the global and the local reference system of the rigid body, the following step is to determine the relative motion between two consecutive segments of the human body.

To determine the position of an anatomical segment in the GLO it is considered the LOC rigidly associated with rather than the segment itself. After locating the LOC with the rigid body in space, the "relative" coordinates of any point belonging to it are known: to obtain these coordinates in the GLO it is necessary to apply a *roto-translation transformation* [2].

The formula that expresses this operation, in particular a rigid transformations of points at each instant of time, is as follows:

$$\begin{bmatrix} P_{Xg} \\ P_{Yg} \\ P_{Zg} \end{bmatrix} = \begin{bmatrix} O_{xg} \\ O_{yg} \\ O_{zg} \end{bmatrix} + \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} \begin{bmatrix} P_{xl} \\ P_{yl} \\ P_{zl} \end{bmatrix} \tag{1}$$

where P_g and P_l are the position vectors of the generic point P of the bone segment expressed respectively in the GLO and LOC systems, the rigid translation is represented by the translation vector O_g, and the transformation of the coordinates of a point from a local and "rotated" reference system to the coordinates of the same point in a global and "fixed" reference system is given by the rotation matrix A:

$$A = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta_{x_g x_l}) & \cos(\theta_{x_g y_l}) & \cos(\theta_{x_g z_l}) \\ \cos(\theta_{y_g x_l}) & \cos(\theta_{y_g y_l}) & \cos(\theta_{y_g z_l}) \\ \cos(\theta_{z_g x_l}) & \cos(\theta_{z_g y_l}) & \cos(\theta_{z_g z_l}) \end{bmatrix} \tag{2}$$

A generic rotation matrix allows to obtain the position of the LOC through three (3) independent rotations of the GLO; this algebraically corresponds to multiplying three (3) rotation matrices, each of which describes the elementary rotation performed [2].
To have significant results, it would be advisable that the axes around which these rotations occur coincide with the functional joint rotation axes: the so called **anatomical reference system** is another usually used reference system, it is characterized by planes approximating the anatomical planes of the human body and whose axes are often used to represent the three (3) translational and rotational joint DOFs [2].

Anatomical components of motion As explained before, the analysis of skeletal kinematics is based on the study of the relative position instant by instant of two rigid bodies that represent two body segments [2].
By supposing two adjacent body segments (i.e., one proximal and the other distal), connected by a rotational joint, each segment is associated with a LOC integrated with it, which makes it possible to find its position in space. In correspondence with the joint, a system of functional axes is defined around which the distal segment performs the relative movements with respect to its proximal segment described below and shown in [Fig. 4].

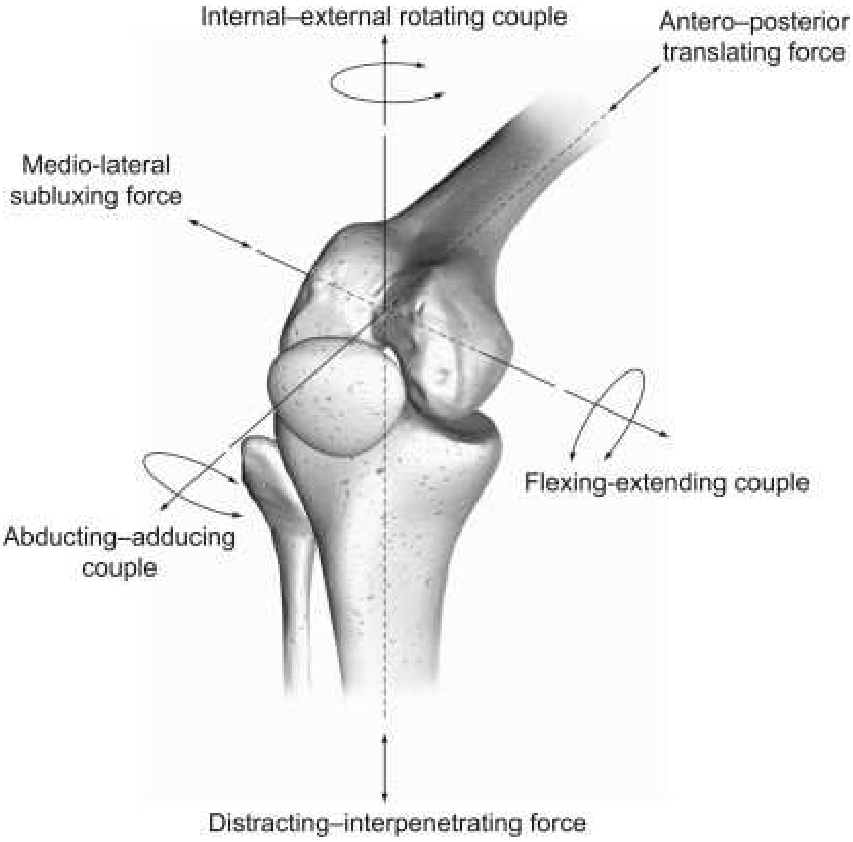

Internal–external rotating couple
Antero–posterior translating force
Medio-lateral subluxing force
Flexing-extending couple
Abducting–adducing couple
Distracting–interpenetrating force

Figure 4: Knee joint angles defined by rotations occurring on the three joint coordinate axes [7].

- The **flexion/extension** movement occurs along the medio-lateral axis of the proximal segment, therefore is perpendicular to its sagittal plane;

- The **adduction/abduction** movement occurs around the antero-posterior axis of the proximal segment, and it is oriented in a direction perpendicular to the longitudinal axis of the distal segment;

- The **internal/external rotation** movement occurs around the longitudinal axis of the distal segment.

In literature, the **joint angles $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$** are used to quantify the extent of these movements, respectively flexion/extension, adduction/abduction, and internal/external rotation.

1.1.3 Dynamics of the musculoskeletal system

A human movement is a complex act resulting from the presence of *internal* (i.e., muscular and articular) and *external* (i.e., gravity, reaction to the ground) forces which presupposes the interaction of different systems in order to maintain balance and to allow the carrying out of motion [2].

Consequently, it is necessary to complete the analysis of human motion with the calculation of dynamic variables such as moments and joint powers: this can be done through the solution of the **Dynamic Problem**. In particular, it allows in its Direct approach to obtain the resulting movement (i.e., displacements and rotations) from the application of a system of external forces, and in its Inverse approach to calculate the forces associated with a certain movement. In motion analysis, the Inverse dynamic approach is the most used one, which starts from the knowledge of kinematic variables and, through algebraic nonlinear equations, calculates forces and moments.

By the assumption of attributing the movement of an anatomical segment to its bone component, and modeling it as a rigid body, also the inertial properties (i.e., mass and moments of inertia) of the entire anatomical part can be attributed to it. In this approach, the entire human body is schematized as a chain of rigid segments connected by ideal friction-free joints located at the level of joint centers. If a part of the structure is considered, which is ideally isolated from the rest of the body at point O, generating at its extreme forces and moments that the rest of the body exerted on it.

By assuming that these segments are subject in a certain instant to a linear acceleration a on the center of gravity, an angular acceleration α and an external force F_e, the dynamic equilibrium equations referred to point O are the following:

$$\frac{\mathrm{d}\Gamma}{\mathrm{d}t} = \sum M$$

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = \sum F$$

The dynamic problem can be traced back to a *static problem* by considering the forces and moments of inertia as external forces and moments applied to the system. In this way, the equations of static equilibrium are:

$$\sum M = 0$$

$$\sum F = 0$$

The solution of these equations presupposes the knowledge of anthropometric, kinematic, and inertial data (i.e., dimensions, mass, position of the center of gravity and moments of inertia of each segment): part of the anthropometric data is collected during the preliminary stages of the acquisition, the rest are calculated with the use of data and tables. In particular, the kinematic variables (i.e., the position of the markers and the estimation of the joint centers) are calculated following the acquisition of evidence using motion capture systems; this will be explained in following paragraphs.

1.2 Gait Analysis

Gait analysis is a subfield of human motion analysis that specifically focuses on the study of walking and running.
Walking is a common and fundamental form of human locomotion: it involves a rhythmic and alternating movement of the legs, arms, and trunk, as well as the coordinated use of various muscle groups [8]. The goal of normal human ambulation is to facilitate travel from one location to another while minimizing effort and maintaining adequate stability across a wide variety of walking conditions [9]. The musculoskeletal system, which is managed by the nervous system, interacts in a highly coordinated, complex manner with the bones, muscles, ligaments, and joints to enable human movement [10]. Furthermore, the locomotor system receives instructions from the central nervous system through the peripheral neural system; the latter is able to transfer the forces required for the body to move and carry out all daily tasks [10].

The aim of gait analysis is to understand the normal patterns of human walking and running and to identify any deviations from these patterns that may indicate the presence of a problem, such as a gait disorder, injury, or a medical condition.

1.2.1 Gait cycle

As explained above, normal human ambulation is designed to make it easier to move from one point to another while exerting the least amount of effort and keeping enough stability under a number of different walking circumstances.
The following sequence [11] can be used to summarize the action of walking:

1. Registration and activation of the gait command inside the central nervous system;
2. Transmission of gait signals to the peripheral nervous system;
3. Muscular contraction;
4. Creation of forces at joints;
5. Control by skeletal segments of joint moments and forces;
6. Generation of ground reaction forces (GRFs).

The most basic method to analyze the walking is to divide the entire cycle into two parts, the initial part defined *stance phase* and the subsequent *swing phase* [Fig. 5].

The stance phase is the part of the gait cycle when the reference limb is in contact with the ground, and accounts for approximately the first 60% of the gait cycle during normal walking. The swing phase begins when the reference limb is not in contact with the ground. This phase covers up roughly the final 40% of the gait cycle during normal walking. Duration of stance and swing phases are used as common temporal measures of each phases' length of the gait cycle [9].

By convention, the starting point of the gait cycle, defined as the *initial contact* (IC, 0% of the gait cycle) occurs at the instant in time when the foot first contacts the ground: in healthy walking the heel makes the first contact, and traditional methods describing the gait cycle refer to this event as *heel strike* (HS) [9].

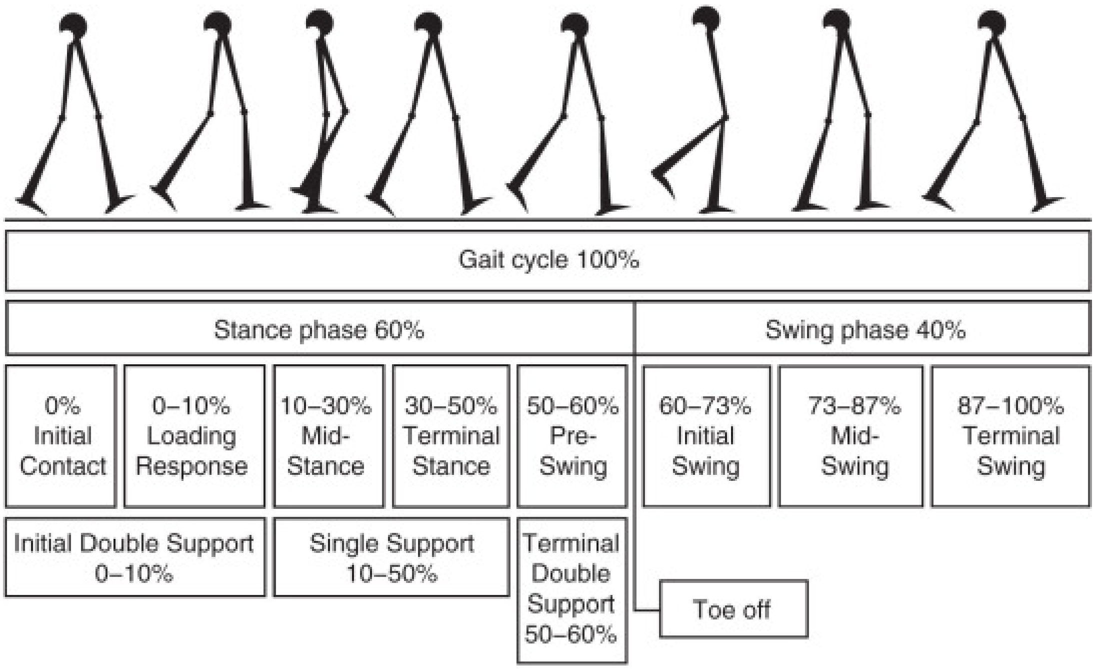


Figure 5: Normal gait cycle periods and timing [9].

The end of stance phase and the beginning of swing phase are defined by *toe-off* (TO) as the foot leaves contact with the ground. The end of the current gait cycle (100% of gait) is determined by the initial contact of the same limb.

Stance phase As previously explained, the point at which the foot comes in contact with the ground is called the initial contact (IC), that typically correspond to the HS in healthy walking; this event is served as the starting point of stance phase and the entire gait cycle (0% of gait). [9].

- **Loading response** (0% - 10% of gait), the period that starts at IC and lasts until the contra-lateral foot leaves the ground, is a double limb support phase which the impact of initial contact is adsorbed and weight is transferred quickly onto the leading limb;

- **Midstance** (10% - 30% of gait) begins when the contra-lateral foot leaves the ground and lasts until the ipsilateral heel leaves the ground. During this phase, the body weight moves forward to be aligned over the foot in contact with the ground;

- **Terminal stance** (30% - 50% of gait) starts when the ipsilateral heel leaves the ground and ends at the time of the contra-lateral foot initial contact with the ground. During this phase, the body weight continues its progress such that the heel rises as weight moves over the forefoot;

- **Preswing** (50% - 60% of gait) occurs, it is the final phase of stance and lasts from the instant of contra-lateral foot IC until the ipsilateral foot TO.

Swing phase

- **Initial swing** (60% - 73% of gait), the first part of swing, represents the time from when the foot leaves the ground to ipsilateral foot alignment with the contra-lateral ankle;

- **Midswing** (73% - 87% of gait) occurs from ankle and foot alignment instant of time to the swing leg tibia becomes vertical;

- **Terminal swing** (87% - 100% of gait), the final portion of swing phase, takes place from the time the tibia reaches the vertical position until HS of the swing foot with the ground [12]. During this, the reference limb completes its forward movement.

1.2.2 Muscle control during the gait cycle

Understanding how muscles behave during the gait is very important to assess normal and pathological gait; in fact, during walking, the muscles in the legs and feet work together to produce movement [13].
The primary muscles involved in walking, depicted in a simplified version in [Fig. 6] include:

- **Gluteus maximus**: this muscle is the largest and most powerful muscle in the buttocks. It works to extend the hip joint during the stance phase of walking, which helps to propel the body forward. As the heel strikes the ground, the gluteus maximus contracts eccentrically to control the forward motion of the body, and then contracts concentrically to extend the hip and push the body forward. The gluteus maximus also plays an important role in stabilizing the pelvis during single-leg stance;

- **Gluteus medius**: This muscle is located in the side of the hip and plays an important role in stabilizing the pelvis during single-leg stance. As the weight of the body shifts over the stance leg, the gluteus medius contracts concentrically to prevent the opposite hip from dropping. This muscle also helps to control the internal and external rotation of the hip during walking;

- **Quadriceps**: the quadriceps muscles are located in the front of the thigh and consist of four muscles: *rectus femoris, vastus lateralis, vastus medialis, and vastus intermedius.* These muscles work together to extend the knee joint during the stance phase of walking. As the foot strikes the ground, the quadriceps muscles contract eccentrically to control the flexion of the knee, and then contract concentrically to extend the knee and push the body forward;

- **Hamstrings**: hamstrings are a group of three muscles located in the back of the thigh: *biceps femoris, semitendinosus, and semimembranosus.* These muscles work to flex the knee joint during the swing phase of walking and also play a role in slowing down the forward movement of the leg during the stance phase. As the foot leaves the ground, the hamstrings contract eccentrically to control the extension of the hip and knee, and then contract concentrically to flex the knee and prepare the leg for the next step;

- **Gastrocnemius and soleus**: these muscles are located in the calf and work together to plantarflex the ankle joint during the push-off phase of walking. As the foot pushes off the ground, the gastrocnemius and soleus contract concentrically to lift the body up and propel it forward. These muscles also play an important role in absorbing shock during the stance phase of walking;

- **Tibialis anterior**: this muscle is located in the front of the shin and is responsible for dorsiflexion of the ankle joint during the swing phase of walking. As the leg swings forward, the tibialis anterior contracts concentrically to lift the foot and prevent it from dragging on the ground. This muscle also helps to control the foot during the stance phase of walking by eccentrically controlling the plantarflexion of the ankle.

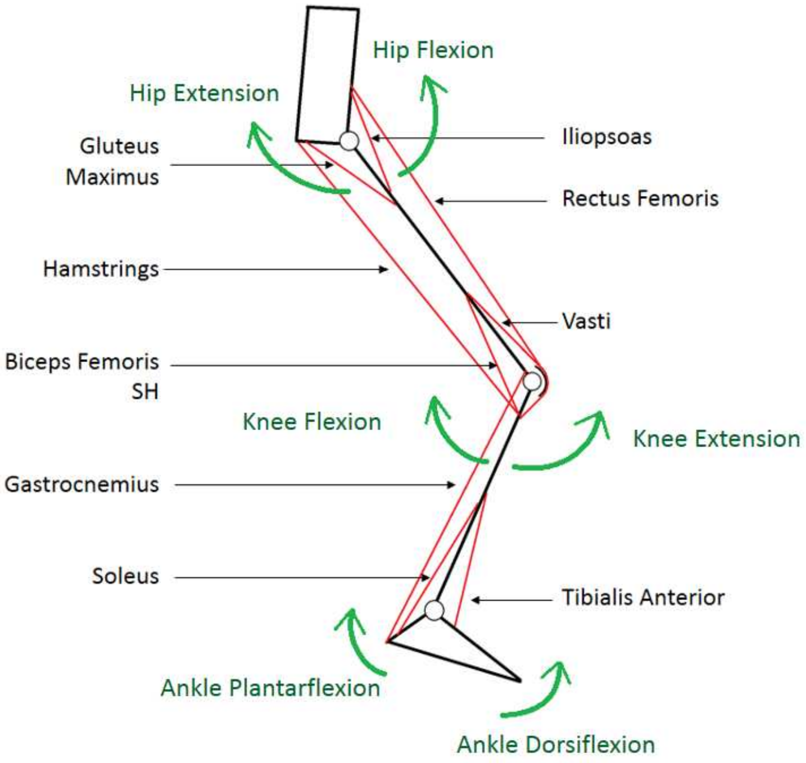


Figure 6: Different motion and simplified muscle groups of the lower limb [14].

1.2.3 Space-time parameters

During the gait analysis, some *space-time parameters* can be evaluated to characterize quantitatively the gait of each subject; in particular, these can be obtained by recording the task by means of different devices, through which some events of the gait (i.e., HS and TO) must be identifiable [2].
The main temporal parameters are:

1. **Stride duration [s]**: the temporal interval between the two heel-contacts of the same foot;

2. **Stance duration [s]**: the time interval between HS and TO of the same foot;

3. **Swing duration [s]**: the time interval between TO and the next HS of the same foot;

4. **Step duration [s]**: the time between ipsilateral and contralateral heel contact;

5. **Single support duration [s]**: the time in mono-podalic support, and begins when the ipsilateral foot comes off the swing phase;

6. **Double support duration [s]**: the time in bi-podalic support, and the support is on both feet;

7. **Cadence [step/min]**: the number of steps at the minute.

Similarly, it is possible to evaluate spatial and space-time parameters:

1. **Step length [m]**: the distance between a point of contact of one foot with the ground (i.e., the heel) and the same point of contact with the ground of the other foot;

2. **Speed of the step [m/s]**: the linear velocity along walking direction measure on one or more steps.

Using these parameters, it is possible to evaluate the differences between a healthy and a pathological subject; in fact a gait alteration can characterize the presence of a pathology or motor dysfunction.

1.3 Motion capture instrumentation

Since motion analysis in recent years has been validated as a valid technique in the study of various movements including walking, several tools are now available to quantitatively measure the path, which can be used both alone and in combination to get different information and a complete picture of the motion under investigation.
In particular, the instrumentation used in gait analysis typically includes:

1. **Motion capture (MoCap) systems**: These systems use *cameras and markers* placed on various parts of the body to track the movement of those body parts during gait;

2. **Force plates (FP)**: These plates are embedded in the floor and measure the forces generated by the feet during walking;

3. **Wearable instrumentation**: such as *electromyography (EMG)*, *pressure sensors*, *inertial measurement units (IMUs)*, and others. These provide a non-invasive way to monitor movement and muscle activity, and can be used in a variety of applications.

Together, these instruments provide a detailed analysis of gait, which can be used to diagnose and treat a variety of conditions related to movement and posture.

A brief explanation of the instrumentation used in this work will be present in the following paragraphs.

1.3.1 MoCap system

Motion capture system allows the simultaneous analysis of multiple parameters and measurements controlled in real time by specialized staff; it is possible to perform studies with high repeatability and precision [2]. Moreover, it is a non-invasive system and it is not limited regarding energy.
However, MoCap also has negative aspects, such as the cost of the instrumentation and the inability to study the subject's daily activities outside the laboratory.

Stereophotogrammetric systems *Stereophotogrammetry* is a highly sophisticated technique used in biomechanics to measure the movement of the human body [15].
This method uses multiple photographs of a subject from different angles and applies specialized software to create a 3D model of the subject's body. Moreover, the system involves the use of **markers** placed in anatomical points defined by appropriate protocols to make the studies comparable to others. By means of two or more cameras, it is possible to reconstruct the position of each marker in three dimensions at any instant of time; then, the software is able to evaluate displacement, speed, acceleration, and joint angles of each body segment [15]. [Fig. 7] below shown the study of gait by means of this system:

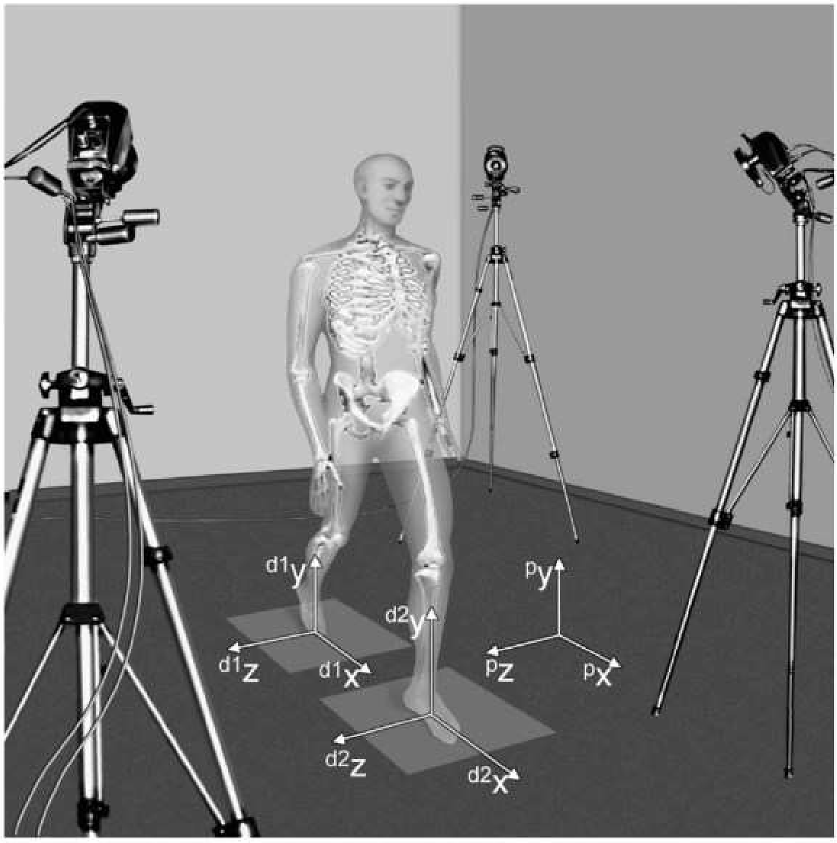


Figure 7: Gait analysis by means of stereophotogrammetry [15].

The positive aspect of this technique lies in the fact that the measurements are reliable, as they come from a video-recorded analysis processed with appropriate software, which makes this a **gold standard tool** for motion analysis [15].

However, despite its potential benefits, stereophotogrammetry has some limitations: the technique can be time-consuming and expensive, and requires specialized equipment and expertise to implement effectively [15]. In addition, an accurate calibration is also required for the quality of the measured data: firstly, it is necessary to carry out a calibration procedure to define the calibration volume, the absolute reference system (origin and axes of the laboratory system), the internal and external parameters of the cameras [2]. Furthermore, it is also limited by the fact that it captures only the external movement of the body, and does not provide information about the internal musculoskeletal system [15].

Although these tools guarantee a high accuracy in the measurement of the coordinates of the markers, the kinematic variables relating to the joints obtained from them can be affected by various types of errors; these can be classified into:

- **Instrumental errors**: these can be *systematic* (i.e., an inaccurate calibration), or *random* (i.e., electronic noise);
- **Incorrect placement of markers on the anatomical landmarks**: identifying the landmark can be complicated and requires specialized and experienced staff;
- **Soft tissue artifacts (STA)**: caused by sliding of the skin, deformations, movement of the underlying muscles, and the presence of adipose tissue.

Gait protocols In order to reduce as much as possible the errors introduced by the misplacement of markers on landmarks different protocols for motion analysis have been developed; these aim to create a standardized and repeatable method of carrying out the analyzes, positioning of the markers, calculating the variables of interest and the methods of presenting the results obtained [2].
In general, a *protocol* uniquely define the anatomical landmarks where apply the markers to follow a predisposed biomechanical model; define the anatomical structures of reference by identifying the correspondence between the position of the markers and the underlying body segment; define the procedures from data acquisition, but also for data processing and analysis; finally, make kinematic and dynamic parameters clinically interpretable [2].

Over the years, many protocols for motion analysis have been developed depending on the specifics of analysis and use.
The most important are:

- **Davis protocol**: this protocol involves a comprehensive assessment of a patient's gait, including a physical exam, a review of medical history, and various tests and measurements;
- **CAST protocol**: also known as the *"Calgary Cambridge Foot and Ankle Protocol"*, this protocol is designed to provide a comprehensive assessment of foot and ankle function during gait, with a focus on identifying the underlying causes of foot and ankle pain and dysfunction;
- **IOR Gait protocol**: also known as the *"Istituto Ortopedico Rizzoli Gait Protocol"*, this protocol is designed to provide a comprehensive assessment of gait in patients with a wide range of orthopedic and neurological conditions, including cerebral palsy, spina bifida, and osteoarthritis.

Force plates Force plates are a type of sensor used in biomechanics to measure the forces exerted by a person's body during activities such as walking, running, or jumping [2].
These consist of a flat surface that is sensitive to pressure and a set of sensors that detect the forces acting on the surface; moreover, these are able to measure the three (3) X, Y, and Z components of the forces and the moments associated with them, making it possible subsequently to estimate the magnitude, direction, and point of application (*center of pressure, CoP*) of the forces [2].

Sensors integrated in these platforms are able to collect forces thanks to load cells consisting of:

- **Resistive strain gauges sensors**: with this technology, force plates have four (4) load cells able to vary their resistance according to the deformation to which are subjected. The relation resistance/deformation is shown below:

  $$R = \rho * (\frac{L}{A})$$

  where R is the resistance, ρ is the resistivity of the material, L is the length of the strain gauge, and A is the constant section of it. If the strain gauge is fed with a current of intensity I, it is able to produce a voltage:

  $$V = R * I$$

  which represents an electrical signal that can be digitized and subsequently converted into the corresponding applied force value;

- **Piezoelectric sensors**: here, force plates have three (3) load cells containing a *piezoelectric crystal* which is able to behave like a capacitor to which a potential difference is applied to. In this way, if the two faces of the force plate are connected through an external circuit, an electric current called piezoelectric current is generated, and can be digitized to obtain the corresponding force value.

Whatever technology is used, each cell collects the force applied along the three (3) axes of a predefined reference system, and returns a result in three (3) dimensions [2].
By using the *statics equations*, it is possible to obtain the point of application and the modulus of the final resulting force in the three (3) spatial directions.

The coordinate system of the platform is determined by the alignment of the resistive or piezoelectric sensors on each pylon, which are aligned with the x, y and z axis of the platform, as shown in [Fig. 8].

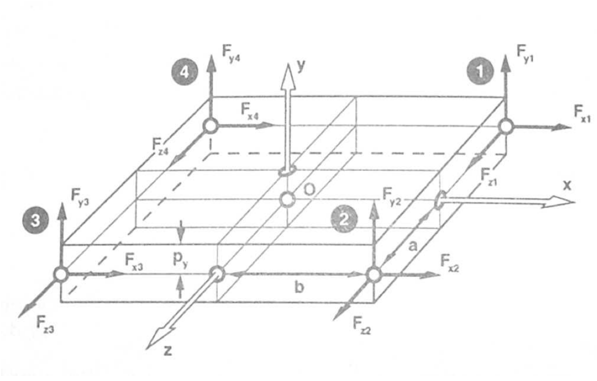


Figure 8: The coordinate system of a force platform and its sensors [2].

Ground reaction forces (GRFs) A ground reaction force (GRF) is a force that acts on a body as a result of the body weight resting or hitting the ground.
For instance, by considering a person standing on the ground the person is exerting a force on the ground, but for the **principle of action and reaction** also the ground is exerting an equal and opposite reaction force on the person; this reaction force is known as the GRF [16].

In motion analysis, GRFs are studied with respect to the different directions and events of the stance; in particular, these can be broken down into three components:

1. **Anterior/posterior component**: this refers to the force that is exerted in the front-to-back direction (i.e., X axis). When walking forward, for instance, the ground exerts an anterior force on the foot, which helps to propel forward; vice-versa, when braking to a stop, the ground exerts a posterior force on your foot which helps to slow down;

2. **Vertical component**: this refers to the force that is exerted in the up-down direction (i.e., Y axes). When jumping, for instance, the ground exerts an upward force on foot which helps to lift off the ground; vice-versa, when landing after the jump, the ground exerts a downward force on foot which helps to absorb the impact;

3. **Medio/lateral component**: this refers to the force that is exerted in the side-to-side direction (i.e., Z axis). When changing directions during a run, for instance, the ground exerts a medio-lateral force on foot, which helps to move in the desired direction.

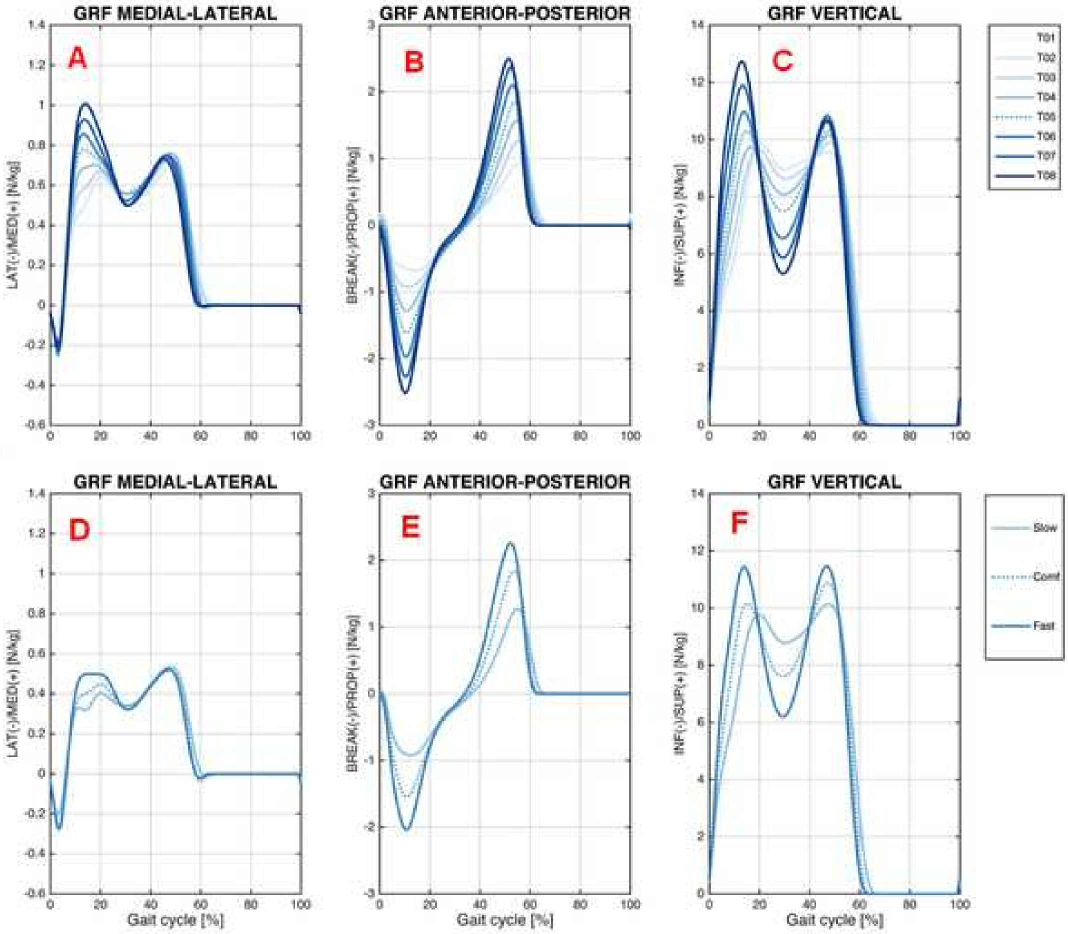


Figure 9: Ensemble averages of the ground reaction force (GRF) on the treadmill (GRF medial-lateral (A), GRF anterior-posterior (B), and GRF vertical (C)); and overground (GRF medial-lateral (D), GRF anterior-posterior (E), and GRF vertical (F)) walking conditions [17].

[Fig. 9] shows an example of GRFs data for the medial/lateral, anterior/posterior, and vertical direction collected by healthy subjects during two different tasks, the treadmill and a normal overground walk, at various speeds [17].

1.4 Inertial measurement units (IMUs)

Inertial measurement units (IMUs) have become increasingly popular in recent years for human motion analysis due to their small size, low cost, and portability.
IMUs are typically composed of miniaturized electronic devices that come with a full set of *micro electro-mechanical systems* (MEMS) sensors to determine the change in relative orientation of a body or body segment over time [18].
These sensors include:

- Accelerometers
- Magnetometers
- Gyroscopes

By combining the measurements from these sensors, an IMU can provide information on the movement, orientation, and position of the sensor in 3D space [18].

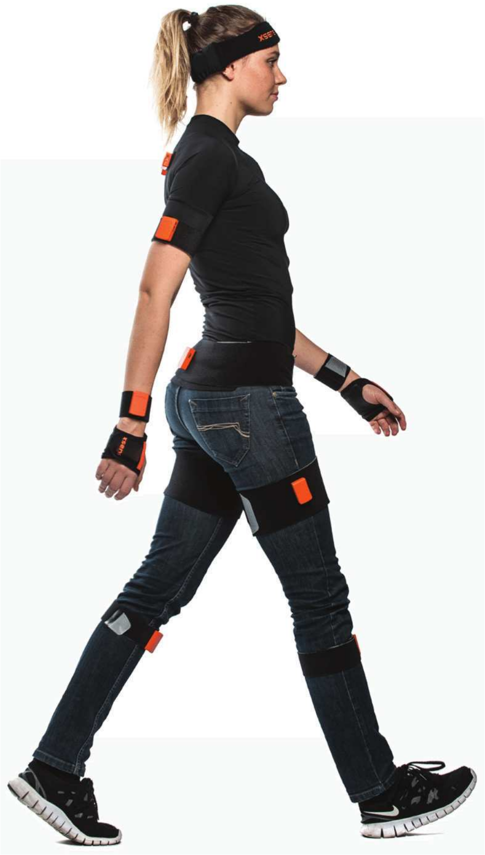


Figure 10: IMUs for biomechanical applications have been incorporated into MoCap suits (i.e., the XSENS MVM suit) [19].

One of the main advantages of IMUs is their ability to capture motion in real-time and in a variety of environments: unlike traditional motion capture systems, IMUs are less affected by occlusion and can still capture motion even when body parts are obstructed from view. Additionally, IMUs can be attached directly to the body or clothing [Fig. 10], which allows for a more natural and unrestricted range of motion during activities. This makes IMUs particularly useful in sports science and biomechanics research, where athletes can perform their usual activities without being restricted by wires or markers [18].

A brief overview of IMUs sensor technology is shown below.

Accelerometers Accelerometers are based on different technologies, but the operating principle is the same: these are based on the detection of the inertia of a mass subjected to acceleration. The mass is connected to an elastic element and its position over time is acquired by a sensor. In the presence of an acceleration the mass, which has its own inertia, moves from its rest position in proportion to the acceleration. The sensor transforms this displacement into a digitizable electrical signal [2]. Accelerometers can be classified according to the operating principle of the position sensor:

- Strain gauge accelerometer
- Piezoresistive and piezoelectric accelerometer
- Linear Variable Differential Transformer (LVDT) accelerometer
- Capacitive accelerometer

Generally, accelerometers are monoaxial; to have a triaxial accelerometer is necessary to arrange 3 monoaxial units in order to form a cartesian triad.

Magnetometers The magnetometer is an instrument that measures the local magnetic field along three predetermined axes [2]. The earth's magnetic field has a component parallel to the earth's surface that always points towards magnetic north, consequently its projection on the horizontal plane can be used to determine the orientation of a magnetometer in space. The design of many magnetometers integrates three magnetic sensors, whose axes form a Cartesian triad. From the projections along the axes, it is possible to trace the horizontal component of the earth's magnetic field, as long as the inclination of the sensor's case is estimated using a gravity sensor.

In motion analysis, the measurement of the earth's magnetic field is necessary to allow monitoring in the three dimensions of the <u>orientation</u> of the sensors; without, it would not be possible to identify the local reference system of the sensors in the global reference system of the earth, in which the measurements are analyzed.

Gyroscopes A gyroscope is a rotating physical device in which, due to the *conservation law of angular momentum*, tends to keep its rotation axis oriented in a fixed direction, providing direct measures of the angular velocity in the local coordinate system of the sensors in the three planes [18].

In [Fig. 11] the principal elements of a gyroscope are displayed: essentially, it consists of a toroid-shaped rotor that rotates around its axis, when the rotor is rotating its axis tends to remain parallel to itself and to oppose any attempt to change its orientation.

In the biomechanical field, gyroscopes can be applied to any part of the body, and thanks to their portability they are often used for recording walking for long periods: for instance, from the measurement of the angular velocity, useful data can be obtained for clinical and sporting purposes, such as the length of the walking cycle, the length of the stride, the stance and swing time, useful for both clinical and sporting purposes.

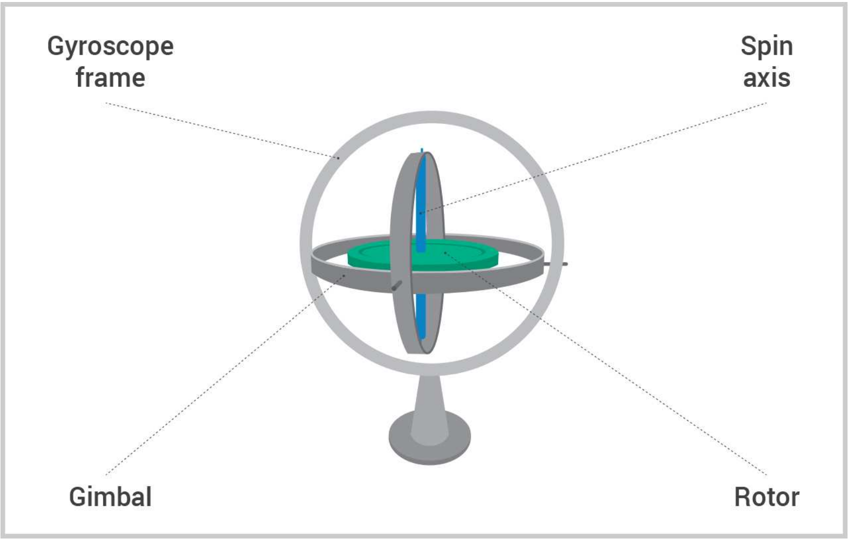


Figure 11: Elements of a gyroscope [20].

Despite all the benefits of using IMUs, there are also limitations for human motion analysis.

Firstly, one limitation is the potential for sensor drift or error accumulation over time, which can lead to inaccurate measurements. Sensor drift occurs when the sensors gradually become misaligned, resulting in a gradual loss of accuracy over time. To avoid this, researchers have developed algorithms for sensor fusion and calibration to improve the accuracy of IMU measurements and reduce the effects of sensor drift; in particular, these algorithms combine measurements from multiple sensors to improve the overall accuracy of the IMU data [18].

Another limitation of IMUs is their difficulty in distinguishing between different types of motion using only IMU data. For example, it can be difficult to distinguish between walking, running, or jumping based solely on IMU data. To address this limitation, researchers have explored combining IMU data with other types of sensors, such as electromyography (EMG) or force sensors, to provide a more comprehensive analysis of human motion [18].

1.5 Musculoskeletal modeling

Musculoskeletal models for biomechanical simulations have become increasingly popular for analyzing human movement. In addition to joint kinematics and kinetics, musculoskeletal models allow researchers to evaluate other biomechanical variables that cannot be experimentally measured *in vivo*, such as muscle lengths and forces. Moreover, these models can be scaled and customized to make the musculoskeletal model subject-specific [2].

Hypothesis In this approach, the entire human body is described by a *multi-body articulated system*: the inertial characteristics of the masses are known, the individual anatomical segments are approximated to rigid bodies rotating about a joint axis and are identified by means of anatomical landmarks. Moreover, the masses of the segments are assumed to be concentrated on the center of mass of the segments, and the lengths of these segments are defined as distances between anatomical landmarks [21].

Various software have been developed to model and analyze human movement (i.e., *AnyBody, OpenSim, and Human Body Model*), and there is always an increase in literature reporting motion analyzes based on these software systems.
This paragraph will cover an overview of the functionality of the software used in this project, focusing on the main steps necessary to create and analyze dynamic simulations of movement and underlining the main workflow when using MoCap and IMUs data.

1.5.1 OpenSim

OpenSim is an open-source software that allows modeling, analysis and simulation of the locomotor system: before the birth of this, it was very difficult to perform simulations outside the laboratory where they were developed, the results were therefore difficult to reproduce, and researchers were forced to spend a lot of time to implement new simulations and the related tools to analyze them.
The first version of the software was developed in 2007 by the *American Society of Biomechanics Conference*, to which an Application Programming Interface (API) was added in the next version, to allow users to have access to the program components and modify them.

With the development of these new software various musculoskeletal systems were born, in particular for the lower limbs, through which various parameters can be estimated (i.e., the length of muscle fibers, speed, acceleration) in both physiological and pathological conditions.

Musculoskeletal models To create a motion simulation, first of all the definition of a dynamic model of the musculoskeletal system and its interactions with the environment is necessary.
The elements of the musculoskeletal system are modeled by differential equations describing the dynamics of muscle contraction, the geometry of the musculoskeletal system and the dynamics of body segments [21]. These equations characterize the time-dependent behavior of the musculoskeletal system in response to neuro-muscular excitation.
The body segments, such as trunk, pelvis, and femur, require the definition of body segment parameters (i.e., the mass, inertia, and center of mass location): in the model, body segments are connected to each other by joints, which describe the degrees of freedom (DOF) or relative motion of the segments with respect to each other.

Consequently, an OpenSim musculoskeletal model consists of:

- **Bodies**: rigid segments representing bone components or other elements of the model like platforms;
- **Joints**: connect the bodies to each other;
- **Forces**: the muscles that extend between the bodies, but also forces estimated by contact geometries;
- **Constraints**: to lock degrees of freedom of the model.

The OpenSim library provides musculoskeletal models already developed, which are mainly divided into models representing the lower limbs and upper limb. In addition, these differ from each other in the presence of one limb or both, in the number of degrees of freedom available and in the muscles represented.

Once the dynamic model of the musculoskeletal system has been formulated, the OpenSim software guides users to create a dynamic simulation through these main steps.

a. Scaling The first step is to scale the generic model in order to attribute to the model the *mass and anthropometric characteristics* of a particular subject, from which the experimental data were collected.

The scaling process changes the dimensions of the bodies present in the model, the masses, the tensor of the moments of inertia and the lengths of the muscle fibers and tendons, while it does not modify the characteristic curves or the maximum isometric forces of the muscles [2].

There are two techniques for scaling the size of model bodies:

- **Measurement-based scaling**: the scale factor is determined by comparing the distance between the markers and the model with the position of the experimental markers present on the subject;
- **Manual scaling**: some predetermined scaling factors are used, in case of no static acquisition of the subject with markers available.

Thus, the masses of the bodies are scaled so that the total mass equals the specific mass of the subject.

The two methods used for scaling the masses are to **preserve the mass distribution**, and based on **scale factors**; in any case, the inertia tensor is recalculated based on the new dimensions and masses.

The scaling operation in OpenSim is carried out through the **Scale Tool**: firstly, it is necessary to set a *Subject_markerSet.xml*, which contains a set of virtual marker position coordinates (x, y, z) representing exactly the markers that are applied on the subject during the experimental test; as input, the tool requires a *Subject_static.trc* file which contains the trajectories of the experimental markers obtained by following a static test (i.e., a test in which the subject remain still for few seconds in a known static position), and the musculoskeletal model chosen for the simulation *Subject_model.osim*.
As output, the tool returns a new model which is the one used as input but scaled in relation to the subject simulation that specific motion furnished as input in the .trc file.
A schematic representation of the process just described is shown in [Fig. 12].

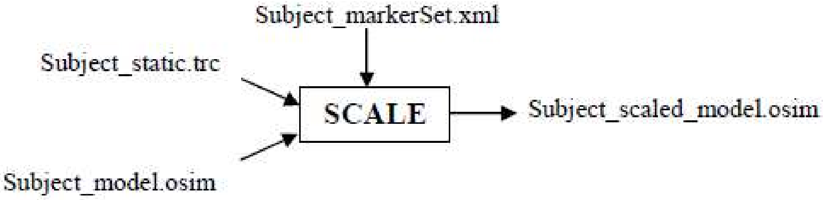


Figure 12: Scaling Tool operating workflow [22].

b. Inverse Kinematics (IK) With the term *"kinematics"* we refer to the study of the motion of an object without considering the forces and the moments that are produced during the entire motion [2].
In particular, *Inverse kinematics* is the process of determining the parameters of a subject that satisfy the achievement of a desired movement; in the case of biomechanical simulations, this procedure is applied to make the musculoskeletal model perform a determined movement.
In motion analysis, the purpose of inverse kinematics is to find a set of generalized coordinates (i.e., the joint angles and the positions of the body segments) for the musculoskeletal model that best reproduces the kinematics of a subject's movement.

OpenSim determines this match by solving a *weighted least squares optimization problem* with the goal of minimizing the marker error: each marker has an associated weight that specifies how strongly that marker error term should be minimized in the least squares problem [22].
The algorithm calculates the quadratic error:

$$err = \sum_{i=1}^{N} w_i (x_i^{exp} - x_i^{mod}(q))^2 + \sum_{j=1}^{M} w_j (q_j^{exp} - q_j^{mod}(q))^2 \qquad (3)$$

where N is the number of markers, M is the number of model DOFs, w_i is the i-th marker weight, w_j is the j-th coordinate weight, x_i^{exp} is the i-th experimental marker position, x_i^{mod} is the i-th virtual marker position (which depends on the coordinate value q), q_j^{exp} is the j-th experimental coordinate value, and q_j^{mod} is the j-th virtual coordinate value. Finally, this error is minimized for all possible coordinates q:

$$min_q(err)$$

The result obtained from this minimization is a set of coordinates, which over time describe the position of the virtual markers and the joint angles of the model, in order to follow the real movement of the subject [22].

The tool used in OpenSim through the inverse kinematics approach is called **InverseKinematics Tool**: by entering the appropriately scaled model and a .trc file containing the trajectories of the experimental markers (*Subject_motion.trc*), the tool gives as output a .mot file *Subject_motion_ik.mot* which contains the value of the angle at each instant for each DOFs of the system.
A schematic representation of the process just described is shown in [Fig. 13].

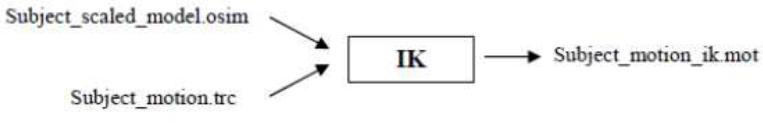


Figure 13: InverseKinematics Tool operating workflow [22].

c. Inverse dynamics (ID) *Inverse dynamics* operates the joint angles, angular velocities, and angular accelerations of the model, along with the external forces and moments applied to resolve the reaction forces and moments in each joint [22].
To determine these, the *equations of motion* are solved iteratively to satisfy the condition of dynamic equilibrium:

$$\tau = M(q)\ddot{q} - C(q,\dot{q}) - G(q) - F \qquad (4)$$

where N are the DOFs of the system, $q, \dot{q}, \ddot{q}$ are vectors containing respectively positions, velocities, and accelerations, $M(q)$ is the matrix of masses of the system, $C(q,\dot{q})$ is the vector of centrifugal and Coriolis forces, $G(q)$ is the vector of gravitational forces, F is the vector of ground reaction forces, and τ is the vector of unknown generalized joint forces.

The tool used in OpenSim is called **InverseDynamics Tool**: it requires the musculoskeletal model, the kinematics of movement $Subject01\_walk1\_ik.mot$ (which can be filtered, if necessary), the initial and final instants of the simulation, and optionally the external loads file $Subject01\_walk1\_grf.xml$ applied to the model.
The tool returns a file $Subject01\_walk1\_InverseDynamics\_force.sto$, containing for each degree of freedom the value of the moment associated with that articulation for each instant of time.
A schematic representation of the process just described is shown in [Fig. 14].

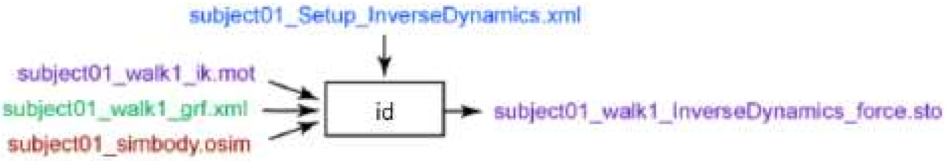

subject01_Setup_InverseDynamics.xml
subject01_walk1_ik.mot
subject01_walk1_grf.xml
subject01_simbody.osim
id
subject01_walk1_InverseDynamics_force.sto

Figure 14: InverseDynamics Tool operating workflow [22].

In this project, these three main functions of OpenSim have been used and briefly introduced; however, the software has other functions, such as *Static Optimization Tool*, *Computed Muscle Control Tool*, and others, specific to each type of task and study in the area of motion analysis.

In addition, in recent versions of the software, tools for placing IMUs and calculating the InverseKinematics using data from them have been implemented; in particular, these two tools will be explained in the following sections.

1.6 Ground reaction forces estimation methods

For functional evaluation and biomechanics investigations, measuring three-dimensional ground reaction forces (GRFs), moments (GRMs), and centers of pressure (CoP), as well as other biomechanical parameters, is a topic of significant interest.

Gait Analysis calls for the measurement of walking kinematics and its boundary conditions, which are represented by GRF, GRM, and CoP [23]. Using an optoelectronic system (OS) along with two or more floor-mounted force plates (FP) represents the state of art for measuring biomechanical parameters in popular activities, such walking or running. These systems are simple to combine with additional acquisition tools like electromyography, video recording, or force sensors, supplying trustworthy data sets for a multifactorial, integrated functional evaluation [23].

However, despite its high reliability, repeatability, and accuracy, the use of an OS has several drawbacks:

1. Due to its complexity, it needs dedicated spaces and a controlled environment, provided for instance by a motion analysis lab;
2. Consequently, measurements of open-field tasks are not permitted;
3. The OS system is expensive and requires specialized staff to use it;
4. Finally, it was noted that subjects might alter their gait when walking on FP or treadmill [24].

In order to overcome (1.) and (2.), to measure GRFs, GRMs, and CoP several methods were developed by using wearable sensors: examples of applications were exposed by Razak *et al.* in their work [25].

Generally, methods based on wearable sensors can be classified into these categories [1]:

- Methods based on **matrix and/or pressure sensors** used as insoles;
- Methods based on **wearable load cells** for direct 3D GRF measurements;
- Methods based on the **kinematic data** obtained by OS;
- Methods based on **inertial measurement units (IMUs)**, that measure motion of body segments and estimate GRFs by using a biomechanical model and/or machine learning approaches.

Nevertheless, the first three classes of methods go beyond the scope of the present work; additional information about those approaches can be found in Razak *et al.* [25] and Shahabpoor and Pavic *et al.* [26] works.

1.6.1 Methods based on biomechanical modeling

To estimate GRF by using inertial measurements, data modeling is needed; in particular, most of the methods proposed in literature are based on inverse dynamics approaches that require biomechanical modeling [1].
With respect to walking and running tasks, a first attempt to record kinematics of a gait cycle outside a motion lab was conducted by Ohtaki *et al.* [27]: here, three inertial units were used placed to distal position of shank and thigh with Velcro-straps, as shown in [Fig. 15], to study only the kinematics of the sagittal plane.

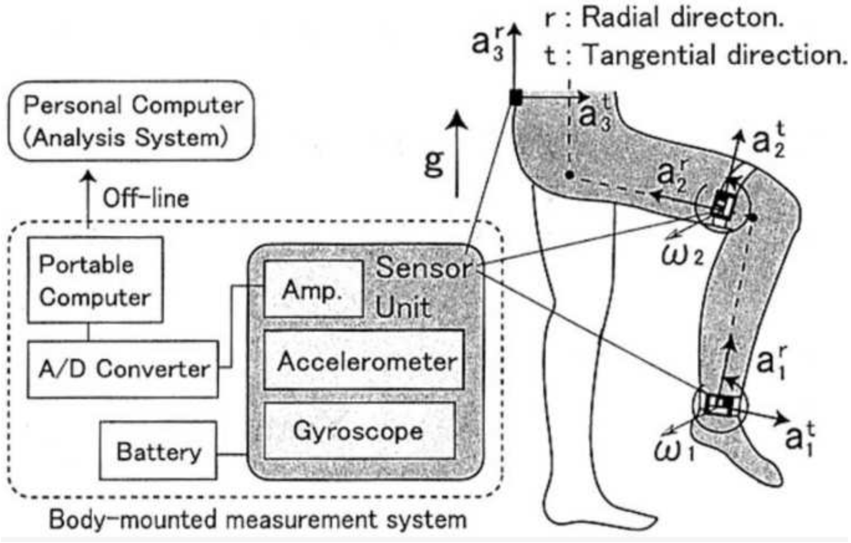


Figure 15: Inertial units structure and placement, and biomechanical model [27].

Furthermore, under the premise of left-right symmetry in normal gait, only the motion of the left leg was collected, and the kinematics of the right leg were acquired; measuring the single leg motion allowed to improve mobility by eliminating sensors on the other leg and reducing power consumption. Conclusively, data was collected on a notebook computer carried in a backpack while subjects completed the task.
After data collection and comparison with data obtained from the gold standard (OS and FP), joint angles were calculated by integrating angular velocities of the segments composing the model, in particular pelvis, thigh, and shanks, as depicted in [Fig. 15].
The total ground reaction force was determined by an inverse dynamic analysis based on the recursive formulation of force and moment balance equations. To conclude, the method presented by Ohtaki *et al.* [27] provided the collection of temporal parameters according to foot movement, kinematic data and GRF in a non-controlled environment, even though some deviations from OS results were found.
However, limitations of this approach were observed, such as a much simplified biomechanical model used with standard anatomical parameters, one-dimensional sensors, and the analysis was limited to the sagittal plane. Additionally, the analysis was limited to the single support phase, since in the in double support phase where both feet are touching the floor, the kinematic chain is indeterminate and equations of motion cannot be solved [26].

Neugebauer *et al.* proposed a different approach in their work [28]. Here, the authors used a statistically based model to estimate only the peak of the vertical component of GRF during walking and running tasks.
Yet, the proposed method still had limitations, mainly due to the sensors used and the biomechanical model.

A more complex approach is the one proposed by Yang *et al.* [29]: the authors designed a method to estimate lower limb forces and moments while walking by using IMUs able to track kinematics data and detailed three-dimensional biomechanical model, depicted in [Fig. 16].

As an advancement, this method aims to measure three-dimensional walking motion; that means kinematics in frontal and transverse planes were considered with sagittal plane.

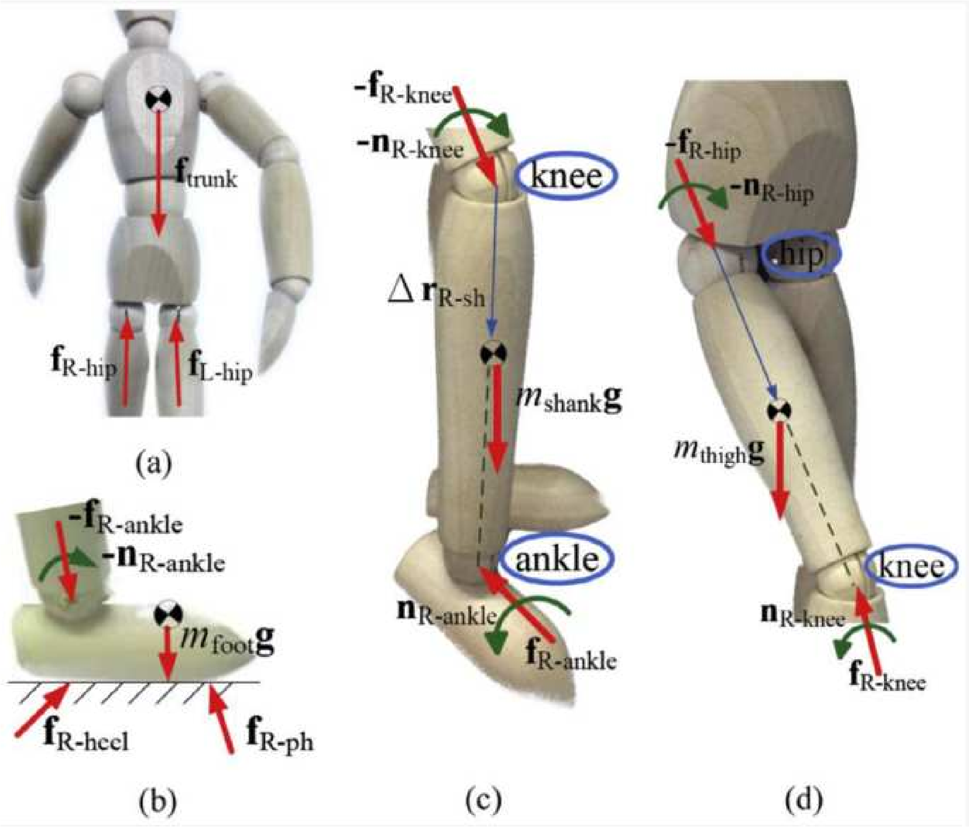

$\mathbf{f}_{\text{trunk}}$
$\mathbf{f}_{\text{R-hip}}$
$\mathbf{f}_{\text{L-hip}}$
(a)
$-\mathbf{f}_{\text{R-ankle}}$
$-\mathbf{n}_{\text{R-ankle}}$
$m_{\text{foot}}\mathbf{g}$
$\mathbf{f}_{\text{R-heel}}$
$\mathbf{f}_{\text{R-ph}}$
(b)
$-\mathbf{f}_{\text{R-knee}}$
$-\mathbf{n}_{\text{R-knee}}$
knee
$\Delta\mathbf{r}_{\text{R-sh}}$
$m_{\text{shank}}\mathbf{g}$
ankle
$\mathbf{n}_{\text{R-ankle}}$
$\mathbf{f}_{\text{R-ankle}}$
(c)
$-\mathbf{f}_{\text{R-hip}}$
$-\mathbf{n}_{\text{R-hip}}$
hip
$m_{\text{thigh}}\mathbf{g}$
knee
$\mathbf{n}_{\text{R-knee}}$
$\mathbf{f}_{\text{R-knee}}$
(d)

Figure 16: Biomechanical model and the respective free body diagrams of: (a) trunk, (b) foot, (c) lower leg, (d) upper leg [29].

This approach was found to be innovative in that it allowed the evaluation of the maximum GRF without the need of FP; nevertheless, the estimation of GRF during double-support phase was still the weakness of Yang *et al.* work, forcing the authors to evaluate it through a statistical approach.

To overcome the indeterminacy problem in the double support phase, Karatsidis *et al.* [30] developed a method based on [29] by adding a distribution algorithm based on a smooth transition assumption function built on empirical data to avoid indeterminate solutions to Newton's equation. Still, this approach presented some limitations, such as poor accuracy due to the fact that the smooth transition assumption was based on data from healthy subjects and may not be applied for other tasks. In addition, the biomechanical model chosen was based on standard anthropometric data, which means other populations such as obese or elderly may not be represented.

In conclusion, this overview of estimation methods of GRFs with IMUs based on biomechanical modeling shown that estimating GRF during double support phase represents the most critical challenge when only relying on kinematic data; moreover, a more advanced anatomical biomechanical model is necessary in order to best estimate kinematic and GRF, and has to be the more suitable possible to represent every scenario, from sport to clinical applications.

1.6.2 Methods based on machine learning

Machine learning represents a modern approach for the estimation of GRF; in fact, these approaches are based on the assumption that there exists a correlation between the acceleration measured somewhere on the human body and the ground reaction forces [1]. In the case of a biomechanical model, this correlation is represented by the biomechanical model itself.

The main advantage represented by these methods is that machine learning does not require an a-priori knowledge of the model, but they build up their model based on *Artificial Neural Network (ANN)* algorithms trained by a large amount of data in order to establish a relationship between input and output quantities [1].

These approaches are not discussed in the present project, as it is focused on the estimation of GRFs by using a method based on a biomechanical model.

1.7 OpenSim Moco

1.7.1 Premise

Musculoskeletal simulations have allowed the study of movement disorders in detail, and are used in many different applications, from the analysis of patients with movement disorders to the design of wearable devices for rehabilitation purposes.
In general, these simulations are categorized by whether the motion under investigation is prescribed from experimental data or completely predicted by the simulation itself. However, tracking motion is a third category, fitting between prescribing and predicting, where deviations between the model kinematics and reference data are part of the cost function instead of being exactly prescribed [31].

Most musculoskeletal simulations are posed as *optimal control problems*; that means these investigate the parameters and time-varying controls of a system that minimize a cost, such as energy consumption, to the system dynamics, expressed as differential-algebraic equations [4]. To solve optimal control problems, biomechanists often use the single shooting method, which involves finding solutions to the initial value problem for different initial conditions until one finds the solution that also satisfies the boundary conditions of the boundary value problem [32].

Yet, more rapid alternatives exist; the direct collocation method avoids the need of time-stepping integration and permits an easily configurable trade-off between accuracy and computational cost compared to the previous method [4].
In direct collocation, states and control of the system are approximated as polynomial splines over a mesh of time points, and an optimizer solves for the knot points that lead the splines to obey the system dynamics [33]. Indeed, the dynamics are enforced by requiring the time derivative of the state splines to match the derivative from the system differential equations at specified time points; in fact, this method is called *direct collocation* because the spline derivatives are *collocated* with the exact derivatives [34].
Further explanations about direct collocation will be provided in the following paragraphs.

The advantages of direct collocation are numerous to lead researchers using this method for prescribing, tracking, and predicting motions; despite of that, very few biomechanics laboratories have been able to apply this technique in their researches, since it requires arduous bookkeeping of variables and efficient calculation of the objective and constraint function required by optimization algorithms [4]. For instance, OpenSim does not currently employ direct collocation.

To improve the accessibility of optimal control methods in musculoskeletal biomechanics, Dembia *et al.* introduced OpenSim Moco with their work [4]: it represents an easy-to-use, customizable, and extensible software tool for solving optimal control problems with OpenSim musculoskeletal models [Fig. 17].

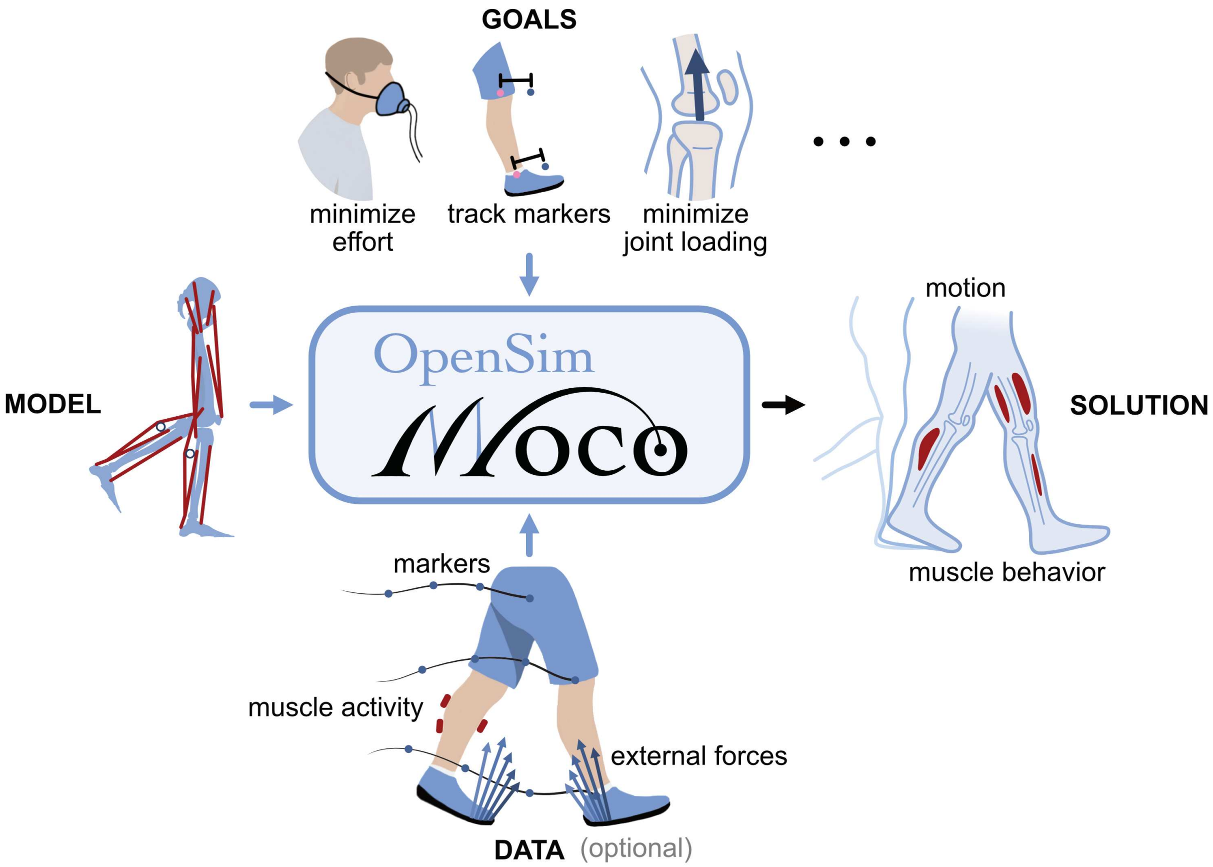


Figure 17: Overview of OpenSim Moco [4].

1.7.2 What is Moco?

Moco is the acronym for *Musculoskeletal Optimal COntrol*, which adequately explains the utility of Moco. In particular, this tool solves optimization problems where control variables are minimized so as to obtain certain musculoskeletal activity values in an OpenSim model [3].

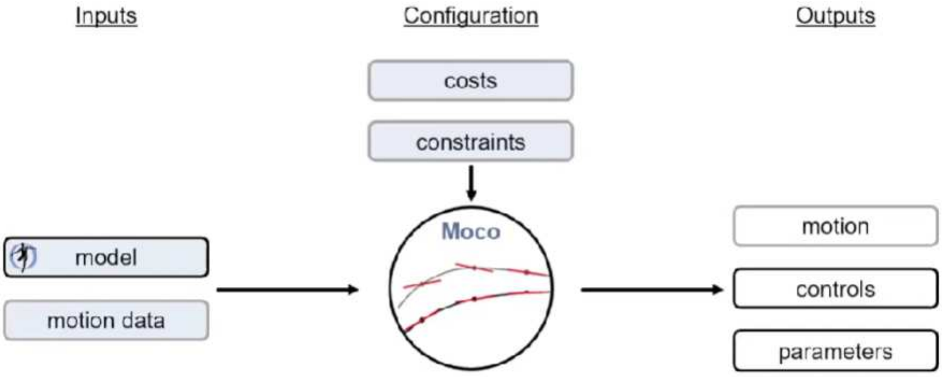


Figure 18: Operating scheme of OpenSim Moco [35].

[Fig. 18] briefly shows how Moco works, what inputs it requires, how this tool has to be configured, and what results can return.

For what is concerned about the inputs, Moco always works with an OpenSim musculoskeletal model that includes all the important information regarding mass properties, muscles, and kinematic constraints. Additionally, motion data can be introduced, depending on the type of problem; particularly, if the motion is prescribed, an *InverseDynamic problem* (MocoInverse) needs to be implemented, and if there is information about the motion but is not certainly defined, a *Tracking problem* (MocoTrack) needs to be implemented [3]. Finally, if there is no information about motion, a generic MocoStudy problem is the one to adoperate.

These possible problems will be further described in the following paragraphs.

1.7.3 The optimal control problem

As previously explained, most of musculoskeletal simulations are posed as optimal control problems, including Moco strategy: in particular, Moco deals with finding those control variables that, under certain dynamical restrictions, are able to minimize a cost function as much as possible [3].

The most generic cost function in Moco is the following one:

$$min \sum_j w_j J_j(t_0, t_f, y_0, y_f, \lambda_0, \lambda_f, x_0, x_f, p, S_{c,j})$$

where J_j corresponds to the j-th cost term, w_j is its associated weight, t_0 and t_f the initial and the final instant of time, y is a vector containing generalized coordinates and speed through time, x is a vector containing time histories of control variable, λ is the vector with kinematic constraints multipliers, p is a parameter which works as an exponent, and $S_{c,j}$ consists on the integral of the j-th cost goal:

$$S_{c,j} = \int_{t_0}^{tf} s_{c,j}(t, y, x, \lambda, p)dt$$

Control variables are those whose function is to regulate the behavior of the system by minimizing the objective function, while *state variables* are those which describe the evolution of the system [3]. Still, some auxiliary state variables may appear in the cost function: these can be either the muscle activation or the muscle fiber length [3].

Once defined the cost function to minimize, constraint must be described. As every dynamical system, the Moco problem is regulated by the equations of motion, which result to be the most important system constraint:

$$u = \dot{q}$$

$$M(q, p) * \dot{u} + G(p.q)^T * \lambda = f_{app}(t, y, x, p) - f_{inertial}(q, u, p)$$

where u is the second derivative of the generalized coordinates, $M(q, p)$ is the mass matrix, G^T is the transpose Jacobian matrix of constraints, λ is the vector containing Lagrange multipliers, f_{app} is the vector applied forces (in particular, gravity and muscle forces), and $f_{inertial}$ is the vector containing inertial forces (centripetal, giroscopic and Coriolis forces) [3].

If auxiliary state variables exist, they follow their own dynamic equations, which can be *explicit* or *implicit*, depending on if their first derivative is respectively cleared or belongs to a term equal to zero:

$$explicit : \dot{z}(t) = f_{\dot{z}_{ex}(y)}(t, y, x, \lambda, p)$$

$$implicit : 0 = f_{\dot{z}_{im}(t)}(t, y, \dot{z}_{im}(t), x, \lambda, p)$$

Subsequently, kinematic constraints must be defined; these determine the relations between generalized coordinates, and are expressed as equations containing a combination of these coordinates equal to zero:

$$\phi(q, p) = 0$$

Boundaries are another set of constraints to be defined to impose upper and lower limits to determined expressions involving every variable in the problem:

$$V_{L,k} \leq V_k(t_0, t_f, y_0, y_f, \lambda_0, \lambda_f, x_0, x_f, p, S_{b,k}) \leq V_{U,k}$$

where $S_{b,k}$ is the integral through time of $s_{b,k}$, a term belonging to the k-th boundary constraint goal:

$$S_{b,k} = \int_{t_0}^{t_f} s_{b,k}(t, y, x, \lambda, p)dt$$

Lastly, path constraints might be described as those which enforce constraints along the trajectory:

$$g_L \leq (t, y, x, \lambda, p) \leq g_U$$

When every constraint has been described, the next step is to define boundaries associated with every variable involved in the problem.

Firstly, limits imposed to initial and final values of states and controls must be set:

$$y_{0,L} \leq y_0 \leq y_{0,U}$$

$$y_{f,L} \leq y_f \leq y_{f,U}$$

$$x_{0,L} \leq x_0 \leq x_{0,U}$$

$$x_{f,L} \leq x_f \leq x_{f,U}$$

then, initial and final instant of time:

$$t_{0,L} \leq t_0 \leq t_{0,U}$$

$$t_{f,L} \leq t_f \leq t_{f,U}$$

following with the states and controls themselves:

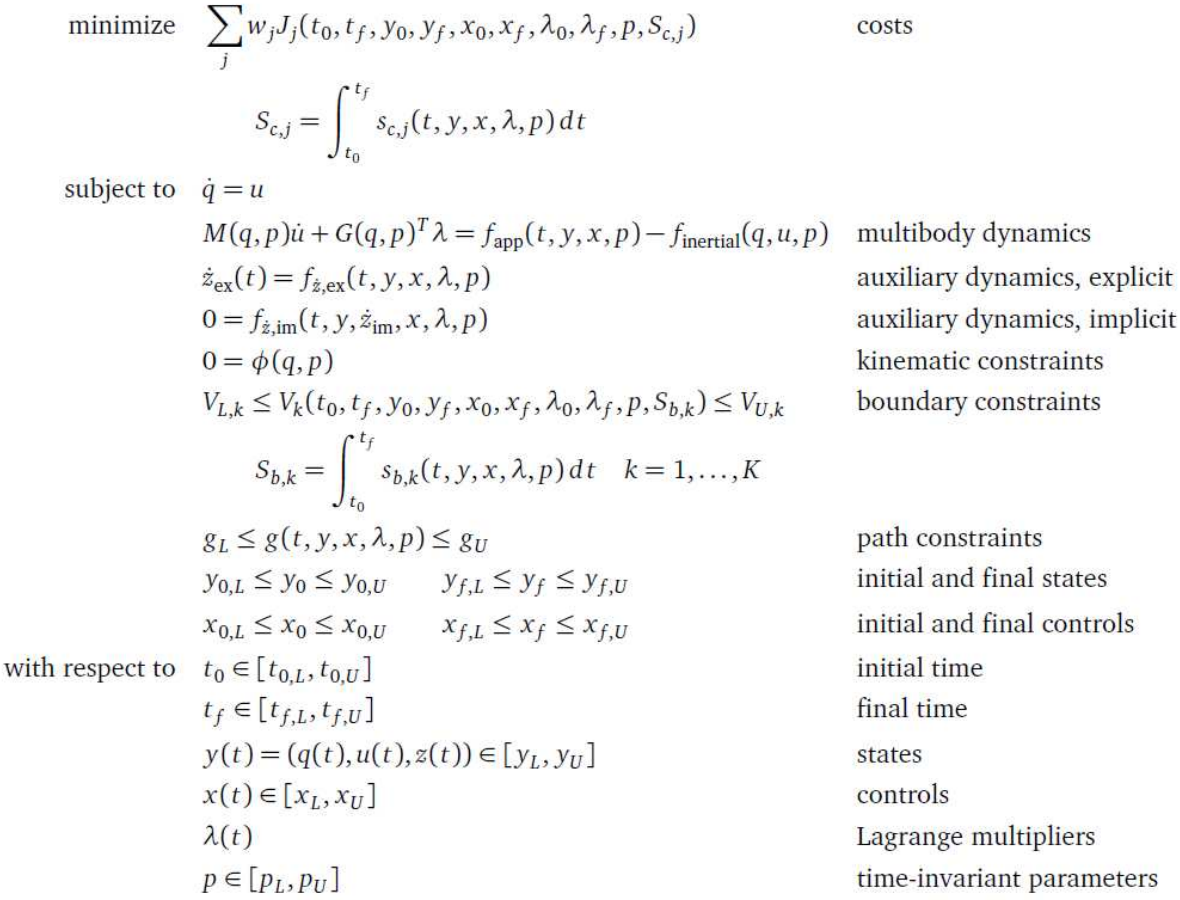
$$y_L \leq y(t) \leq y_U$$

$$x_L \leq x(t) \leq x_U$$

lastly, time-invariant parameter p:

$$p_L \leq p \leq p_U$$

Now that cost function, constraints, boundaries, and variables have been described, all terms can be expressed in the generic formulation of the optimal control problem faced by Moco, as shown in [Fig. 19].



$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_j w_j J_j(t_0, t_f, y_0, y_f, x_0, x_f, \lambda_0, \lambda_f, p, S_{c,j}) & \text{costs} \\
 & \qquad S_{c,j} = \displaystyle\int_{t_0}^{t_f} s_{c,j}(t, y, x, \lambda, p)\, dt & \\
\text{subject to} & \dot{q} = u & \\
 & M(q,p)\dot{u} + G(q,p)^T \lambda = f_{\text{app}}(t, y, x, p) - f_{\text{inertial}}(q, u, p) & \text{multibody dynamics} \\
 & \dot{z}_{\text{ex}}(t) = f_{\dot{z},\text{ex}}(t, y, x, \lambda, p) & \text{auxiliary dynamics, explicit} \\
 & 0 = f_{\dot{z},\text{im}}(t, y, \dot{z}_{\text{im}}, x, \lambda, p) & \text{auxiliary dynamics, implicit} \\
 & 0 = \phi(q, p) & \text{kinematic constraints} \\
 & V_{L,k} \leq V_k(t_0, t_f, y_0, y_f, x_0, x_f, \lambda_0, \lambda_f, p, S_{b,k}) \leq V_{U,k} & \text{boundary constraints} \\
 & \qquad S_{b,k} = \displaystyle\int_{t_0}^{t_f} s_{b,k}(t, y, x, \lambda, p)\, dt \quad k = 1, \ldots, K & \\
 & g_L \leq g(t, y, x, \lambda, p) \leq g_U & \text{path constraints} \\
 & y_{0,L} \leq y_0 \leq y_{0,U} \qquad y_{f,L} \leq y_f \leq y_{f,U} & \text{initial and final states} \\
 & x_{0,L} \leq x_0 \leq x_{0,U} \qquad x_{f,L} \leq x_f \leq x_{f,U} & \text{initial and final controls} \\
\text{with respect to} & t_0 \in [t_{0,L}, t_{0,U}] & \text{initial time} \\
 & t_f \in [t_{f,L}, t_{f,U}] & \text{final time} \\
 & y(t) = (q(t), u(t), z(t)) \in [y_L, y_U] & \text{states} \\
 & x(t) \in [x_L, x_U] & \text{controls} \\
 & \lambda(t) & \text{Lagrange multipliers} \\
 & p \in [p_L, p_U] & \text{time-invariant parameters}
\end{array}
$$

Figure 19: Generic formulation of the optimal control problem faced by Moco [4].

1.7.4 Direct collocation method

As previously introduced, the method used by Moco to solve the optimal control problem is the **direct collocation method**; the optimal control is a trajectory optimization problem continuous in time (as it introduces integrals in the objective function). To solve it, Moco makes use of these algorithms which transform the optimal control problem into a non-linear discrete optimization problem, by discretizing the trajectory optimization problem into a determined number of control points, named *nodes* [35].

One of the main advantages about using this method is that dynamic constraints are taken into account in the cost function, as they include control variables which must minimize the goal; this allows finding a solution for the optimization problem easier [35].

Moco can use two different collocation methods to solve the optimal control problem: the *trapezoidal method*, and the *Hermite-Simpson's method*.

Trapezoidal method The trapezoidal direct collocation method uses the *trapezoidal quadrature rule* to discretize the continuous problem in N-control points and approximate the integrals as summations of discrete variables, evaluated in those control nodes:

$$\int_{t_0}^{t_f} f(\tau, x(\tau))d\tau \approx \sum_{i=0}^{N-1} trap_i = \sum_{i=0}^{N-1} \frac{1}{2} * (t_{i+1} - t_i) * (f_i + f_{i+1})$$

where N is the total number of control points, and $f(\tau, x(\tau))$ is any function belonging to an integrand in the optimization problem [35].

Naturally, the following condition needs to be respected in each control point:

$$f(t_i) = f_i$$

In control points, the value of the continuous function and its approximative spline must be the same [35].

The process to obtain control and state variables can be explained as follows: first of all, the spline used to approximate the control trajectories is a linear polynomial expressed as below:

$$u(t) \approx u_k + \frac{\tau}{h_k} * (u_k + u_{k+1})$$

where τ and h_k are expressed as:

$$\tau = t - t_k$$

$$h_k = t_{k+1} - t_k$$

For what concerned states, the dynamic equations can be represented in function of the first derivative of state variables:

$$\dot{x} = f$$

If trapezoidal quadrature rule is used, the following relation is obtained:

$$x \approx x_k + f_k * t + \frac{\tau^2}{2h_k} * (f_k + f_{k+1})$$

Finally, by applying the trapezoidal method, the non-linear discrete optimization problem depicted in [Fig. 20] appears:

$$
\begin{array}{rlll}
\text{minimize} & \displaystyle\sum_{j} w_j J_j(t_0, t_f, y_0, y_n, x_0, x_n, \lambda_0, \lambda_n, p, S_{c,j}) + w_\lambda \sum_{i=1}^{n} \text{trap}_i(\|\lambda\|_2^2) & & \\
 & \qquad \displaystyle S_{c,j} = \sum_{i=1}^{n} \text{trap}_i(s_{c,j}(t, y, x, \lambda, p)) & & \\
\text{subject to} & q_i = q_{i-1} + \text{trap}_i(f_{\dot{q}}(u)) & & i = 1, \ldots, n \\
 & u_i = u_{i-1} + \text{trap}_i(f_{\dot{u}}(t, y, x, \lambda, p)) & & i = 1, \ldots, n \\
 & z_{\text{ex},i} = z_{\text{ex},i-1} + \text{trap}_i(f_{\dot{z},\text{ex}}(t, y, x, \lambda, p)) & & i = 1, \ldots, n \\
 & z_{\text{im},i} = z_{\text{im},i-1} + \text{trap}_i(\zeta) & & i = 1, \ldots, n \\
 & 0 = f_{\dot{z},\text{im}}(t_i, y_i, \zeta_i, x_i, \lambda_i, p) & & i = 0, \ldots, n \\
 & 0 = \phi(q_i, p) & & i = 0, \ldots, n \\
 & V_{L,k} \leq V_k(t_0, t_f, y_0, y_f, x_0, x_f, \lambda_0, \lambda_f, p, S_{b,k}) \leq V_{U,k} & & \\
 & \qquad \displaystyle S_{b,k} = \sum_{i=1}^{n} \text{trap}_i(s_{b,k}(t, y, x, \lambda, p)) & & k = 1, \ldots, K \\
 & g_L \leq g(t_i, y_i, x_i, \lambda_i, p) \leq g_U & & i = 0, \ldots, n \\
\text{with respect to} & t_0 \in [t_{0,L}, t_{0,U}] & t_n \in [t_{f,L}, t_{f,U}] & \\
 & y_0 \in [y_{0,L}, y_{0,U}] & y_n \in [y_{f,L}, y_{f,U}] & \\
 & x_0 \in [x_{0,L}, x_{0,U}] & x_n \in [x_{f,L}, x_{f,U}] & \\
 & y_i \in [y_L, y_U] & & i = 1, \ldots, n-1 \\
 & x_i \in [x_L, x_U] & & i = 1, \ldots, n-1 \\
 & \zeta_i \in [\zeta_L, \zeta_U] & & i = 0, \ldots, n \\
 & \lambda_i \in [\lambda_L, \lambda_U] & & i = 0, \ldots, n \\
 & p \in [p_L, p_U]. & &
\end{array}
$$

$$
\begin{array}{rll}
\text{subject to} & u_i = u_{i-1} + \text{trap}_i(\upsilon) & i = 1, \ldots, n \\
 & M(q_i, p)\upsilon_i + G(q_i, p)^T \lambda_i = f_{\text{app},i} - f_{\text{inertial},i} & i = 0, \ldots, n \\
\text{with respect to} & \upsilon_i \in [-\upsilon_B, \upsilon_B] & i = 0, \ldots, n.
\end{array}
$$

Figure 20: Non-linear discrete optimization problem obtained by using the trapezoidal method, where ν is generalized acceleration,and ν_B a very large quantity [4].

Hermite-Simpson method This algorithm is the one used as default by Moco; it allows more accurate results than the previous method, since it introduces additional collocation points at the mesh interval midpoints, resulting to a total of $2N + 1$ grid points in which the trajectory is discretized [35]. With this strategy, control trajectories are approximated by quadratic splines, while state trajectories are approximated by cubic splines.

To apply this method, it is necessary to define the Simpson's rule of quadrature firstly:

$$\int_{t_0}^{t_f} f(\tau, x(\tau))d\tau \approx \sum_{i=0}^{N-1} simpson_i = \sum_{i=0}^{N-1} \frac{h_i}{6} * (f_{i-1} + 4 * f_i + f_{i+1})$$

The value of the state variable in the midpoint belonging to the instant $t_{k+\frac{1}{2}}$ is obtained with the Hermite interpolant:

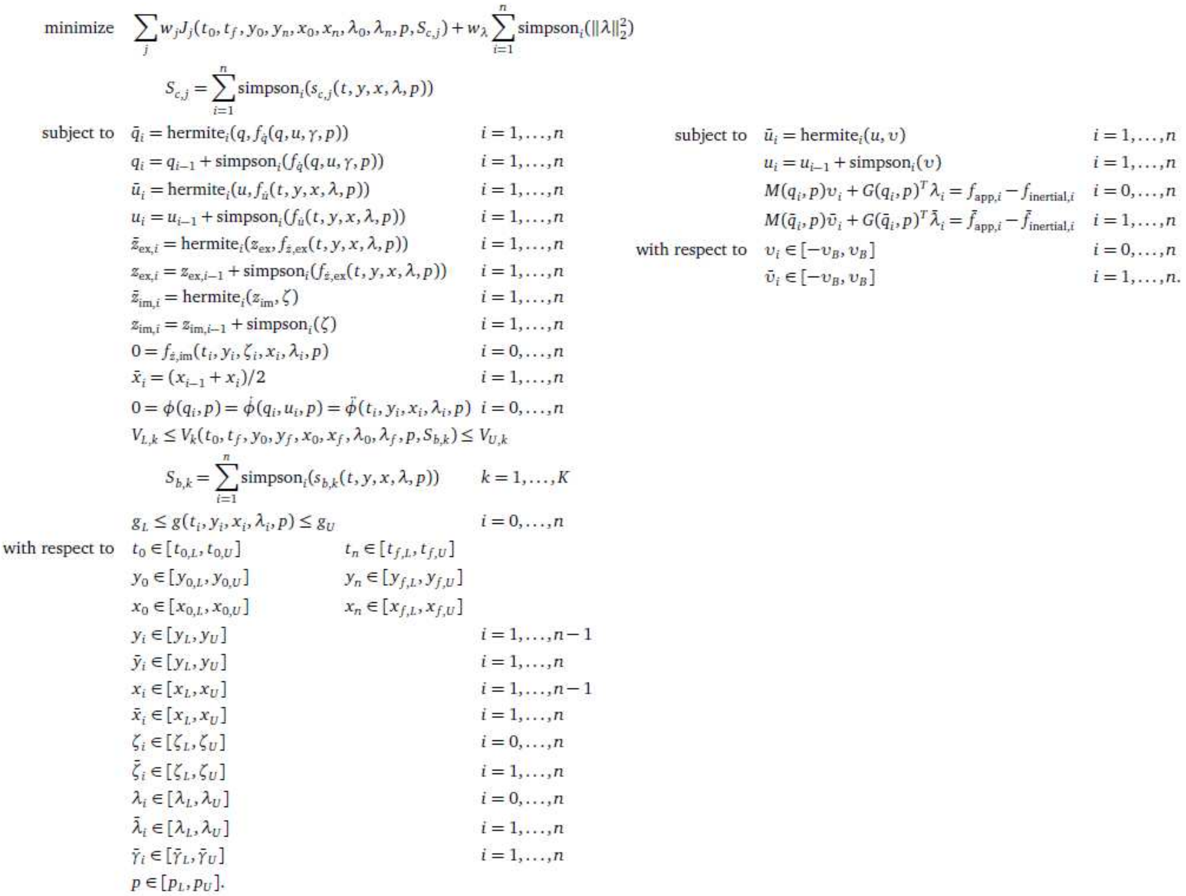
$$hermite_{k+\frac{1}{2}} = x_{k+\frac{1}{2}} = \frac{1}{2} * (x_k + x_{k+1}) + \frac{h_k}{8} * (f_k - f_{k+1})$$

The following two equations to obtain the control and state variables through time are shown below:

$$u(t) = \frac{2}{h_k^2} * (\tau - \frac{h_k}{2}) * (\tau - h_k) * u_k - \frac{4}{h_k^2} * (\tau) * (\tau - h_k) * u_{k+\frac{1}{2}} + \frac{2}{h_k^2} * (\tau) * (\tau - h_k) * u_{k+1}$$

$$x(t) = x_k + f_k(\frac{\tau}{h_k}) + \frac{1}{2} * (-3f_k + 4f_{k+\frac{1}{2}} - f_{k+1}) + (\frac{\tau}{h_k})^2 + \frac{1}{3} * (2f_k - 4f_{k+\frac{1}{2}} + 2f_{k+1}) * (\frac{\tau}{h_k})^3$$

As a result, the formulation of the optimal control problem after applying the Hermite-Simpson collocation method is shown in [Fig. 21].



$$\begin{array}{lll}
\text{minimize} & \sum_j w_j J_j(t_0, t_f, y_0, y_n, x_0, x_n, \lambda_0, \lambda_n, p, S_{c,j}) + w_\lambda \sum_{i=1}^{n} \text{simpson}_i(\|\lambda\|_2^2) & \\
 & \quad S_{c,j} = \sum_{i=1}^{n} \text{simpson}_i(s_{c,j}(t, y, x, \lambda, p)) & \\
\text{subject to} & \bar{q}_i = \text{hermite}_i(q, f_{\dot{q}}(q, u, \gamma, p)) & i = 1, \ldots, n \\
 & q_i = q_{i-1} + \text{simpson}_i(f_{\dot{q}}(q, u, \gamma, p)) & i = 1, \ldots, n \\
 & \bar{u}_i = \text{hermite}_i(u, f_{\dot{u}}(t, y, x, \lambda, p)) & i = 1, \ldots, n \\
 & u_i = u_{i-1} + \text{simpson}_i(f_{\dot{u}}(t, y, x, \lambda, p)) & i = 1, \ldots, n \\
 & \bar{z}_{\text{ex},i} = \text{hermite}_i(z_{\text{ex}}, f_{\dot{z},\text{ex}}(t, y, x, \lambda, p)) & i = 1, \ldots, n \\
 & z_{\text{ex},i} = z_{\text{ex},i-1} + \text{simpson}_i(f_{\dot{z},\text{ex}}(t, y, x, \lambda, p)) & i = 1, \ldots, n \\
 & \bar{z}_{\text{im},i} = \text{hermite}_i(z_{\text{im}}, \zeta) & i = 1, \ldots, n \\
 & z_{\text{im},i} = z_{\text{im},i-1} + \text{simpson}_i(\zeta) & i = 1, \ldots, n \\
 & 0 = f_{\dot{z},\text{im}}(t_i, y_i, \zeta_i, x_i, \lambda_i, p) & i = 0, \ldots, n \\
 & \bar{x}_i = (x_{i-1} + x_i)/2 & i = 1, \ldots, n \\
 & 0 = \phi(q_i, p) = \dot{\phi}(q_i, u_i, p) = \ddot{\phi}(t_i, y_i, x_i, \lambda_i, p) & i = 0, \ldots, n \\
 & V_{L,k} \leq V_k(t_0, t_f, y_0, y_f, x_0, x_f, \lambda_0, \lambda_f, p, S_{b,k}) \leq V_{U,k} & \\
 & \quad S_{b,k} = \sum_{i=1}^{n} \text{simpson}_i(s_{b,k}(t, y, x, \lambda, p)) & k = 1, \ldots, K \\
 & g_L \leq g(t_i, y_i, x_i, \lambda_i, p) \leq g_U & i = 0, \ldots, n \\
\text{with respect to} & t_0 \in [t_{0,L}, t_{0,U}] \qquad t_n \in [t_{f,L}, t_{f,U}] & \\
 & y_0 \in [y_{0,L}, y_{0,U}] \qquad y_n \in [y_{f,L}, y_{f,U}] & \\
 & x_0 \in [x_{0,L}, x_{0,U}] \qquad x_n \in [x_{f,L}, x_{f,U}] & \\
 & y_i \in [y_L, y_U] & i = 1, \ldots, n-1 \\
 & \bar{y}_i \in [y_L, y_U] & i = 1, \ldots, n \\
 & x_i \in [x_L, x_U] & i = 1, \ldots, n-1 \\
 & \bar{x}_i \in [x_L, x_U] & i = 1, \ldots, n \\
 & \zeta_i \in [\zeta_L, \zeta_U] & i = 0, \ldots, n \\
 & \bar{\zeta}_i \in [\zeta_L, \zeta_U] & i = 1, \ldots, n \\
 & \lambda_i \in [\lambda_L, \lambda_U] & i = 0, \ldots, n \\
 & \bar{\lambda}_i \in [\lambda_L, \lambda_U] & i = 1, \ldots, n \\
 & \bar{\gamma}_i \in [\bar{\gamma}_L, \bar{\gamma}_U] & i = 1, \ldots, n \\
 & p \in [p_L, p_U]. &
\end{array}$$

$$\begin{array}{lll}
\text{subject to} & \bar{u}_i = \text{hermite}_i(u, \upsilon) & i = 1, \ldots, n \\
 & u_i = u_{i-1} + \text{simpson}_i(\upsilon) & i = 1, \ldots, n \\
 & M(q_i, p)\upsilon_i + G(q_i, p)^T \lambda_i = f_{\text{app},i} - f_{\text{inertial},i} & i = 0, \ldots, n \\
 & M(\bar{q}_i, p)\bar{\upsilon}_i + G(\bar{q}_i, p)^T \bar{\lambda}_i = \bar{f}_{\text{app},i} - \bar{f}_{\text{inertial},i} & i = 1, \ldots, n \\
\text{with respect to} & \upsilon_i \in [-\upsilon_B, \upsilon_B] & i = 0, \ldots, n \\
 & \bar{\upsilon}_i \in [-\upsilon_B, \upsilon_B] & i = 1, \ldots, n.
\end{array}$$

Figure 21: Optimal control problem formulation by applying the Hermite-Simpson method [4].

1.7.5 Design and implementation

Researchers can use Moco to solve optimal control problems by defining a library of cost and constraint modules, which are implemented through configurable software classes [4]. In particular, the problem under investigation is described by the *MocoProblem* class, while to solve the problem *MocoSolver* is used.
Moco classes are avaiable via C++, MATLAB, Python, and XML text files, allowing every OpenSim user to find the most familiar interface. Developers package *MocoProblem* and *MocoSolver* together into a *MocoStudy*; a briefly overview of *MocoStudy* is depicted in [Fig. 22].

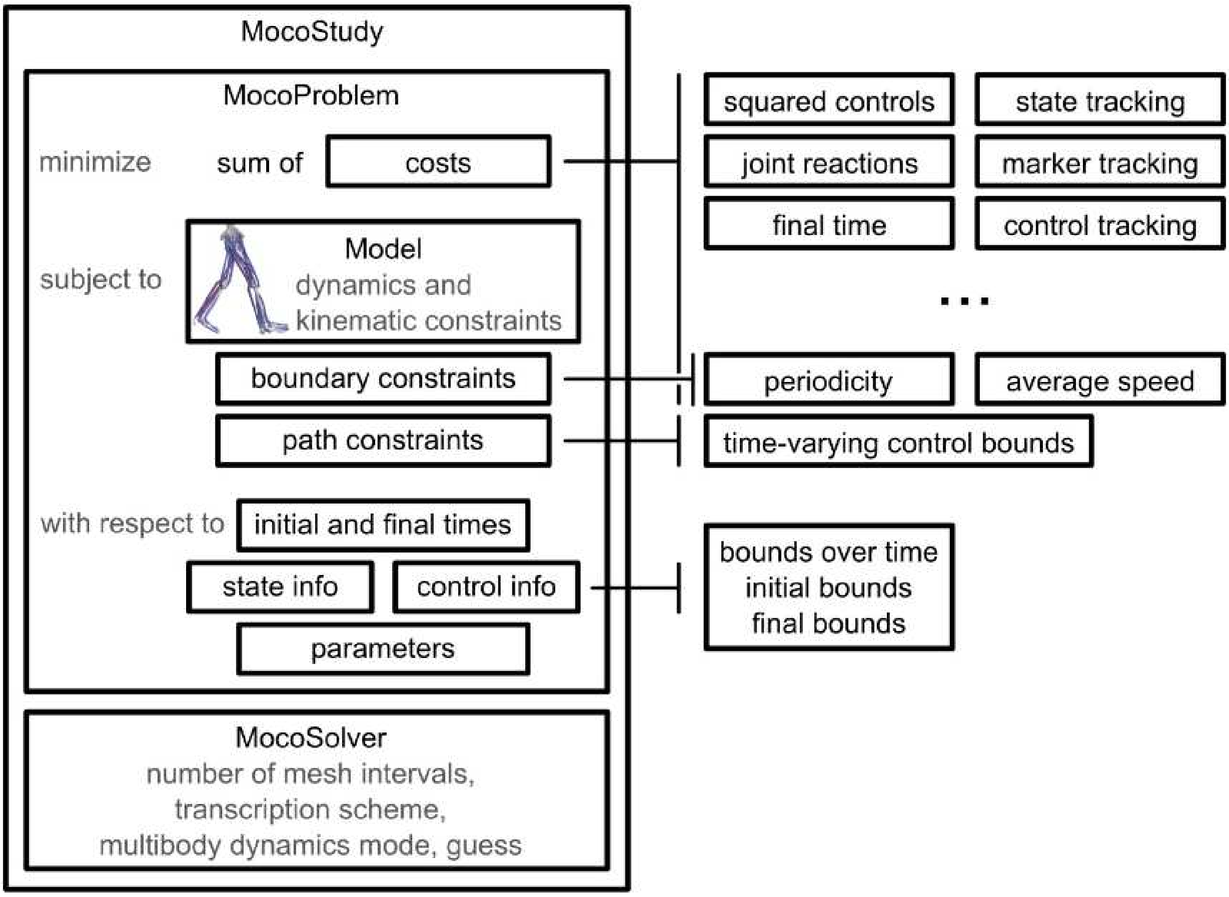


Figure 22: Overview of *MocoStudy* [4].

Solver Moco provides two solvers as subclasses of *MocoSolver*:

1. *MocoCasADiSolver* uses the third-party CasADi library, an open-source package for algorithmic differentiation, a bridge to nonlinear program solvers IPOPT, SNOPT, and others [36, 37, 38].

2. *MocoTropterSolver* uses a direct collocation solver developed by [4] authors called Tropter.

By default *MocoSolver* uses the CasADi library to transcrive the continuous optimal control problem defined by *MocoProblem* into a finite-dimensional nonlinear program; CasADi can also provide the derivatives of the cost and constraint functions using either finite differences or algorithmic differentiation, but Moco uses only the first method to avoid the complexity of adapting the OpenSim codebase to support algorithmic differentiation [4].

1.7.6 Tool for standard problems

At present, Moco provides two tools for solving standard problems, depending on the presence of information about the motion under investigation [4]. [Fig. 23] provides a summary of these, in particular in terms of inputs required and results obtained.

Both *MocoInverse* and *MocoTrack* require a model and kinematic data as inputs, and produce controls and actuators states as outputs; however, these tools solve different optimal control problems. In addition, the tools build internally a *MocoStudy* with solver settings that allow fast and reliable convergence on problems under investigation [4].

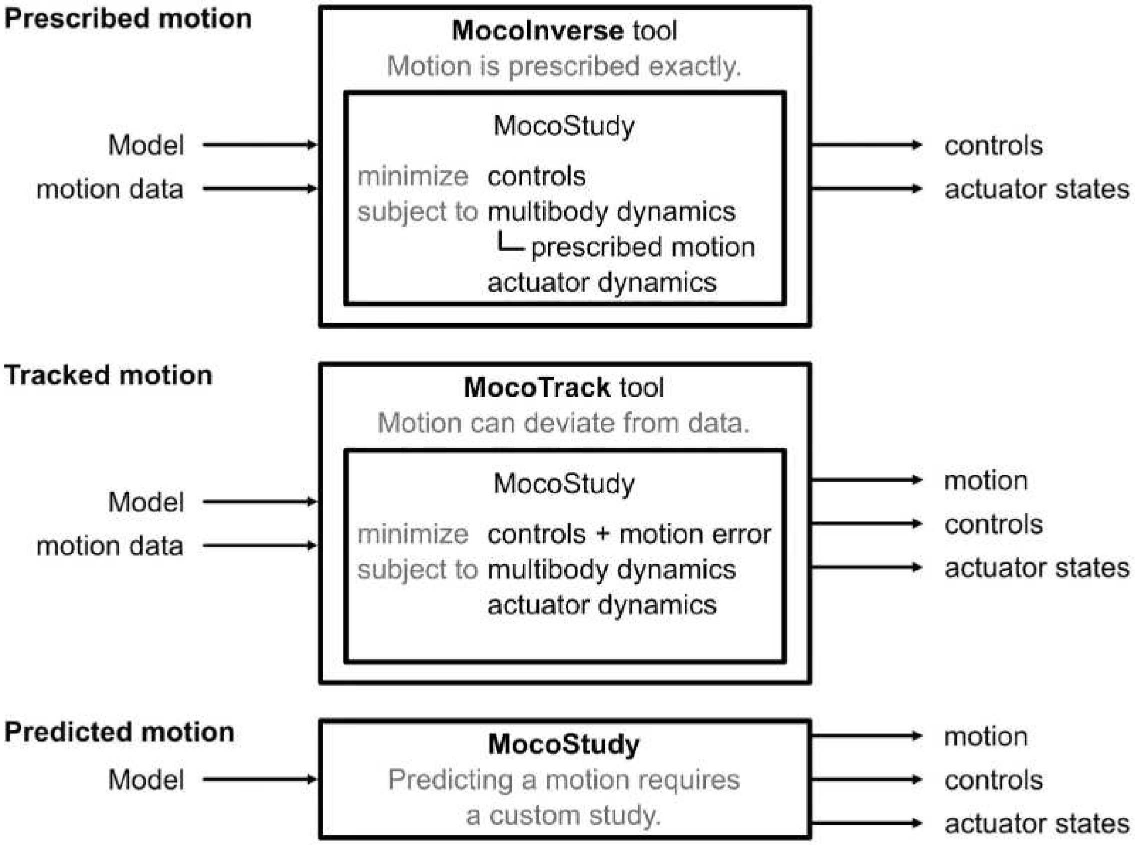


Figure 23: Solving prescribed, tracked, and predicted motion with OpenSim Moco [4].

MocoInverse In *MocoInverse*, the model's kinematics is completely prescribed, so the objective is to obtain the forces which produce this motion with the control variable present in the cost function and the actuator states [3].
This tool provides a faster option when the motion should be enforced exactly, or when measured external forces need to be applied directly to the model [4].

MocoTrack This is the type of problem faced in this project.
Here, information about the motion performed by the model exists, and normally consists in experimental data collected by sensors. Nevertheless, kinematics becomes unknown, so it is one of the variables to obtain with controls and states [3]. The cost deals with the equations of motion while minimizing as much as possible the deviation of the resulting motion from the experimental data [35]. Since the motion is now part of the objective function, it takes longer to compute in comparison with *MocoInverse*; however, this tool permits using contact models, which is the main aim of the following project.

Chapter 2

Methods

The current chapter will go over the methods, techniques, and operations utilized for the experimental portions of this thesis. In specific, the data needed to begin the work, the programs and algorithms used, the protocols used, and the necessary alterations made will be covered.

2.1 Data acquisition

The goal of this thesis is to develop a method for researchers to measure human movement and musculoskeletal function outside of the traditional motion capture laboratory. Consequently, using the data collected by wearable sensors as input for musculoskeletal modeling represents a new strategy for the estimation of various movement parameters (i.e., muscle activations and joint contact forces) which does not require a controlled environment. Moreover, to eliminate the use of force platforms (FP) to obtain information regarding the forces acting during walking, in this thesis coupled with data from IMUs a model is used that predicts contact geometries on the feet capable of recording force values during analysis. Finally, these two elements will be implemented in a MocoTrack analysis.

The first step was to obtain the results using the gold standard for capturing human motion as input for the Motion capture (MoCap) system combined with in-ground embedded force plate systems. Secondly, the same approach was applied to obtain results using the Inertial capture (InCap); these results were compared with some reference data depending on the method under investigation (i.e., results obtained from InCap data were compared with InCap input data).

In order to construct a working dataset, 3D MoCap data were collected at the *Movement & Posture Analysis Laboratory Leuven (MALL)*, at the Department of Movement Science, KU Leuven, Leuven, Belgium. Ten healthy adults with no musculoskeletal or neurological disorders volunteered for the study (4 males and 6 females; age: 59.7 ±7 yr; height: 1.55 ± 0.52 m; weight: 66.5 ± 24.0 kg) and performed multiple walking trials. [Tab. 1] displays an overview of the subjects' attributes and gait information. It should be noted that the total number of gait trials is not the same across participants, because it reflects the variation in the number of valid trials per participant.

	Body weight (kg)	**Height (m)**	**Gender**	**BMI (kg/m^2)**	**n° of walking trials**	**Average gait speed (m/s)**
H01	87.6	1.81	M	26.25	3	1.65
H02	77.5	1.69	F	26.96	5	1.22
H03	76.5	1.70	F	26.30	4	1.33
H04	55.6	1.62	F	20.96	6	1.34
H06	68.0	1.58	F	26.84	4	1.08
H07	78.5	1.85	M	23.08	5	1.46
H08	71.2	1.63	F	26.72	4	1.44
H011	66.5	1.76	M	21.63	3	1.19
H012	87.5	1.78	M	27.77	6	1.29
H014	63.4	1.60	F	25.39	4	1.12

Table 1: Summary of 10 healthy subject characteristics and gait data.

A full-body Plug-in-Gait marker set was used to place the retro-reflective markers which were supplemented with 3-marker clusters on the upper and lower arms and legs, as well as anatomical markers on the sacrum, medial femur epicondyles, and medial malleoli, for a total of 65 markers.

Ground-reaction forces and marker trajectories were then recorded using 3 ground-embedded force plates (AMTI, Watertown, USA,1000Hz) and a 10-camera motion capture system (100 Hz, Vicon, Oxford, UK).

On the other hand, 3D Inertial capture (InCap) was used to synchronously measure using an Xsens IMC system (Xsens Awinda, Xsens Technologies BV, Enschede, Netherlands) sampled at 60 Hz and processed by the matching software Xsens MVN Analyze 2018 (Xsens Awinda, Xsens Technologies BV, Enschede, Netherlands). 17 IMU modules were mounted on the head, sternum, pelvis, upper legs, lower legs, feet, shoulders, upper arms, forearms and hands.

The individuals were told to move barefoot at their own, personal walking rhythm. Each participant stood upright while their segment dimensions were recorded and later entered into the XsensMVN program before starting trials: these measures included the length of the foot, the lengths between the ankle, knee, hip, and top of the head from the ground, the width of the pelvis, and the shoulders. The individual was in a neutral stance throughout the calibration of the IMC system, which was then followed by a walk calibration. Then, using a *sensor-to-segment* calibration technique, XsensMVN 2018 determines the kinematics of 23 body segments by relating the 17 sensor orientations [39].

In order to scale the generic musculoskeletal model, participants were next required to complete a **static calibration trial** that was captured by the MoCap system. The opto-reflective markers employed in this trial identified key anatomical landmarks.

The subjects conducted 3D motion analysis following the calibration of both motion capture devices. While performing a number of walking trials, the two motion capture systems simultaneously and synchronously captured ground reaction forces and moments as well as 3D body movements. The synchronization was based on the manufacturer's guidelines with a specific trigger at the start/stop recording time.

2.2 Musculoskeletal model

The musculoskeletal model used results from the model employed in the article *"Modeling toes contributes to realistic stance knee mechanics in three-dimensional predictive simulations of walking"* [40] modified *ad hoc* for the purposes of this work.

Falisse *et al.* [40] started from a model that existed in the OpenSim library, known as the *"Full body running model"* developed by Hamner *et al.* [41]. [Fig. 24] displays the musculoskeletal model used in [40].

In particular, the model presented thirty-one (31) degrees of freedom: six (6) pelvis-to.ground DOFs, three (3) hip DOFs, one (1) knee DOF, one (1) ankle DOF, one (1) subtalar DOF, one (1) metatarsophalangeal-toe DOF, three (3) lumbar DOFs, three (3) shoulder DOFs, and one (1) elbow DOF; moreover, ninety-two (92) muscles actuating the lower limb and lumbar joints and eight (8) ideal torque motors actuating the shoulder and elbow joints were added to the model [40].

Since the purpose of the article involved using such a model in order to simulate a movement (i.e., walking) and then estimate the necessary parameters, the authors added **six (6) contact spheres** for each foot, three of which referred to the body calcaneus (named *calcn* in OpenSim) and the remaining ones to the body toes (named *toes* in OpenSim) [40]. Moreover, the moving knee flexion axis of the generic model was fixed to its anatomical reference position in order to increase computational speed [40]. Then, passive stiffness (exponential) and damping (linear) were added to the lower limb and lumbar joint, to model ligaments and other passive structures.

Furthermore, the authors used the *Raasch's model* to describe muscle excitation-activation coupling, and a *Hill-type muscle model* to describe muscle-tendon interaction and dependence of muscle force on fiber length and velocity [42, 43]. The skeletal motion was modeled with the *Newtonian rigid body* assumption and smooth approximations of compliant were described by *Hunt-Crossley foot-ground contacts* [40].

More details about the musculoskeletal model can be found in the cited work [40] and related ones.

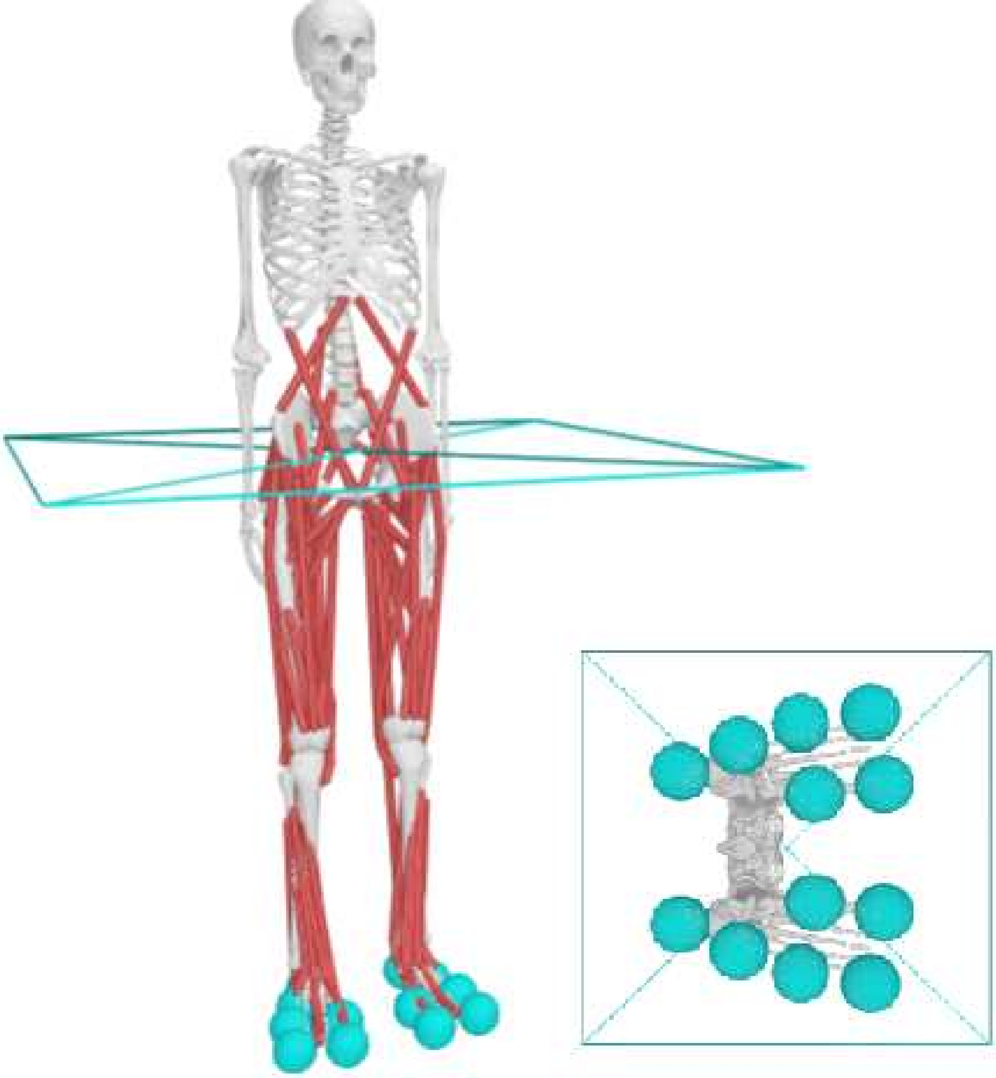


Figure 24: Musculoskeletal model used in [40] with focus on contact spheres placement.

Starting from the model just illustrated, the musculoskeletal model used in the present project has been developed: first, in terms of degrees of freedom (DOFs), two more DOFs were added to the knee joint, thus allowing the joint to perform the movements of adduction-abduction and intra-extra rotation in addition to flexion-extension.

Secondly, **reserve actuators** were added to each joint: in particular, reserve actuators are torques added on each joint to augment the actuator's force, in order to enable the simulation to run [22]. Thanks to these, it was possible to carry out the analyses and obtain consistent results from a dynamic point of view.

A further modification was made regarding the muscle groups present on the previous model: since this work aims to study only the movement of a single leg (particularly, the right leg), the muscle groups of the leg not under investigation were eliminated; thanks to this modification, computational time was drastically reduced, allowing us to arrive at convergence significantly faster.

Moreover, according to the protocol chosen, markers were added to the model; [Fig. 25] depicts the correct placement of markers [44]. It should be noted that the anatomical parts under investigation are the lower limbs, so for this purpose the marker set added to the model includes thirty-five (35) markers placed on the legs and trunk, leaving out the upper limbs.

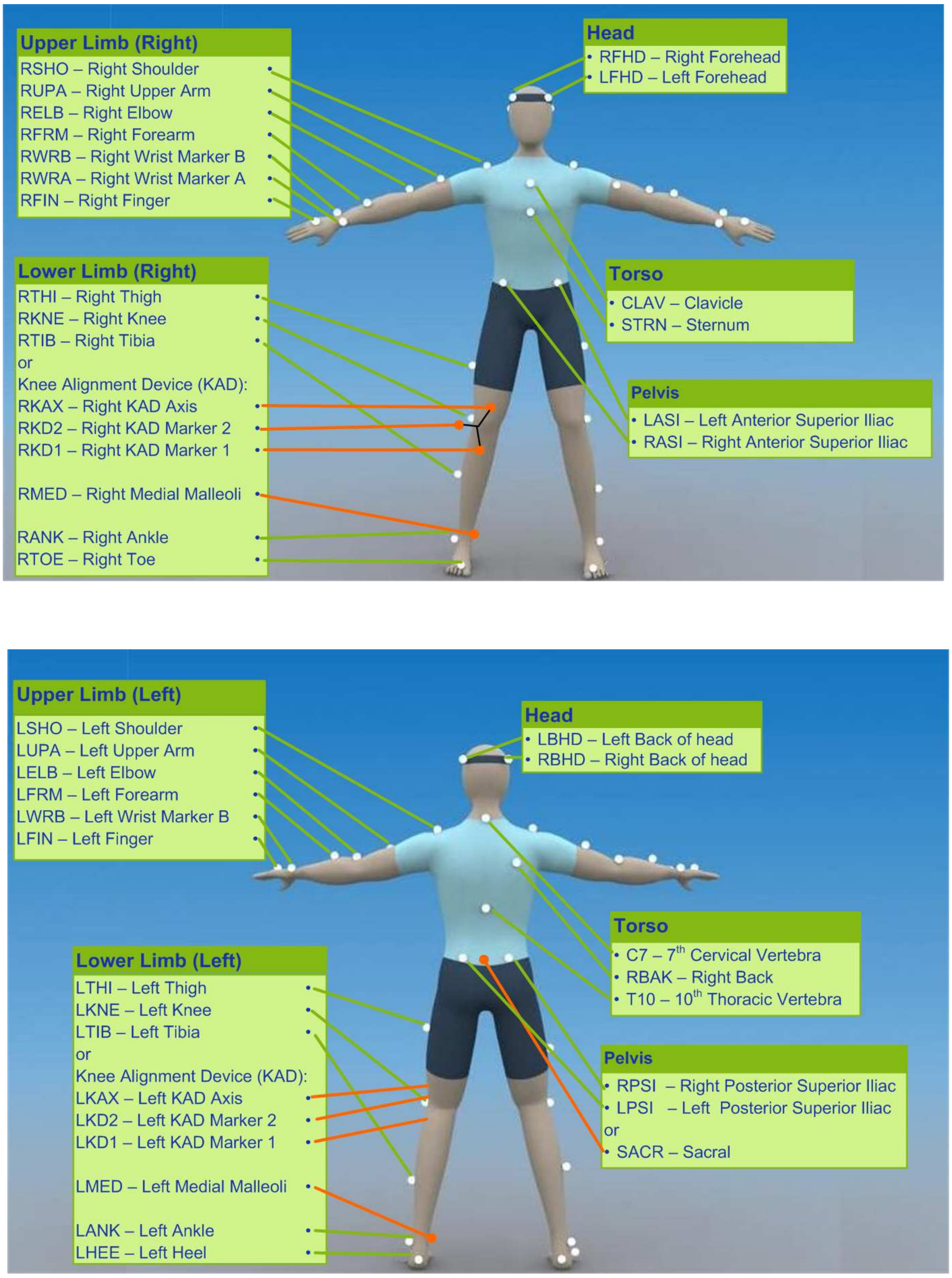


Figure 25: Full Plug-in-Gait marker set, front and back view [44].

Finally, the main change was made with regard to contact geometries: a new configuration of spheres was designed so that the forces recorded by the spheres themselves could be better estimated. [Fig. 26] shows in detail the spheres placement on the foot.

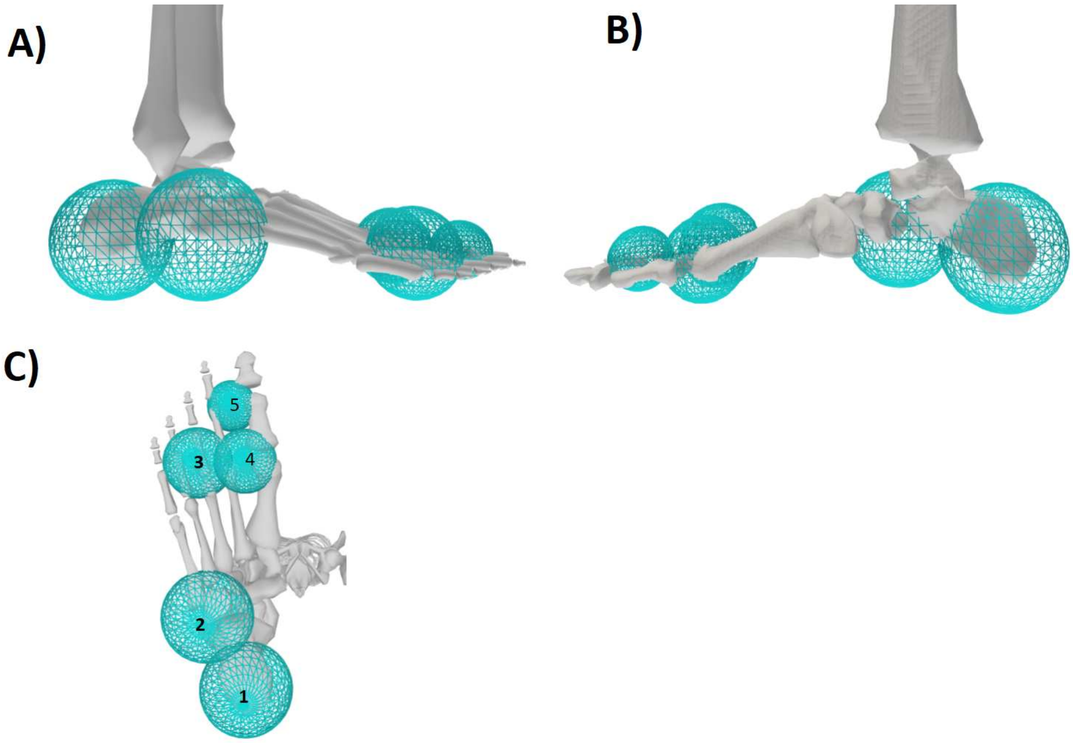


Figure 26: Contact spheres placement on foot. In particular: A) right view, B) left view, C) plantar view.

These five (5) spheres were designed in order to estimate and record values of force exerted by the ground during a task. The <ContactGeometry> class was added to the musculoskeletal model where within it the geometric characteristics of the spheres were defined, such as position, reference body, and radius; in particular, all the spheres were defined with respect to the *calcn_r* body, presenting different position coordinates based on their specific positioning but with reference to the calcaneus frame. Moreover, three (3) different radii were assigned to three groups of spheres such that the size of each sphere was adapted to the anatomical measurements of the foot. [Tab. 2] below shows the radii with reference to the numbered spheres in [Fig. 26].

Sphere n°	**Radius (m)**
1-2	0.032
3-4	0.021
5	0.016

Table 2: Values of radius used for each contact sphere.

In addition to geometric characteristics, mechanical characteristics designed so that they can best estimate forces were also assigned to each sphere. Specifically, the parameters selected for the following study are as follows:

- **Stiffness constant**: the rigidity of the sphere;
- **Dissipation coefficient**: the amount of energy that is dissipated or lost in a system due to friction or other forms of resistance;
- **Static friction coefficient**: the amount of force required to initiate sliding motion between two surfaces in contact with each other;
- **Dynamic friction coefficient**: the amount of force required to maintain relative motion between two surfaces in contact with each other;
- **Viscous friction coefficient**: the amount of frictional force that arises between two objects in relative motion through a viscous fluid (i.e.,air or water);
- **Transition velocity**

More detail in terms of chosen values can be found in [Tab. 3] below.

Properties	**Value**
Stiffness constant	10000000 (N/m^2)
Dissipation coefficient	2 (s/m)
Static friction coefficient	0.8
Dynamic friction coefficient	0.8
Viscous friction coefficient	0.8
Transition velocity	0.2 (m/s)

Table 3: Values of properties used for each contact sphere.

For the same reason explained above, the spheres were placed only on one foot (specifically, on the right foot).

To summarize, the new model developed for this project and named *generic_ model.osim* presents:

1. **Thirty-five (35) DOFs** (pelvis-to-ground: 6 DOFs, hip: 3 DOFs, knee: 3 DOF, ankle: 1 DOF, subtalar: 1 DOF, metatarsophalangeal-toe: 1 DOF, lumbar: 3 DOFs, shoulder: 3 DOFs, and elbow: 1 DOF);
2. **Forty-nine (49) muscles** actuating the right leg and the lumbar joint;
3. **Reserve actuators** applied to every joint to enable the simulation to run;
4. **Thirty-five (35) markers** placed on the lower limbs and the trunk;
5. **Five (5) contact spheres** placed on the right foot.

[Fig. 27] depicts the musculoskeletal model introduced above.

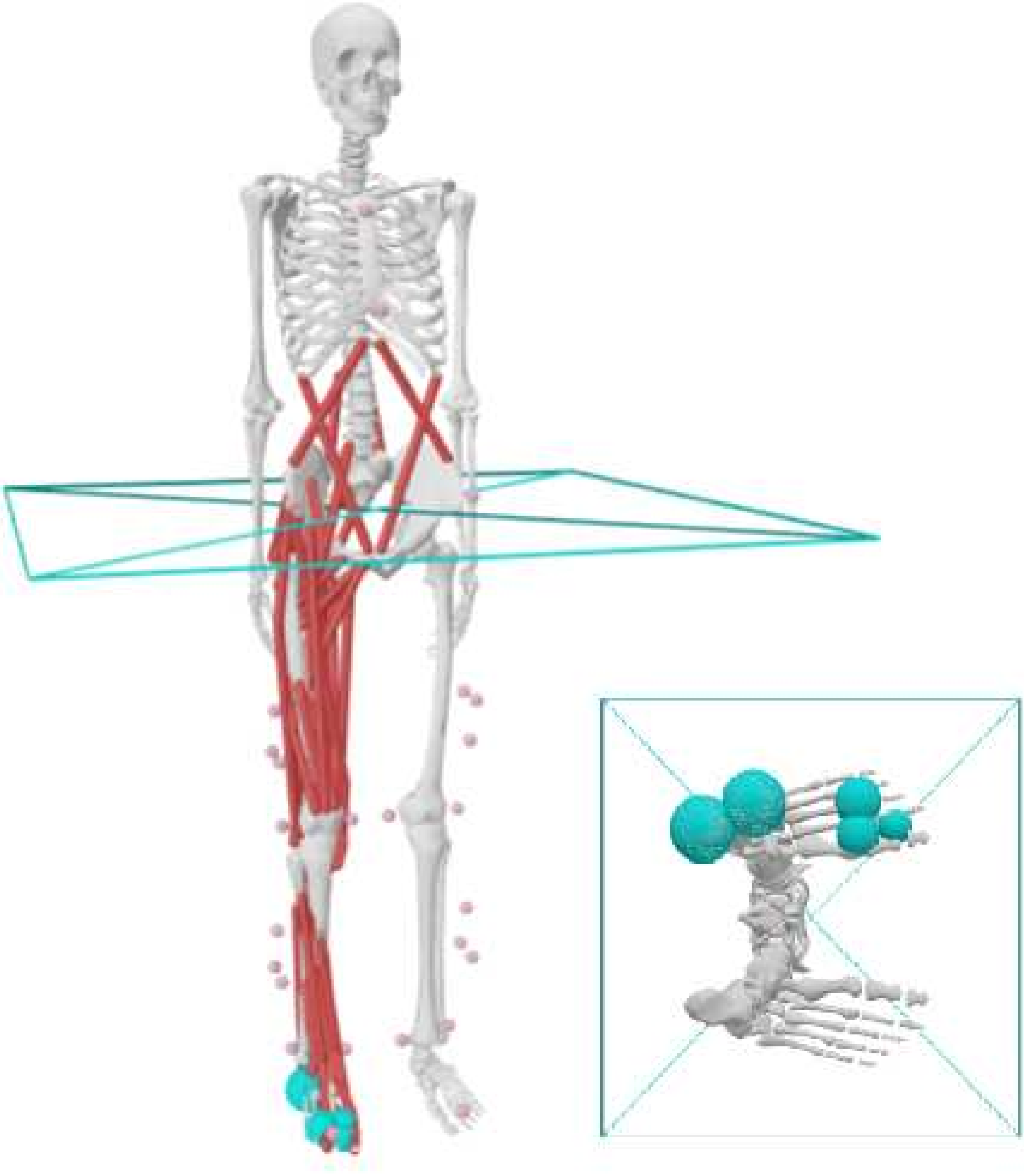


Figure 27: Musculoskeletal model used in this thesis with focus on contact spheres placement.

2.3 Operative workflow

The operative workflow that was developed and followed in this thesis is shown in [Fig. 28]: it represents a two-way scheme to highlight the different procedures used for the two strategies, but ultimately to arrive at the same results of interest (*Moco workflow*), preceded by a common section (*common workflow*). All the steps and procedures for the two approaches are detailed in two separate sections.

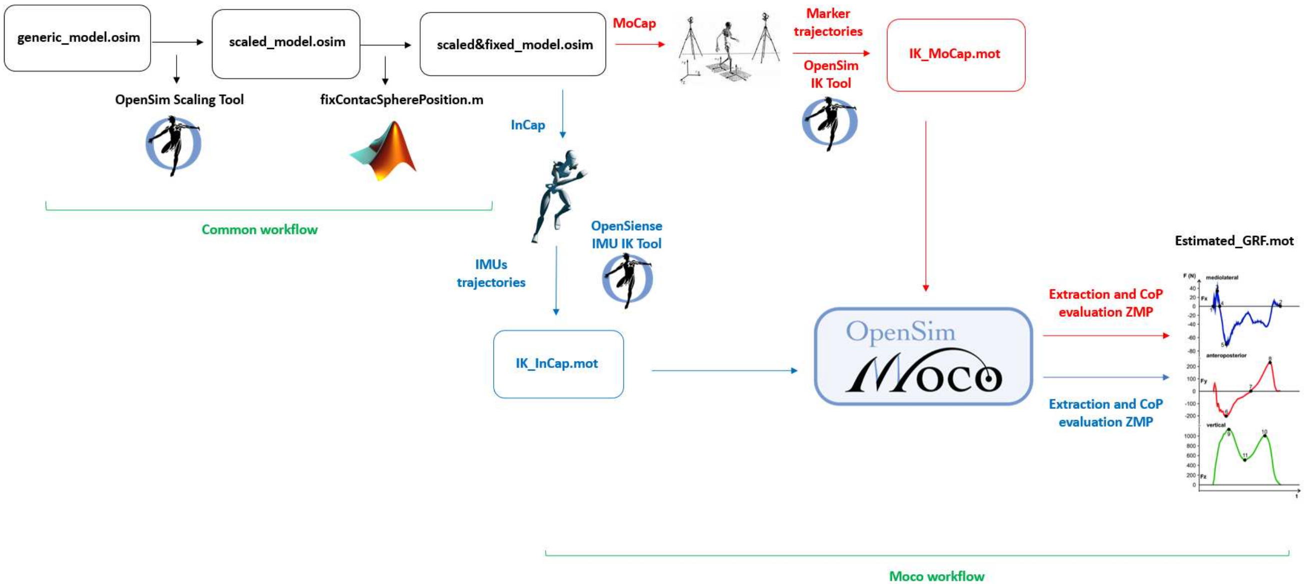


Figure 28: General workflow developed for this research project. Focus on the two main procedures will be provided in the following paragraphs.

2.3.1 Common workflow

The first portion of the workflow shown in [Fig. 28] is named *the common workflow*: it consists of deriving from the generic model shown in the previous paragraph (*generic_ model.osim*) the model based on the anthropometric measurements of the subject under study.
For this, the generic model is scaled by means of the OpenSim Scaling tool, using a .trc file from MoCap as a reference; in particular, this file contains the positions of the markers applied to the subject collected during a static position, called *T-pose*. An example of T-pose contained in a .trc file is shown in [Fig. 29].
By analyzing in more detail the scaling operation performed on the generic model, the following points emerge:

1. **Customized scale factors** were used, obtained by comparing distances between markers on the model and experimental marker positions provided in a .trc file;

2. **Customized static pose weight** were used so as to assign less contribution to markers that are part of a cluster during the scaling operation.

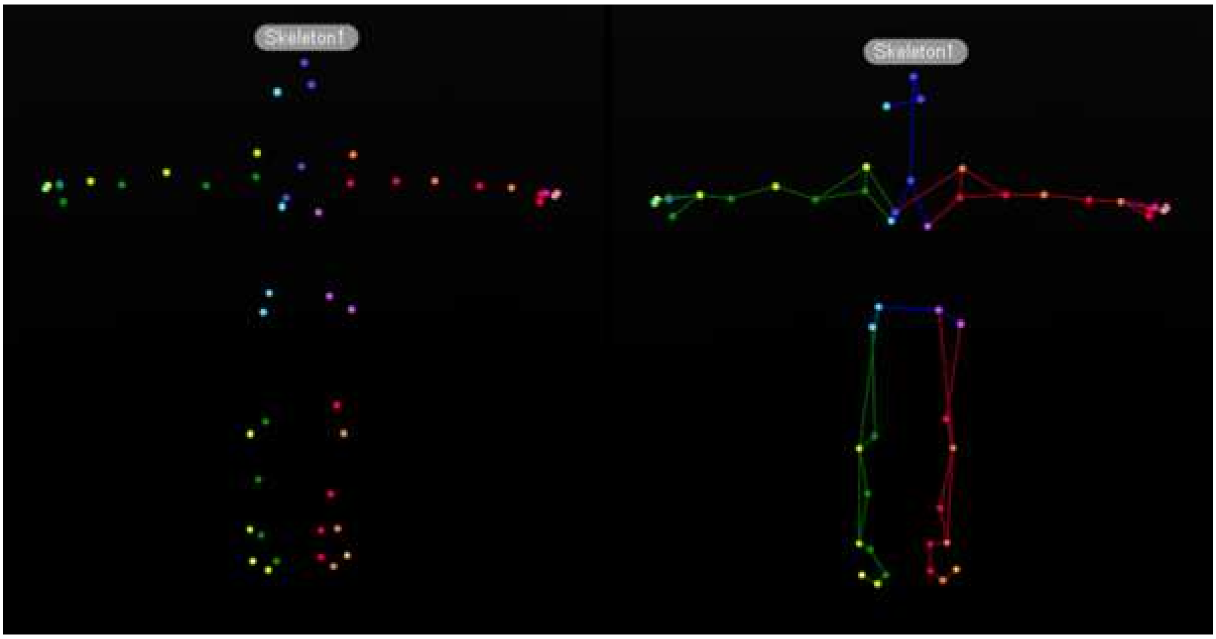


Figure 29: Example of T-pose [45].

However, the Scaling Tool does not take into account any contact geometries present on the model; if any, their position relative to the body to which they are assigned remains unchanged. This could be a source of errors, since variation in the anthropometric measurements of the model results in a certain change in the measurements of the body to which the geometries are assigned (i.e., the foot). In this case, it was necessary to change the position of the spheres placed on the right foot so that they are also positioned correctly in the new scaled model.
For this purpose, the Matlab function **fixContactSpherePosition.m** was implemented: providing it with the generic model and the newly scaled model as input, the function calculates new scaling parameters related to the spheres on the foot thus adjusting their positions, providing as output a new model with the "fixed" positions of the spheres.

The common workflow steps previously illustrated can be depicted as in [Fig. 30]; the result is the model scaled to the subject's measurements prepared for the next steps, called *scaled&fixed_model.osim*.

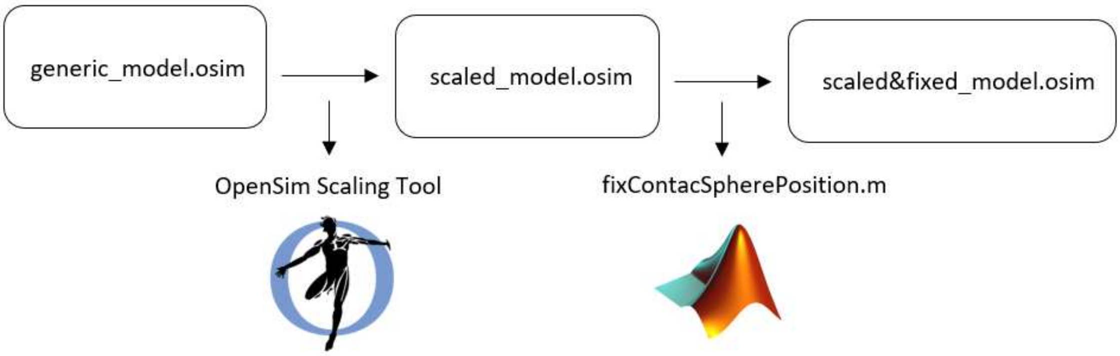


Figure 30: The common workflow.

2.3.2 Moco workflow

The following section will explain in detail the steps shown in the second part of [Fig. 28]. The Matlab script used and described below can be found in *Appendix A*.
The purpose of this thesis is to use data from InCap strategy to estimate the parameters of interest, so as to eliminate the use of the MoCap apparatus; however, in order to validate the method, results from MoCap data were also obtained and analyzed.

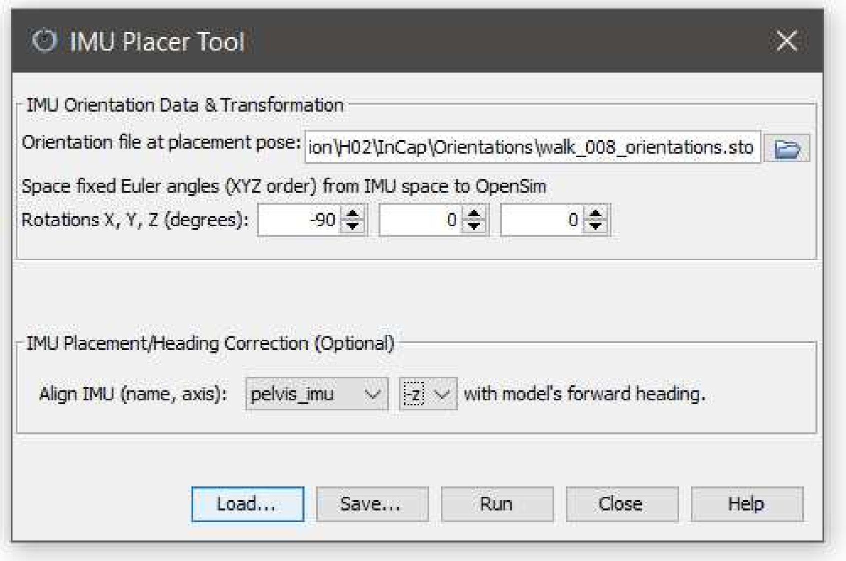
InCap Moco workflow The data of each walking trials acquired by the InCap measurement system was available in the .mnvx format. However, the conversion to OpenSim's file format was needed: it was performed by means of custom-built Matlab script developed to load the Xsens data from .mvnx files and save them in the .mat format, where a structure array was created to store values of acceleration, orientation, and magnetic field data for each sensor, and all the information about the static calibration, segments' and joints' names.
Another Matlab function was then used to create an OpenSense file for kinematic analysis in OpenSim. In particular, it transforms the rotation matrices into quaternions to create a single, time-synchronized storage .sto file type for orientations. The output file *walk_00x_orientations.sto* contains orientations for each sensor, where each column corresponds to an IMU sensor represented as a Frame (X, Y, Z coordinates), named as *<bodyname>_imu*. Each sensor was associated with the respective body segment; in this study, only the lower limb and trunk were taken into consideration, while upper body segments were avoided to lower computation costs.
All the sensors used are listed in [Tab. 4].

Body name	**Sensor attached**
T8	torso_imu
Pelvis	pelvis_imu
RightUpperLeg	femur_r_imu
RightLowerLeg	tibia_r_imu
Rightfoot	calcn_r_imu
LeftUpperLeg	femur_l_imu
LeftLowerLeg	tibia_l_imu
Leftfoot	calcn_l_imu

Table 4: List of sensors used and their respective body segments.

Once all the input data have been stored and saved in the correct way, OpenSim *OpenSense Tool* can be used to analyze kinematics data measured by IMUs.
Firstly, a static calibration was performed to connect and register each IMU sensor with a body segment of the OpenSim model created for each subject by the common workflow previously introduced. This step can be performed either directly from the OpenSim GUI or by using the OpenSim package on Matlab and a specially created function called **OpenSense_CalibrateModel_fx.m**, both leading to the same result; in the present project the second option was used since it is faster, but for the purpose of explanation the procedure on OpenSim GUI is more illustrative.

[Fig. 31] depicts the first option's settings.



IMU Placer Tool

IMU Orientation Data & Transformation

Orientation file at placement pose: ion\H02\InCap\Orientations\walk_008_orientations.sto

Space fixed Euler angles (XYZ order) from IMU space to OpenSim

Rotations X, Y, Z (degrees): -90 0 0

IMU Placement/Heading Correction (Optional)

Align IMU (name, axis): pelvis_imu -z with model's forward heading.

Figure 31: Example of IMU Placer Tool settings.

The tool used is the **IMU Placer Tool**: as input, it requires the model and the orientations file .sto previously generated to find the initial orientation of the IMU frame relative to the OpenSim model's body segments. To align the forward direction of the IMU data and the model's one, it was set the IMU forward direction following the z-axis of the *pelvis_imu sensor*. Moreover, since the reference system of Xsens is different from the OpenSim one, a rotation of -90° along the x-axis was performed to align the y-axis of the IMU Xsens and OpenSim reference systems [Fig. 32].

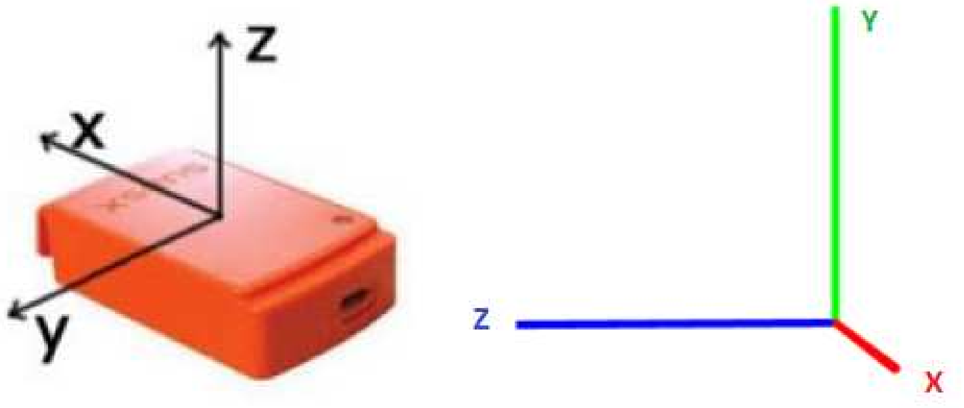


Figure 32: On the left is the coordinate system of an IMU sensor [39], and on the right is the OpenSim ground frame.

As output, [Fig. 33] shows the new model updated with IMU sensors correctly oriented with respect to each body segment.

As the next step, the inverse kinematics was calculated using the data collected from IMUs. Similar to the above, two approaches are available, one using OpenSim GUI and the other the tool implemented on Matlab; again, in the present study the tool implemented on Matlab was used, but for the purpose of explanation the procedure using OpenSim GUI will be illustrated, and [Fig. 34] shows the tool settings.

The **IMU Inverse Kinematics Tool** was used to find the set of joint angles for each time step of the walking motion, which minimizes the discrepancies between the experimental IMU orientations and the calibrated model's IMU frame. As input the calibrated model and the sensor orientations file were needed, specifying the Euler angles rotations to adopt to correctly transform from the IMU space to OpenSim.

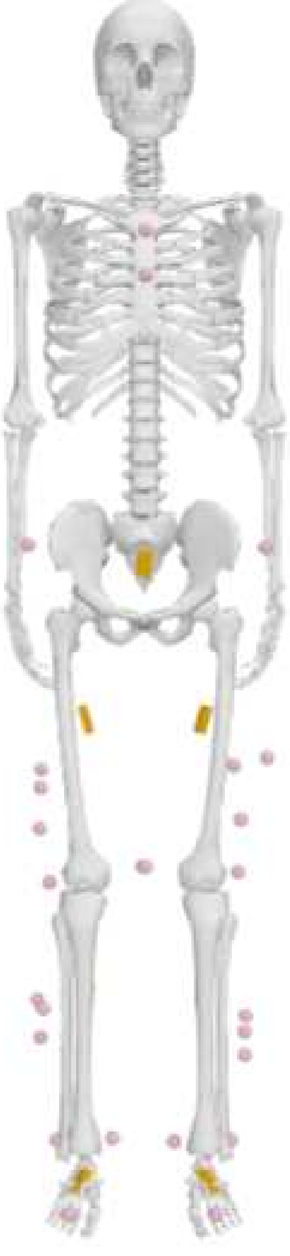
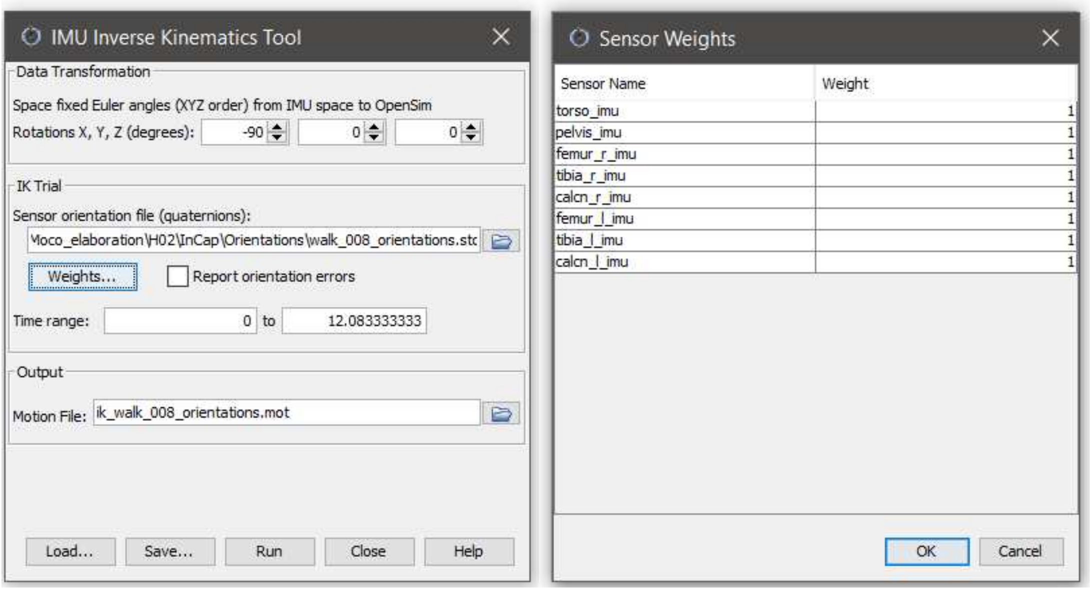


Figure 33: IMU Placer Tool model updated with IMUs with the correct orientation on each body segment. Notice that, to avoid confusion, all muscles and contact geometries have been hidden.

At first, the same rotations as the one used in the previous step for the IMU placer tool were adopted (i.e., a rotation of -90° along the x-axis). However, the output motion file containing the joint angles and/or translations calculated by the IMU IK Tool, revealed that the forward direction of the inertial capture system was not the same as the motion capture one: in fact, there was an angle between the IMU forward direction (in particular, along the z axis of the pelvis sensor) and the MoCap forward direction.



IMU Inverse Kinematics Tool

Data Transformation

Space fixed Euler angles (XYZ order) from IMU space to OpenSim

Rotations X, Y, Z (degrees): -90 0 0

IK Trial

Sensor orientation file (quaternions): Moco_elaboration\H02\InCap\Orientations\walk_008_orientations.stc

Weights... □ Report orientation errors

Time range: 0 to 12.083333333

Output

Motion File: ik_walk_008_orientations.mot

Sensor Weights

Sensor Name	Weight
torso_imu	1
pelvis_imu	1
femur_r_imu	1
tibia_r_imu	1
calcn_r_imu	1
femur_l_imu	1
tibia_l_imu	1
calcn_l_imu	1

Figure 34: Example of IMU InverseKinematics Tool. It should be noted that the same rotations as for the IMU Placer Tool are used. On the right, it is shown that each sensor has the same weight to resolve the IK problem.

In order for the two reference systems to coincide, especially the forward direction, a further modification on the angles between the two systems was necessary: from the IK file just calculated by the Tool, the first calculated *pelvis_rotation angle* is called the **heading angle**, and it specifically represents the angle between the two forward directions (i.e., z axis of the pelvis sensors) existing between the two reference systems.

Once known, a new IK file was calculated using the heading angle value in addition to the previous information.

To summarize, the rotation angles specified in the section *"Data Transformation"* of the IMU Inverse Kinematics Tool [Fig. 34], precisely are:

1. A rotation of -90° around the x-axis for each subject;

2. A rotation of a specific angle (heading angle) around the y-axis, as result in the conversion of the quaternions previously described. This angle is different for every walking trial of each subject.

The new output will contain final the joint angles and/or translations calculated by the IMU IK Tool.

MoCap Moco workflow As mentioned earlier, although the focus of this thesis is on the use of inertial sensors, data from MoCap system were also processed and analyzed. The data used initially consisted on the .c3d files for each walking trial, a standard format that contains all the information needed to read, display, and analyze 3D MoCap data with additional analog data from force plates, electromyography, accelerometers, and other sensors. Instead of using custom-built Matlab scripts, Nexus 2.12 was used to export *Track row Column* (.trc) and *motion* (.mot) formats containing respectively the marker trajectories coming from the camera system and ground reaction forces from the force plates. In addition, a .csv file was exported containing the time instants of the main gait events (i.e., Heel-strike, Toe-off).

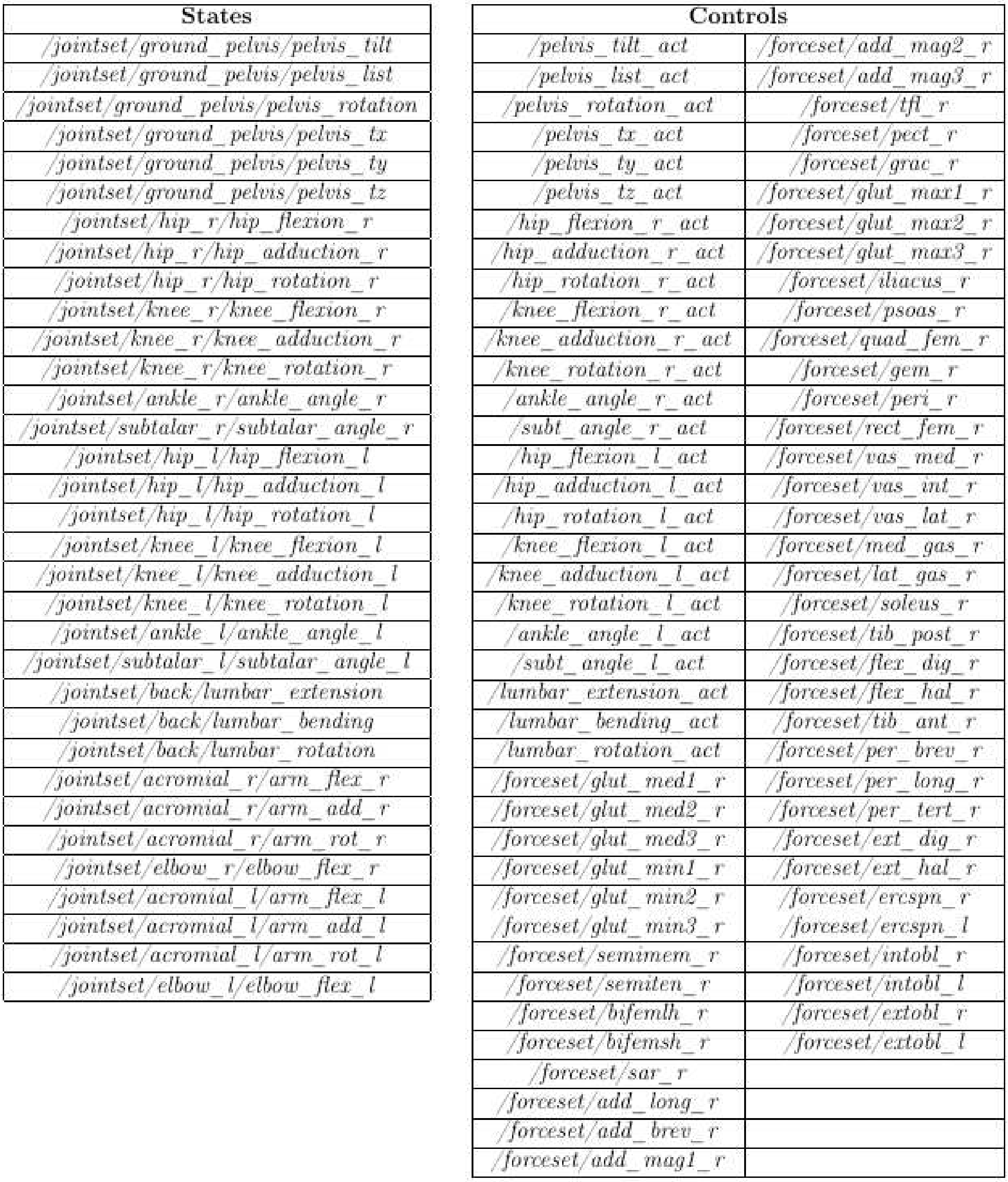
Each subject's walking trial file (.c3d) was analyzed prior to the export process to verify that the acquisition was valid. It had to be ensured that each marker was properly labeled, that there were no gaps in any marker trajectories, and, most importantly, that each foot was perfectly inside the force platform, with the ground reaction forces correctly detected.

After this operations, the joint angles for the scaled musculoskeletal models were calculated using the **InverseKinematics Tool** previously introduced, in order to obtain the .mot file containing all the information regarding the motion.

Moco setup The following section will explain the Moco approach chosen for this study. First, it is necessary to define states and controls that Moco considers in the musculoskeletal model used. In particular, the DOFs of the joints on the model, both angular values and velocities, were defined as **states**. Regarding **controls**, the muscles present on the model in terms of activations and fiber lengths, and the value of the added reserve actuators were assigned.
The two tables shown in [Fig. 35] list respectively the names of states and controls used.

As previously mentioned, data from the MoCap and InCap approaches are used in the following project, in particular IK files. Yet, .mot files previously produced with the appropriate tools require further modification to be employed on Moco: in particular, a *storage* (.sto) file must be used, and in order to correctly assign angle and velocity values to the states it is necessary to change the column names of that file.
This modification was made through a purpose-created Matlab script (**changeNames.m**) shown in *Appendix A*, for both IK file from MoCap and InCap approaches.

The **OpenSim Moco Tool**, as mentioned earlier, can be used via several programming languages; in this project, Matlab R2022a was used.
First, it was necessary to provide the inputs required by MocoTrack in order to obtain the desired solution: in particular, a musculoskeletal model and a file containing the motion to be tracked were needed. For this purpose, the Matlab *ModelProcessor* and *TableProcessor* commands were used, respectively; particularly, *ModelProcessor* command accept a base model and allow easily modifications to the model [22]. The same can be applied regarding the *TableProcessor* command with respect to IK file given as input. Both modifications regarding model and IK file are performed by operators (i.e., *ModelOperators* and *TableOperators*).



States
/jointset/ground_pelvis/pelvis_tilt
/jointset/ground_pelvis/pelvis_list
/jointset/ground_pelvis/pelvis_rotation
/jointset/ground_pelvis/pelvis_tx
/jointset/ground_pelvis/pelvis_ty
/jointset/ground_pelvis/pelvis_tz
/jointset/hip_r/hip_flexion_r
/jointset/hip_r/hip_adduction_r
/jointset/hip_r/hip_rotation_r
/jointset/knee_r/knee_flexion_r
/jointset/knee_r/knee_adduction_r
/jointset/knee_r/knee_rotation_r
/jointset/ankle_r/ankle_angle_r
/jointset/subtalar_r/subtalar_angle_r
/jointset/hip_l/hip_flexion_l
/jointset/hip_l/hip_adduction_l
/jointset/hip_l/hip_rotation_l
/jointset/knee_l/knee_flexion_l
/jointset/knee_l/knee_adduction_l
/jointset/knee_l/knee_rotation_l
/jointset/ankle_l/ankle_angle_l
/jointset/subtalar_l/subtalar_angle_l
/jointset/back/lumbar_extension
/jointset/back/lumbar_bending
/jointset/back/lumbar_rotation
/jointset/acromial_r/arm_flex_r
/jointset/acromial_r/arm_add_r
/jointset/acromial_r/arm_rot_r
/jointset/elbow_r/elbow_flex_r
/jointset/acromial_l/arm_flex_l
/jointset/acromial_l/arm_add_l
/jointset/acromial_l/arm_rot_l
/jointset/elbow_l/elbow_flex_l

Controls	
/pelvis_tilt_act	/forceset/add_mag2_r
/pelvis_list_act	/forceset/add_mag3_r
/pelvis_rotation_act	/forceset/tfl_r
/pelvis_tx_act	/forceset/pect_r
/pelvis_ty_act	/forceset/grac_r
/pelvis_tz_act	/forceset/glut_max1_r
/hip_flexion_r_act	/forceset/glut_max2_r
/hip_adduction_r_act	/forceset/glut_max3_r
/hip_rotation_r_act	/forceset/iliacus_r
/knee_flexion_r_act	/forceset/psoas_r
/knee_adduction_r_act	/forceset/quad_fem_r
/knee_rotation_r_act	/forceset/gem_r
/ankle_angle_r_act	/forceset/peri_r
/subt_angle_r_act	/forceset/rect_fem_r
/hip_flexion_l_act	/forceset/vas_med_r
/hip_adduction_l_act	/forceset/vas_int_r
/hip_rotation_l_act	/forceset/vas_lat_r
/knee_flexion_l_act	/forceset/med_gas_r
/knee_adduction_l_act	/forceset/lat_gas_r
/knee_rotation_l_act	/forceset/soleus_r
/ankle_angle_l_act	/forceset/tib_post_r
/subt_angle_l_act	/forceset/flex_dig_r
/lumbar_extension_act	/forceset/flex_hal_r
/lumbar_bending_act	/forceset/tib_ant_r
/lumbar_rotation_act	/forceset/per_brev_r
/forceset/glut_med1_r	/forceset/per_long_r
/forceset/glut_med2_r	/forceset/per_tert_r
/forceset/glut_med3_r	/forceset/ext_dig_r
/forceset/glut_min1_r	/forceset/ext_hal_r
/forceset/glut_min2_r	/forceset/ercspn_r
/forceset/glut_min3_r	/forceset/ercspn_l
/forceset/semimem_r	/forceset/intobl_r
/forceset/semiten_r	/forceset/intobl_l
/forceset/bifemlh_r	/forceset/extobl_r
/forceset/bifemsh_r	/forceset/extobl_l
/forceset/sar_r	
/forceset/add_long_r	
/forceset/add_brev_r	
/forceset/add_mag1_r	

Figure 35: List of states, on the left, and controls, on the right, names. To avoid confusion, only the name of the individual states (and controls) has been reported, but they should be considered in terms of respectively angular values and velocities for states, and activations and fiber lengths for controls.

In the present work, the following modifications to the model and the IK file were applied:

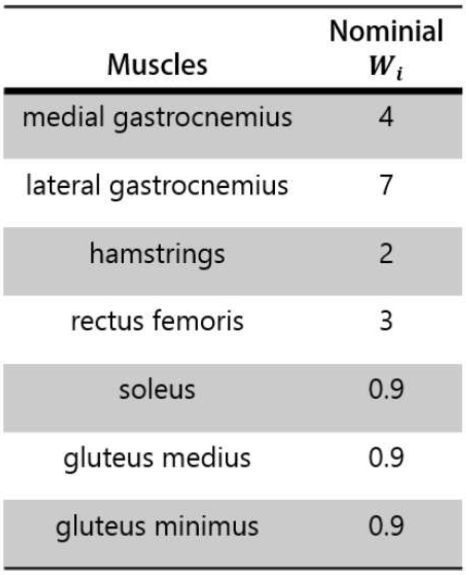
- ***ModOpIgnoreTendonCompliance()***: turn off tendon compliance for all muscles in the model;
- ***ModOpReplaceMusclesWithDeGrooteFregly2016()***: replace muscles of other types (particularly, *Millard2012EquilibriumMuscles* and *Thelen2003Muscles*) in the model with muscles of this type;
- ***ModOpIgnorePassiveFiberForcesDGF()***: turn off passive fiber forces for all *DeGrooteFregly2016Muscles()* in the model;
- ***ModOpScaleActiveFiberForceCurveWidthDGF()***: scale the active fiber force curve width for all *DeGrooteFregly2016Muscles* in the model;
- ***TabOpLowPassFilter()***: apply a low-pass filter to the trajectory.

After the inputs were set correctly, weights were set for the two (2) cost functions implemented by Moco. Regarding the latter, the following cost functions were applied in the present project:

1. **MocoTrackingGoal**: the default MocoTrack cost function, minimize the error between provided reference data and the associated model quantities (i.e., joint angles, joint velocities, marker positions, etc);
2. **MocoControlGoal**: minimize the sum of the absolute value of the controls raised to a given exponent, integrated over the phase [22].

For both, specific weight sets had to be set in order to obtain movement tracking as close as possible to the original input data and estimated muscle activations as close as possible to those measured.

Regarding the first cost function, global weights of value ten (10) were set for all DOFs; however, in order to obtain deviations from the input data as small as possible, much higher weights (in particular, value two hundred (200)) were set on the DOFs of the leg under investigation (i.e., the right leg).
The same reasoning was applied for the weights concerning the MocoControlGoal cost function: a value of ten (10) was set as the weight assigned globally to all controls, while muscle groups of interest to the results were set custom weights as shown in [Fig. 36].



Muscles	Nominial w_i
medial gastrocnemius	4
lateral gastrocnemius	7
hamstrings	2
rectus femoris	3
soleus	0.9
gluteus medius	0.9
gluteus minimus	0.9

Figure 36: List of muscles and respective weights used in MocoControl Goal. It should be noticed that the reciprocal of the w_i value shown in the figure was used, assigning these controls a weight of $1/w_i$.

Next, the settings required by the MocoProblem built into the MocoTrack package were set for the purpose of solving the problem. Here, in addition to entering the start and end times of analysis (particularly, heel-strike and toe-off of the right foot was set respectively as start and end times), three particular commands have been added:

- ***set_ allow_ unused_ references()***: if *true*, this setting allows extra data columns contained in the states reference that don't correspond to model coordinates;
- ***set_ track_ reference_ position_ derivatives()***: if *true*, this setting is enabled to fill in the missing coordinate speed data using the derivative of splined position data;
- ***set_ apply_ tracked_ states_ to_ guess()***: if *true*, this setting set the value of the *apply_ tracked_ states_ to_ guess* property.

The next step was to set bounds, that is, ranges of values within states and controls produced as output by Moco must be. In this study, only bounds for states were applied, so as to impose an additional constraint to limit deviations from the data provided as input.

Finally, by invoking the MocoSolver solver present in the MocoTrack package, it was possible to launch the MocoTrack analysis, obtaining as output the *Moco_ solution.sto* file containing the tracked motion (in particular, the values of joint angles and their respective velocities), and the values of muscle activations and length of muscle fibers.

Forces extraction As introduced earlier, the force values that the ground exerts on the foot during walking are collected by the five (5) spheres placed on it.
As the Moco analysis produced a new motion (tracked), force values were generated and recorded within the contact geometries placed on the right foot. Through Moco's **createExternalLoadsTableForGait()** function, these values can be extracted: it is necessary to provide the newly evaluated model states and the name of the spheres in order for this function to produce a *Moco_ GRF_ solution.mot* file containing ground reaction forces and moments estimated with respect to the global reference system.

CoP evaluation However, the force values recorded by the spheres are collected with respect to the origin of the reference system, thus implying that the forces do not act on the foot. For this reason, it was necessary to use the **Zero Moment Point (ZMP)** to calculate the coordinates of the center of pressure.
The ZMP is a concept in robotics and control theory that is used to determine the stability of a humanoid robot or any other object: it is defined as the point on the ground where the sum of all forces is equal to zero, and the sum of all moments around that point is also equal to zero [46]. This approach is important for maintaining balance and stability, as it indicates the location where the center of mass should be located to prevent it from falling over. For this reason, the ZMP method was exploited in this thesis for the estimation of the point of application of the GRFs, because it represents the point on the ground where the resultant tangential moments of the active forces are equale to zero.

The position of ZMP can be calculated using the conditions:

$$M_z = 0$$

$$M_x = 0$$

where x is the walking direction, z is the lateral direction, and y is the vertical direction. Basically, ZMP computations use a force and moment balance to transform a set of forces and moments applied at any point on the system to a dynamically equivalent set applied at the ZMP, which is a globally defined point on the ground [46].

Given a set of forces and moments $(\overline{F_p}, \overline{M_p})$ applied at the pelvis $(\overline{r_p})$, the equivalent forces and moments $(\overline{F_{ZMP}}, \overline{M_{ZMP}})$ applied at the ZMP $(\overline{r_{ZMP}})$ results as:

$$\overline{F_{ZMP}} = \overline{F_p}$$

$$\overline{M_{ZMP}} + \overline{r_{ZMP}} \times \overline{F_{ZMP}} = \overline{M_p} + \overline{r_p} \times \overline{F_p}$$

Since the $\overline{F_{ZMP}}$ and $\overline{M_{ZMP}}$ are applied at the ground, $\overline{M_{ZMP}}$ consists of only the vertical component, such that $\overline{M_{ZMP}} = (0, M_{ZMP}, 0)$ and $\overline{r_{ZMP}} = (X_{ZMP}, 0, Z_{ZMP})$, where X_{ZMP} and Z_{ZMP} are defined as below:

$$X_{ZMP} = f(MZ_p, FX_p, FY_p) = X_p + \frac{(MZ_p - Y_p FX_p)}{FY_p}$$

$$Z_{ZMP} = g(MX_p, FZ_p, FY_p) = Z_p - \frac{(MX_p + Y_p FX_p)}{FY_p}$$

By using these equations, the resultant active moment at ZMP along the y-axis is obtained as follows:

$$MY_{ZMP} = g(FX_p, FY_p, FZ_p, MX_p, MY_p, MZ_p) = MY_p + \frac{FX_p MX_p}{FY_p} + \frac{FZ_p MZ_p}{FY_p}$$

In this study, the assumption made to apply this method was that $F_p(FX_p, FY_p, FZ_p)$ and $M_p(MX_p, MY_p, MZ_p)$ represent the set of forces and moments equivalent to the estimated GRFs and GRMs by the spheres; in this way, it was possible to get the center of application of the results obtained, correctly positioned in space.

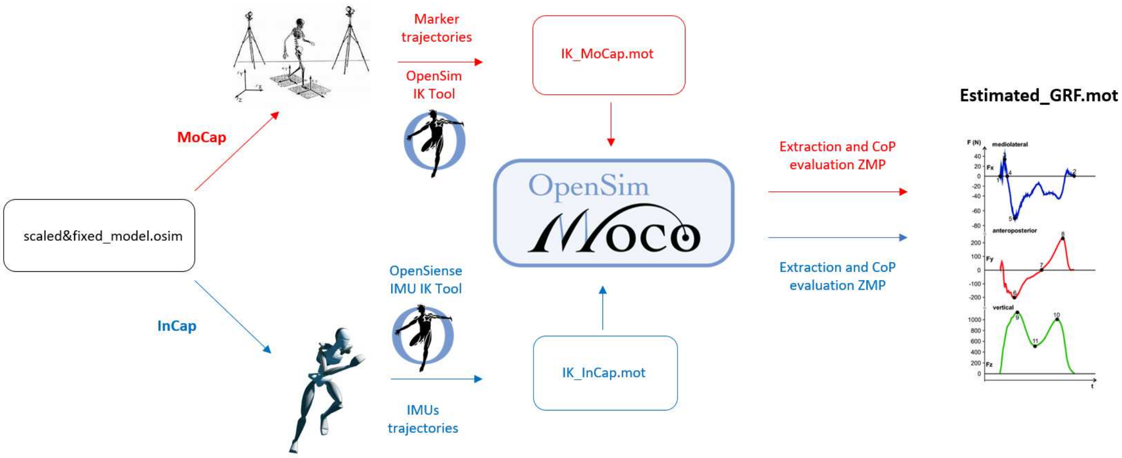


Figure 37: The Moco workflow, both for InCap and MoCap approaches.

Chapter 3

Results

All the results obtained using the InCap workflow introduced in the previous chapter (*Methods*) will be displayed here for a first visual comparison. In this study, referring to [Tab. 1], two (2) subjects were evaluated (in particular, H01 and H02) for a total of eight (8) walking tasks studied. To allow a more accurate comparison of the results, the average and the respective standard deviation (std) were calculated across all subjects, with each subject distinguished by the average over the specific number of trials of each participant's leg under investigation (particularly, the right one); by this, a clearer graphical comparison of the outcomes was possible.

The calculation was computed in Matlab R2022a: firstly, given that the displayed time duration might be expressed as percentage of the portion of gait cycle under investigation (in this study, the stance phase), all movements were time normalized with the Matlab function **TimeNormalize.m** as one hundred and one (101) data point at one (1) % intervals of the gait cycle; once this adjustment, it was possible to calculate the mean and the std of the data.

Moreover, in order to better organize the results obtained and the reference data with which to compare them, Matlab objects defined as follows were created:

$$results(n \times m \times l)$$

where n represents the number of rows in each matrix (equal to 101, since all results were normalized), m the number of columns in each matrix varying according to the type of comparison (IK: 15, ID: 15, GRF: 10), and l the number of tasks analyzed. By this, six (6) matrix were obtained for IK, ID, and GRF comparisons.

As a legend for the graphs, red always indicates the reference data, and blue always refers to the results obtained from the InCap approach.

The x-coordinate corresponds to the stance phase of the gait cycle as a percentage, and the ordinate axis represents the value of the variables under analysis:

- **Degrees (°)** for joint angles (IK);
- **BW** for ground reaction forces;
- **meters (m)** for center of pressure coordinates;
- **Nm/kg** for joint moments (ID).

For the IK and ID results, the graphs are presented in such a way that the rows correspond to a joint and the columns to a type of movement. The first row is dedicated to the hip, the second to the knee, and the third to the ankle. As for the columns, the first is dedicated to flexion/extension (FE), the second to abduction/adduction (AA), and the third to intra/extra rotation (IE). Regarding the results obtained from the spheres and the respective reference data, the anterior/posterior (AA), vertical (V), and medio/lateral (ML) components will be plotted with regard to the forces.

In addition to the graphs, some coefficients were calculated in order to provide more resources to evaluate the accuracy of the results obtained.

Specifically:

1. The **Pearson's correlation coefficient** was used to measuring the *linear correlation* between reference and results;
2. The **Root Mean Square Error (RMSE) value**, as well as the maximum and median differences;
3. The **Maximum value of the difference** between reference and results.

These values will be listed in specific tables for IK, ID, and GRF comparisons.

The same graphs and coefficients for MocoTrack MoCap results were also evaluated; these can be found in *Appendix B*.

3.1 Kinematics: reference vs. MocoTrack InCap

The comparison of the joint angle values obtained from the MocoTrack analysis and the values computed with IMU orientations was done using the **plot_distribution.m** Matlab function (particularly, it represents data in terms of mean and std). [Fig. 38] shows the comparison of the following lower limb joints: hip, knee, and ankle; flexion/extension, adduction/abduction, and internal/external rotation are all taken into account (except for the ankle, which only has dorsi/plantar flexion DOF).

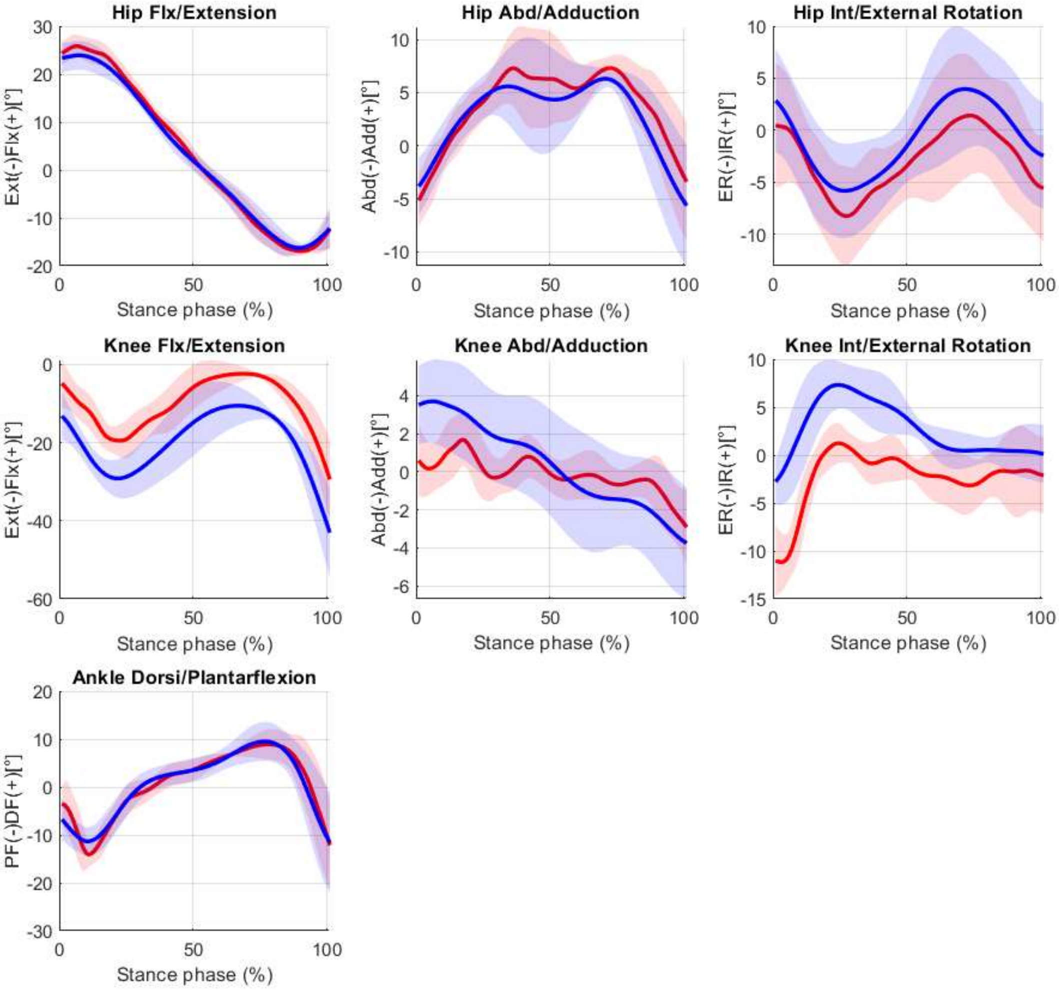


Figure 38: Kinematics comparison: in red are the InCap data, while in blue are the tracked Moco results obtained with InCap data as input. The continue lines represent the mean (± std) across the 2 subjects both for InCap and Moco results.

From a graphical perspective, the sagital plane (specifically, the flex/extension degree of freedom) appears to be well plotted in the Moco results, while the other planes show deviations from the input data.

The Pearson correlation between each subject and joint angle was calculated and it was also reported the mean and standard deviation for the correlation coefficients, RMSE, and maximum difference [Tab. 5].

	mean ± std		
	Correlation coefficient	**RMSE [°]**	**Maximum difference [°]**
Hip FE	0.99 ± 0.003	1.65 ± 0.41	4.41 ± 0.77
Hip AA	0.96 ± 0.015	0.98 ± 0.15	3.59 ± 0.045
Hip IE	0.99 ± 0.001	0.87 ± 0.15	2.83 ± 0.33
Knee FE	0.93 ± 0.026	10.94 ± 0.85	16.78 ± 2.05
Knee AA	0.74 ± 0.1353	2.36 ± 0.16	5.19 ± 0.65
Knee IE	0.85 ± 0.1013	5.12 ± 2.76	11.88 ± 6.85
Ankle DP Flex	0.92 ± 0.005	3.20 ± 1.02	11.86 ± 2.98

Table 5: Average value (± std) of the correlation coefficient (Pearson value), root mean square error (RMSE), and maximum value of the difference between each joint angle of the reference (InCap approach) and MocoTrack result. It is the average between subjects (**inter-subjects**), each one characterized by the average of their walking trials (**intra-subjects**).

Average RMS differences between InCap data and MocoTrack results were in a range of values 1-6° for all joint angles except knee flexion/extension (around 11°). Still concerning the knee joint, this presented the lowest correlation value with regard to abd/adduction (r = 0.74), while in terms of maximum differences, values around 11° were also found with regard to the dorsi/plantarflexion of the ankle [Tab. 5].

3.2 Ground reaction forces: measured vs. estimated InCap

The estimated GRFs extracted from the values recorded by the contact spheres placed on the right foot are compared in [Fig. 39] with the forces recorded by the force platforms in the MoCap system. Both are represented with respect to the global reference system.

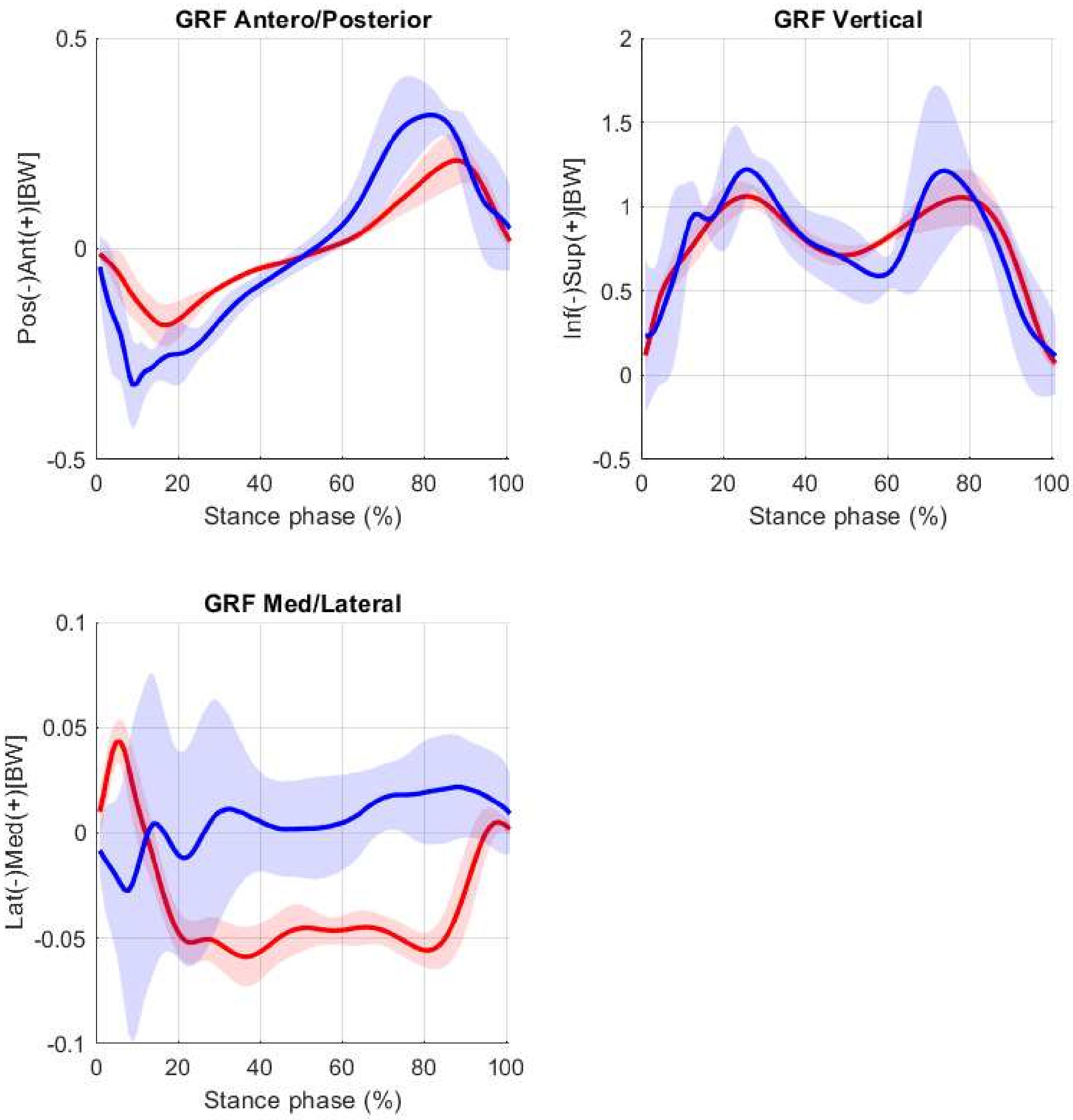


Figure 39: GRFs comparison: in red are the MoCap measured values, while in blue are the Moco estimated results obtained with InCap data as input. The continue lines represent the mean (± std) across the 2 subjects for both measured and estimated results.

From the graphical representation shown in [Fig. 39], it is possible to appreciate that the estimated data exhibit a great deal of variability. Focusing on the mean values, in terms of the anterior/posterior and vertical components the values obtained turn out to be comparable with those measured, while the same cannot be said for the medio/lateral component.

Confirming the findings of [Fig. 39], in [Tab. 6] are reported the mean and standard deviation for the correlation coefficients, the root mean square error (RMSE), and maximum difference values.

	mean ± std		
	Correlation coefficient	**RMSE [BW]**	**Maximum difference [BW]**
AP Force	0.93 ± 0.047	0.10 ± 0.019	0.26 ± 0.078
V Force	0.75 ± 0.013	0.25 ± 0.061	0.71 ± 0.3421
ML Force	-0.20 ± 0.240	0.06 ± 0.027	0.13 ± 0.008

Table 6: Average value (± std) of the correlation coefficient (Pearson value), root mean square error (RMSE), and maximum difference between the measured GRFs by the MoCap system (particularly, by the force plates) and the estimated one by using the contact geometry model and IMUs data. It is the average between subjects (**inter-subjects**), each one characterized by the average of their walking trials (**intra-subjects**).

Moco solution showed an average (± std) RMSE of the vertical, anterior/posterior, and medio/lateral ground reaction forces of 0.25 ± 0.0061 BW, 0.10 ± 0.019 BW, 0.06 ± 0.027 compared to measured data [Tab. 6]. In addition, a strong correlation was found for the anterior/posterior and vertical components (respectively, r = 0.93 and r = 0.75), while a negative correlation emerged for the middle/lateral component (r = -0.2) [Tab. 6]. Finally, mean values of maximum differences between measured and estimated data for the three components of ground reaction forces result in respectively 0.26 for the anterior/posterior component, 0.71 for te vertical component, and 0.13 for the medio/lateral component [Tab. 6].

3.3 Center of pressure: measured vs. estimated InCap

In the second chapter (*Methods*), it had been discussed about the problem of estimating the CoP of the estimated ground reaction forces for the inertial-based approach. This represents one of the limitations found in the present approach: a discrepancy was found between the coordinates measured by the CoP force platforms and those calculated by the zero moment point method from ground reaction forces and moments values estimated from the contact spheres. [Tab. 7] shows these differences in terms of correlation coefficient, and root mean square error (RMSE).

	mean ± std	
	Correlation coefficient	**RMSE (m)**
CoP AP Force	0.93 ± 0.07	0.067± 0.09
CoP ML Force	-0.02 ± 1.37	0.038 ± 0.035

Table 7: Average value (± std) of the correlation coefficient (Pearson value), and root mean square error (RMSE) between the measured center of pressure by the MoCap system (particularly, by the force plates) and the ZMP evaluated by using InCap estimated results over all subjects. It is the average between subjects (**inter-subjects**), each one characterized by the average of their walking trials (**intra-subjects**).

[Tab. 7] revealed a strong correlation (r = 0.93) between the CoP trends in the anterior/posterior direction, but a poor negative correlation (r = -0.02) in the medio/lateral direction. However, RMSE values show how obvious discrepancies exist between measured data and the ones calculated from estimated values [Tab. 7].

3.4 Dynamics: reference vs. MocoTrack InCap

Finally, [Fig. 40] shows joint moments for the hip, knee, and ankle joints during the overground walking, obtained from the OpenSim InverseDynamics Tool. As explained above, in red are the outcomes from the MoCap approach that used the kinematics and GRFs measured by the stereophotogrammetric system as input, while in blue the inertial derived kinematics and estimated GRFs by Moco with the point of application obtained via ZMP calculations.

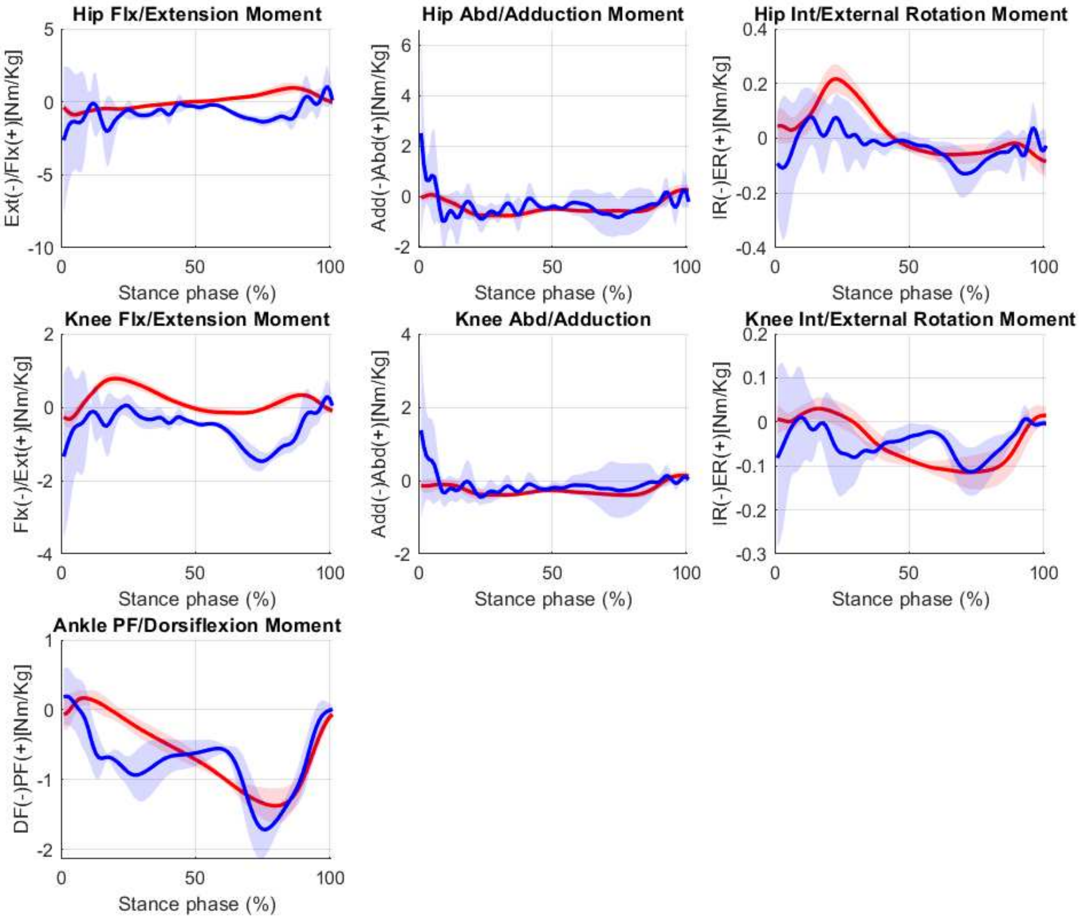


Figure 40: Dynamics comparison: in red are the MoCap system results, while in blue are the inertial-based ones using estimated GRFs. The continue lines represent the mean (± std) across the 2 subjects both for MoCap and InCap + estimated GRFs results.

As might be expected, in [Fig. 40] the blue curves turn out to be more approximate and variable; this may be due to the variable estimation of ground reaction forces and ground reaction moments. In terms of peak values, it is apparent from [Fig. 40] that some are overestimated (i.e., hip flexion/extension and abd/adduction) while others appear to be in line with the data provided as reference depending on what force is used to estimate joint moments: for instance, moments calculated from the medio/lateral component will lead to results more divergent from the reference results.

To confirm the description previously made just from the visual comparison, in [Tab. 8] are reported the mean and standard deviation for the correlation coefficient, and the root mean square error (RMSE) between reference and obtained data.

	mean ± std	
	Correlation coefficient	**RMSE [Nm/kg]**
Hip FE Moment	-0.18 ± 0.348	72.71 ± 34.11
Hip AA Moment	0.40 ± 0.050	42.10 ± 25.34
Hip IE Moment	0.51 ± 0.001	4.11 ± 3.57
Knee FE Moment	0.30 ± 0.003	45.11 ± 12.17
Knee AA Moment	0.50 ± 0.017	24.40 ± 14.81
Knee IE Moment	0.60 ± 0.298	3.88 ± 1.54
Ankle DP Flex Moment	0.63 ± 0.053	50.65 ± 2.99

Table 8: Average value (± std) of the correlation coefficient (Pearson value), and root mean square error (RMSE) between each joint moment of the reference (MoCap approach) and InCap + estimated GRFs result. It is the average between subjects (**inter-subjects**), each one characterized by the average of their walking trials (**intra-subjects**).

Furthermore, [Tab. 8] highlights the big difference between all flexion/extension moments of all DOFs. As for Pearson's coefficient, all the moments obtained are found to be moderately correlated with their respective reference values, resulting in a coefficient r varying in the range of r = 0.4 - 0.6. The only exception concerns the hip flex/extension moment, for which a negative correlation value of r = -0.18 was reported [Tab. 8].

In addition, high values of RMS differences emerged for moments mostly in the sagittal plane: the average (± std) RMSE of the hip and knee flexion/extension moments and the ankle dorsi/plantar flexion moment are respectively 72.71 ± 34.11 Nm/kg, 45.11 ± 12.17 Nm/kg, and 50.65 ± 2.99 Nm/kg [Tab. 8]. Finally, high RMSE values were also found for hip and knee abd/adduction, respectively, of 42.10 ± 25.34 Nm/kg, and 45.11 ± 12.17 Nm/kg [Tab. 8].

Chapter 4

Discussion

The findings illustrated in the previous chapter (*Results*) allow some considerations to be made about the use of IMUs coupled with models that have contact geometries in the analysis of human movements, particularly in estimating the ground reaction forces.

The goal of this thesis was to develop an approach that, from data derived from inertial wearable sensors IMUs, would allow the most correct estimation of ground reaction forces by eliminating the use of force platforms for their detection.

In general, optical motion capture systems with skin-mounted reflective markers are used to capture human motion kinematics. However, such a system allows the study of movements in a controlled environment (i.e., a motion laboratory) limiting the applications of motion analysis. For this reason, the approach presented in this thesis is an interesting development in order to obtain viable alternatives to the gold standard represented by the MoCap system.

Regarding the results, in order to be as illustrative as possible in the discussion of these, this chapter will be divided as follows:

- Discussion of Kinematics results;
- Discussion of Ground reaction forces results;
- Discussion of Center of pressure results;
- Discussion of Dynamics results;

In addition, as mentioned in the previous chapter (*Results*), the same results in terms of [Fig.] and [Tab.] are reported in this thesis by comparing reference data from the gold standard (specifically, MoCap system) and data processed through the operative workflow [Fig. 28] described above that use data from optical-based approach as input. These results can be found in *Appendix B*. However, the following comments can generally be considered the same for the results shown in *Appendix B*, with a few exceptions due to the differences present between the two data collection approaches.

4.1 Discussion of Kinematics results

As for the results obtained using the OpenSim Moco tool to track a movement provided as input, it can be stated that the joint angles obtained from this approach using IMUs data were consistent with the provided ones.

The RMS differences were all between 1-6°, except for knee flexion/extension where the values obtained are approximately 11° [Tab 5]. In addition, the maximum differences found on all DOFs had similar values for the hip joint, while for the knee and ankle joints higher values have been reported in a range of 11-16° [Tab. 5]. In order to justify these values, Moco developers [4] state that using MocoTrack automatically implies that the "tracked" motion has deviations from that provided as input: this is because Moco employs an optimization algorithm that minimizes a cost function in order to obtain a solution. However, these deviations must be contained within a certain range for the results obtained to be considered clinically employable: an error of less than 2° is acceptable in most clinical applications, while errors of 2° to 5° are also acceptable but require specific interpretation [17].

Such values obtained and listed in [Tab. 5] may find justification in the low weight assigned during this study to that DOFs during the Moco setup illustrated in the previous chapter (*Methods*): particularly, these thesis represents a first approach involving the use of this tool for such analyses, so a specific set of weights was preferred with regard to the pelvis and hip joint, leaving lower weights for the other joints. This choice was also justified by the computational time required for the analyses: increasing the weights assigned to the DOFs of the model results in a directly proportional increase in the time required by the solver to reach convergence.

Finally, the strong correlation values obtained along with the graphical representation provided in [Fig. 38] demonstrate that the tracked result obtained are well correlated with the provided ones as input and identified here as reference (specifically, data from InCap approach).

4.2 Discussion of Ground reaction forces results

The purpose of this thesis is to validate a new method involving the estimation of ground reaction forces based on a biomechanical model that has specific contact geometries suitable for recording values of forces acting on them. To this, the OpenSim Moco tool was coupled with the goal of tracking a motion provided as input from IMU measurements. Because this approach is totally novel, nothing has been found in the literature with which to compare and justify the results obtained. Thus, the following considerations arise from observations regarding the results obtained and critical sense of the candidate.

From the visual comparison provided by [Fig. 39], it is possible to establish how the approach presented provides an interesting starting point for developing a new workflow reliable as the gold standard for studying a wider range of movements in uncontrolled environments that require specific equipment. This is confirmed by the great values obtained for the coefficients presented in [Tab. 6] for the anterior/posterior and vertical components of GRFs: in fact, a strong correlation can be seen for both components (r = 0.93 and r = 0.75, respectively), and an average (± std) RMSE of 0.10 ± 0.019 BW, and 0.25 ± 0.061 BW.

The only negative point is the estimation of the medio/lateral component: it can be visually seen a discrepancy from the red curve representing the force values measured by the MoCap system and the blue curve representing the force values estimated by the spheres [Fig. 39]; this graphical consideration is confirmed by the values shown in [Tab. 6], which establish a negative correlation between reference and estimated data (r = -0.20) and a maximum difference value (mean ± std) of 0.13 ± 0.008 BW.

The reason for such a poor estimation of GRFs (in particular, of the medio/lateral component) can be totally attributed to the geometric configuration of the contact spheres and their mechanical properties: during the development of the musculoskeletal model chosen for the present project and illustrated in the second chapter (*Methods*), several configurations were tested in order to find the optimal one suitable for the purposes described above; for clarity, only the final configuration chosen that turned out to be the best among those tested was reported. However, from this preliminary study, it was found that a single change in a property of the spheres is responsible for a significant change in terms of estimated forces.

To give an example, changing the size of the spheres (in particular, their radius) resulted in overestimated values of forces recorded by the spheres due to the inversely proportional relationship between force and area of application (specifically, the smaller the area the greater the force recorded). In order to provide a visual representation of what has just been explained, [Fig. 41] shows the estimated forces (in red) of a single subject's walk (specifically, H01 walk_5) employing the musculoskeletal model presented by Falisse *et al* [40] previously introduced that specifically has six (6) contact spheres with different positioning, size, and mechanical properties from those employed in the present project.

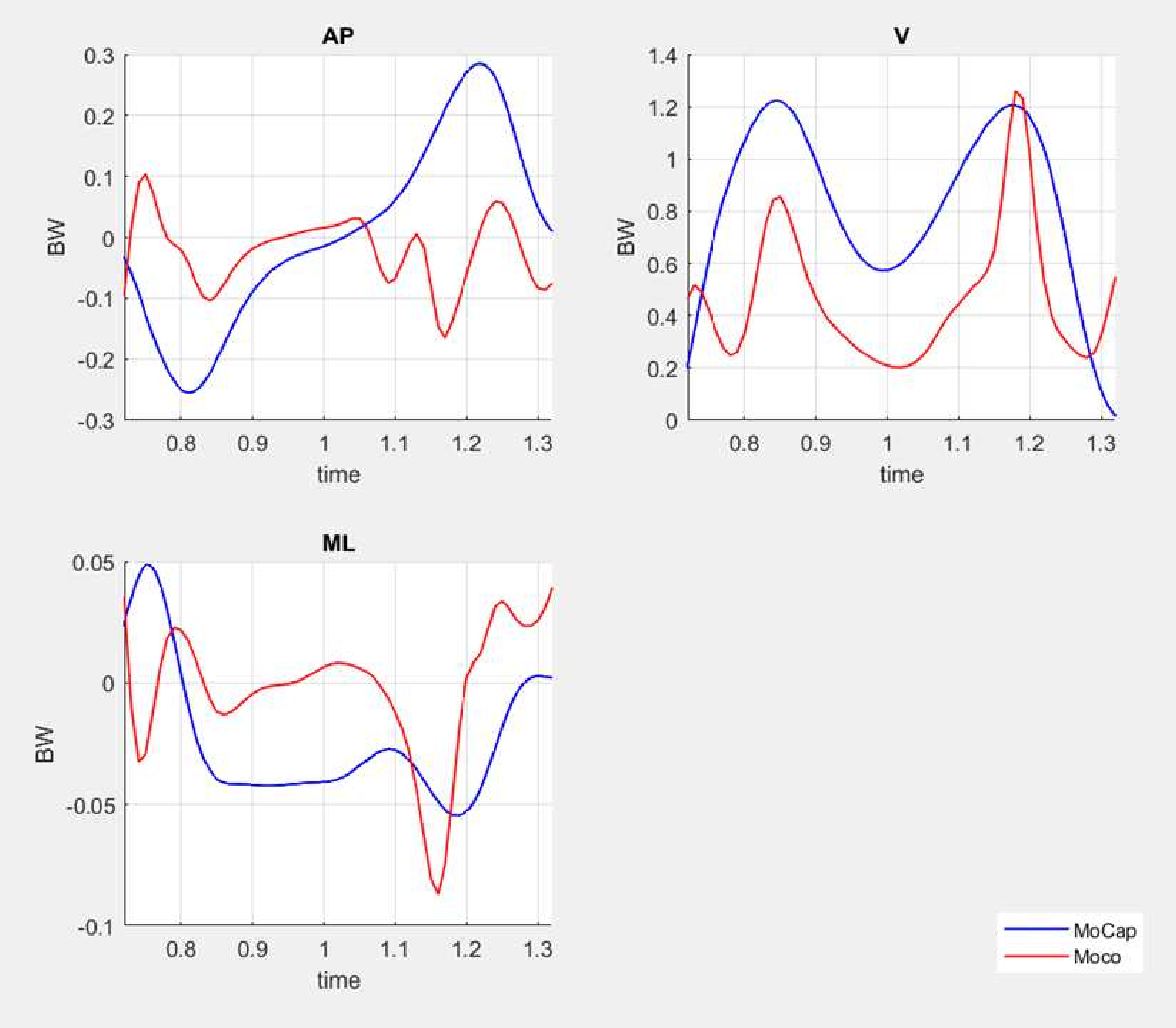


Figure 41: Graphic comparison between MoCap measured GRFs (in blue) and estimated GRFs (in red) by using the [40] model. The x-axis shows the time instants of the stance phase (specifically, from HS to TO).

From [Fig. 41], it is quite evident from the graphical point of view how changes made to the spheres directly result in changes in the estimated forces recorded by them.

For the considerations made above, the rough estimation of GRFs, particularly the poor estimation of the medio/lateral component, finds justification: although the configuration used including position, geometry, and mechanical properties is the best one obtained in the preliminary study of the present work, it is not to be considered definitive; on the contrary, it should be represented as a starting point of future analyses to improve the estimation of all three components of GRFs, improving both graphically the similarity with the reference curves and from the point of view of coefficients to obtain smaller errors and stronger correlations.

4.3 Discussion of Center of pressure results

In the previous section, it was explained how changes concerning the contact spheres cause important variations in the estimated values of forces and moments. This variability strongly influences the estimation of the center of pressure.

As explained in the second chapter (*Methods*), the **createExternalLoadsTableForGait.m** Matlab function contained in the OpenSim Moco package returns values of ground reaction forces and moments centered in the origin of the global reference system. In order to obtain the point of action of these forces, defined as the center of pressure (CoP), the zero moment point (ZMP) method was employed [46]. However, it is evident that providing inaccurate values to the ZMP method equations in order to calculate the anterior/posterior and medio/lateral CoP components yield results that are not in line with the same components collected by the force plates in the MoCap system. Such discrepancies are reported in [Tab. 7], where high RMSE values can be seen for both components. Surprisingly, the correlation coefficient for the anterior/posterior component appears to be remarkably strong (r = 0.93) [Tab. 7].
Although the components obtained cannot be considered comparable with the gold standard and consequently employable, based solely on the values reported in [Tab. 7] it would appear that the calculated CoP is reported outside the normal area of application (i.e., in this study, the right foot). [Fig. 42] intends to provide a graphical visualization of a single walk (specifically, H01 walk_5) in order to show how the CoP calculated from the values of GRFs and GRMs is correctly positioned in the application area.

Taking into account all the considerations made earlier about the correlation between contact spheres and force estimation, it is evident that the results obtained in the present study cannot be considered reliable and clinically applicable. However, as shown in [Fig. 42], these results represent a starting point for future studies to estimate forces, moments, and consequently their point of application more and more effectively.

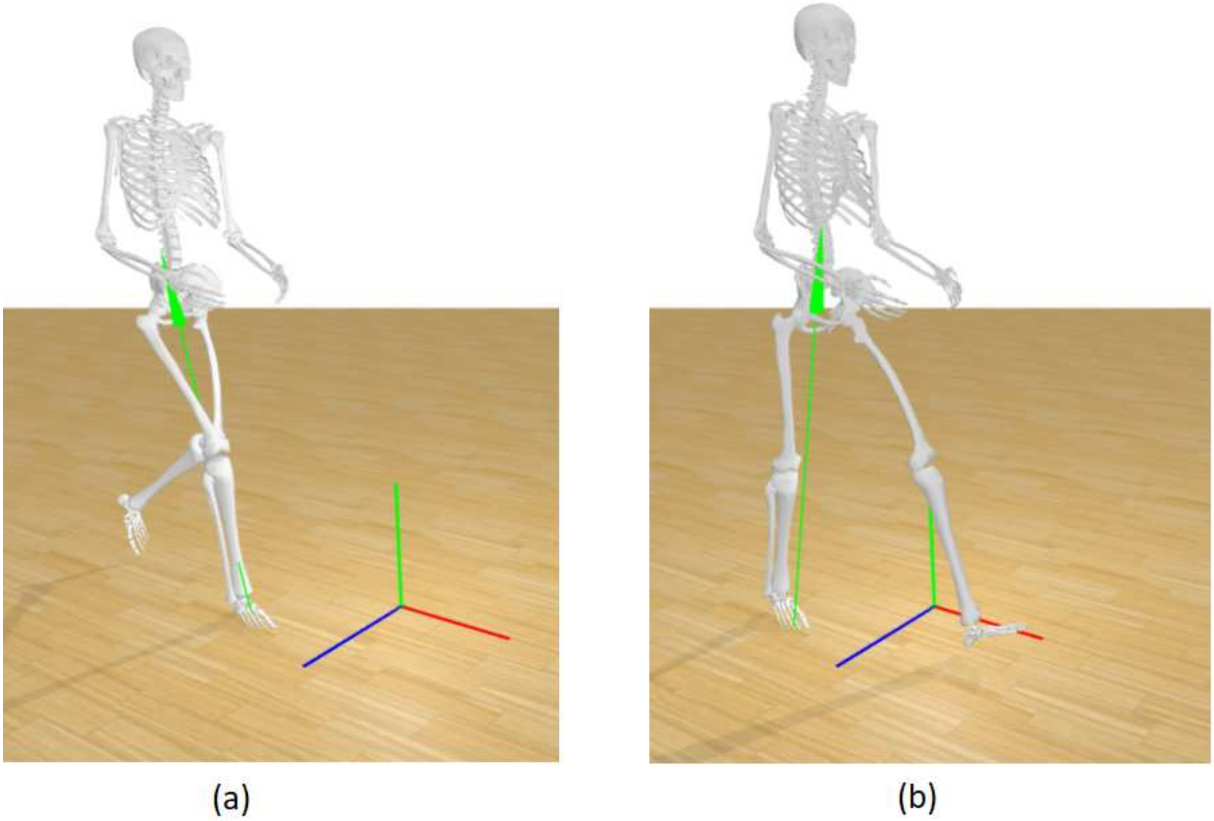


Figure 42: Visual representation of the estimated GRFs (in green) coupled with the evaluated CoP components with the ZMP method for subject H01, walk_5. In particular, (a) represents the forces at 25% of stance phase, while (b) the same at 75% of stance phase. To avoid confusion, all muscles, contact geometries, and markers placed on the model have been hidden.

4.4 Discussion of Dynamics results

Further confirmation of the considerations made earlier is evident by considering the dynamics results obtained using the ground reaction forces and moments data estimated from the spheres and their positioning calculated by the zero moment point method.

Graphically, [Fig. 45] clearly reports how the results from the estimated GRFs are highly variable: in particular, regarding the moments of hip flexion/extension (hip FE) and internal/external rotation (hip IE), it is possible to find anomalous peaks at the beginning with strongly overestimated values. This overestimation can be attributed to the inaccurate estimates of GRFs, GRMs, and CoP components, which, when added together, produce such errors. The same reasoning can be applied for the same degrees of freedom of the knee, were similar scenarios are depicted in the second row of [Fig. 45].
As confirmation of the graphs shown in [Fig. 40], the correlation coefficient (Pearson coefficient) and root mean square error (RMSE) were calculated and reported in [Tab. 8]. As expected, [Tab. 8] highlights the differences illustrated above, reporting mean (± std) RMSE values of 72.71 ± 34.11 BW for hip FE moment, and 42.10 ± 25.34 BW for hip abd/adduction (hip AA) moment. As before, the same higher values of RMSE were found regarding knee FE moment (45.11 ± 12.17 BW), knee AA moment (24.40 ± 14.81 BW), and finally ankle dorsi/plantar flexion (ankle DF) moment (50.65 ± 2.99 BW) [Tab. 8]. As for the Pearson's coefficient, all the moments obtained are found to be moderately correlated with respect to the reference values (in particular, the coefficient r varies in the range of r 0 0.4-0.6); the only exception concerns the hip FE moment, for which a negative correlation value was reported [Tab. 8].

The results just discussed represent yet another confirmation that the present project cannot be considered completed; on the contrary, further testing and development are needed in order to reduce the peaks present in [Fig. 40], consequently increasing the correlation between reference data and data obtained by minimizing the errors present.

Chapter 5

Conclusions and further developments

In this thesis, a new method for estimating ground reaction forces was developed based on a biomechanical model and data from an IMU-based approach. The ultimate goal is to develop a reliable and clinically employable alternative to the gold standard for the study of motion analysis, represented by the combination of retro-reflective markers and a stereophotogrammetric system, known as the motion capture (MoCap) system [2].

The results presented in the dedicated chapter (*Results*) clearly show how the approach developed in this project is promising: through a configuration of contact geometries designed specifically for the present study involving spherical elements with specific geometrical and mechanical characteristics ([Fig. 26], [Tab. 2], [Tab. 3]), it was possible to estimate all three components of ground reaction forces, the anterior/posterior, the vertical, and the medio/lateral components (shown in [Fig. 39]) and the ground reaction moments. In order to make the presented approach as least "optical-based" as possible, data from wearable IMU sensors were used in the present project, which provide an interesting alternative to the gold standard for motion detection during a specific task [18]. Finally, the OpenSim Moco tool allowed the two previously described tools to be coupled as inputs, allowing via the MocoTrack package to track a motion and detect the GRFs derived from it by reducing the problem to an optimization problem and using direct collocation as the solution method [4]. In addition to the main objective of this thesis (particularly, the estimation of GRFs), in the results chapter (*Results*), the results obtained were presented compared with the respective reference values of joint angles ([Fig. 38], [Tab. 5]), coordinates of the center of pressure ([Tab. 7]), and joint moments ([Fig. 40], [Tab. 8]).

Conclusions based on the results obtained and the discussion of these presented in the dedicated chapter (*Discussion*) can be drawn: the workflow developed and illustrated in [Fig. 28] represents a new method for the estimation of ground reaction forces that sees the use of a biomechanical approach that expands the variety of existing methods [1]. Although the results obtained have proved promising, many factors must be considered. In particular, the results estimated from the spheres have some discrepancies from the values measured from the force platforms; these discrepancies are directly responsible for errors in the subsequent calculation of center-of-pressure coordinates and joint moments. This shows how, despite the promising results, further improvements in this direction suitable for enhancing the reported estimated values are needed, since the gold standard devices still appear to be the most reliable.

To summarize, the approach developed and the results presented represent the beginning of a new method for estimating ground reaction forces.
Future developments and improvements will lead to validation of this method as an alternative to the gold standard, represented by direct measurements of such values, that will be reliable and clinically applicable. Among these, a sensitivity study devoted entirely to contact spheres is necessary, since through this it is possible to understand in depth how changes in geometrical and mechanical characteristics introduced during the discussion of the results lead to tangible changes in terms of estimated force values; based on this important study it will be possible to define the best configuration as a combination of geometry, position, number, and mechanical characteristics of the balls that produces the most reliable results. A further development concerns the model: in order to make it as comparable as possible with the human musculoskeletal system, it is necessary to add additional degrees of freedom to the joints (i.e., to the ankle joint) with the ultimate goal of employing such a model coupled with the developed method in the study of movements of patients with diseases that impair normal walking, such as osteoarthritis, and neurological diseases. Finally, after the previous improvements, since the OpenSim Moco package turns out to be open-source and therefore completely customizable, a new Moco package may be developed *ad-hoc* so as to better optimize solver performance, analysis, and results.

Appendix A: Code used

1. **changeNames.m**

The script reported in the following page was used to change the state names for both InCap and MoCap approach.

In addition, the following script is for the subject H01, suitable for the change of names of walk_5_ik, walk_8_ik, and walk_11_ik. The code remains unchanged for the subject H02, for the change of names of walk_7_ik, walk_9_ik, walk_10_ik, walk_12_ik, and walk_14_ik.

Change joint names with path to joint

```
close all; clear all; clc;

walk = 5; % CHANGE HERE TO SWITCH WALK! For HS1 = [5 8 11]

IK_path = 'C:\Users\lbot9\OneDrive - KU Leuven\Tesi_Ludovica\Moco_elaboration\H01\MoCap\IK\';
IK_file_mot = strcat('walk_', num2str(walk), '\walk_', num2str(walk), '_ik.mot');
IK_file_sto = [IK_path, strcat('walk_', num2str(walk), '\walk_', num2str(walk), '_ik.sto')];

[data, names, table] = readMOTSTOTRCfiles_v1(IK_path, IK_file_mot);
names_path = {'time', ...
    '/jointset/ground_pelvis/pelvis_tilt/value', ...
    '/jointset/ground_pelvis/pelvis_list/value', ...
    '/jointset/ground_pelvis/pelvis_rotation/value', ...
    '/jointset/ground_pelvis/pelvis_tx/value', ...
    '/jointset/ground_pelvis/pelvis_ty/value', ...
    '/jointset/ground_pelvis/pelvis_tz/value', ...
    '/jointset/hip_r/hip_flexion_r/value', ...
    '/jointset/hip_r/hip_adduction_r/value', ...
    '/jointset/hip_r/hip_rotation_r/value', ...
    '/jointset/knee_r/knee_flexion_r/value', ...
    '/jointset/knee_r/knee_adduction_r/value',    ...
    '/jointset/knee_r/knee_rotation_r/value', ...
    '/jointset/ankle_r/ankle_angle_r/value', ...
    '/jointset/subtalar_r/subtalar_angle_r/value', ...
    '/jointset/hip_l/hip_flexion_l/value',    ...
    '/jointset/hip_l/hip_adduction_l/value', ...
    '/jointset/hip_l/hip_rotation_l/value', ...
    '/jointset/knee_l/knee_flexion_l/value', ...
    '/jointset/knee_l/knee_adduction_l/value',    ...
    '/jointset/knee_l/knee_rotation_l/value', ...
    '/jointset/ankle_l/ankle_angle_l/value', ...
    '/jointset/subtalar_l/subtalar_angle_l/value', ...
    '/jointset/back/lumbar_extension/value', ...
    '/jointset/back/lumbar_bending/value',  ...
    '/jointset/back/lumbar_rotation/value', ...
    '/jointset/acromial_r/arm_flex_r/value', ...
    '/jointset/acromial_r/arm_add_r/value', ...
    '/jointset/acromial_r/arm_rot_r/value', ...
    '/jointset/elbow_r/elbow_flex_r/value',    ...
    '/jointset/elbow_r/pro_sup_r/value', ...
    '/jointset/acromial_l/arm_flex_l/value', ...
    '/jointset/acromial_l/arm_add_l/value',...
    '/jointset/acromial_l/arm_rot_l/value', ...
    '/jointset/elbow_l/elbow_flex_l/value', ...
    '/jointset/elbow_l/pro_sup_l/value'};
generateMotFile_v1(data, names_path, IK_file_sto)
```

2. MocoTrack_InCap.m

Only the script regarding Moco analysis using InCap data is reported since this is the focus of this thesis; however, the code remains almost unchanged for the MoCap approach. The reported script is for the subject H01, suitable for analyses of walk_5, walk_8, and walk_11 of it. The code remains unchanged for the subject H02, for the analyses of walk_7, walk_9, walk_10, walk_12, and walk_14.

Subject HS1 InCap

```
close all;
clc;
clear all;

walk = 11; % CHANGE HERE TO SWITCH WALK! For HS1 = [5 8 11]

main_path = 'C:\Users\lbot9\OneDrive - KU Leuven\Tesi_Ludovica\Moco_elaboration\H01\InCap\';

frames = [main_path, strcat('Frames\walk_', num2str(walk), '_frames.mat')];
ik = [main_path, strcat('IK\walk_', num2str(walk), '\walk_', num2str(walk), '_ik.sto')];
model = [main_path, 'HS1_scaled_calibrated.osim'];
```

MocoTrack

```
% Load the Moco libraries.
import org.opensim.modeling.*;
load(frames);

% Define the optimal control problem.
% ===================================
track = MocoTrack();
track_file = strcat('Moco results\walk_', num2str(walk), '\HS1_InCap');
track.setName([main_path, track_file]);

% Set the weights for the terms in the objective function.
% ===================================
controlEffortWeight = 10;
stateTrackingWeight = 10;

% Reference data for tracking problem.
% ===================================
tableProcessor = TableProcessor(ik);
tableProcessor.append(TabOpLowPassFilter(6));
modelProcessor = ModelProcessor(model);
modelProcessor.append(ModOpIgnoreTendonCompliance());
modelProcessor.append(ModOpReplaceMusclesWithDeGrooteFregly2016());
modelProcessor.append(ModOpIgnorePassiveFiberForcesDGF());
modelProcessor.append(ModOpScaleActiveFiberForceCurveWidthDGF(1.5));
track.setModel(modelProcessor);
track.setStatesReference(tableProcessor);
```

```
% Customized weight for MocoTrackingGoal.
% ==================================
track.set_states_global_tracking_weight(stateTrackingWeight);
w = 200;
weights = MocoWeightSet();
% pelvis
weights.cloneAndAppend(MocoWeight("/jointset/ground_pelvis/pelvis_tilt/value", w));
weights.cloneAndAppend(MocoWeight("/jointset/ground_pelvis/pelvis_list/value", w));
weights.cloneAndAppend(MocoWeight("/jointset/ground_pelvis/pelvis_rotation/value", w));
weights.cloneAndAppend(MocoWeight("/jointset/ground_pelvis/pelvis_tx/value", w));
weights.cloneAndAppend(MocoWeight("/jointset/ground_pelvis/pelvis_ty/value", w));
weights.cloneAndAppend(MocoWeight("/jointset/ground_pelvis/pelvis_tz/value", w));
% hip_r
weights.cloneAndAppend(MocoWeight("/jointset/hip_r/hip_flexion_r/value", w));
weights.cloneAndAppend(MocoWeight("/jointset/hip_r/hip_adduction_r/value", w));
weights.cloneAndAppend(MocoWeight("/jointset/hip_r/hip_rotation_r/value", w));
% hip_l
weights.cloneAndAppend(MocoWeight("/jointset/hip_l/hip_flexion_l/value", w));
weights.cloneAndAppend(MocoWeight("/jointset/hip_l/hip_adduction_l/value", w));
weights.cloneAndAppend(MocoWeight("/jointset/hip_l/hip_rotation_l/value", w));
track.set_states_weight_set(weights);

% MocoProblem settings.
% ==================================
track.set_allow_unused_references(true);
track.set_track_reference_position_derivatives(true);
track.set_apply_tracked_states_to_guess(true);
track.set_initial_time(f.f_HS_R(1));
track.set_final_time(f.f_TO_R);
study = track.initialize();
problem = study.updProblem();

% Customized weight for MocoControlGoal.
% ==================================
effort = MocoControlGoal.safeDownCast(problem.updGoal('control_effort'));
effort.setWeight(controlEffortWeight);
% glu_med
effort.setWeightForControl("/forceset/glut_med1_r", 1/0.9);
effort.setWeightForControl("/forceset/glut_med2_r", 1/0.9);
effort.setWeightForControl("/forceset/glut_med3_r", 1/0.9);
% glu_min
effort.setWeightForControl("/forceset/glut_min1_r", 1/0.9);
effort.setWeightForControl("/forceset/glut_min2_r", 1/0.9);
effort.setWeightForControl("/forceset/glut_min3_r", 1/0.9);
% hamstrings
effort.setWeightForControl("/forceset/semimem_r", 1/2);
effort.setWeightForControl("/forceset/semiten_r", 1/2);
effort.setWeightForControl("/forceset/bifemlh_r", 1/2);
effort.setWeightForControl("/forceset/bifemsh_r", 1/2);
% leg posterior group
```

```
effort.setWeightForControl("/forceset/soleus_r", 1/0.9);
effort.setWeightForControl("/forceset/med_gas_r", 1/4);
effort.setWeightForControl("/forceset/lat_gas_r", 1/7);
% thigh anterior group
effort.setWeightForControl("/forceset/rect_fem_r", 1/3);
effort.setWeightForControl("/forceset/vas_med_r", 1/3);
effort.setWeightForControl("/forceset/vas_int_r", 1/3);
effort.setWeightForControl("/forceset/vas_lat_r", 1/3);

% Bounds
% ==================================
problem.setStateInfo('/jointset/ground_pelvis/pelvis_tilt/value', ...
    [-9.5*pi/180, 9.5*pi/180]);
problem.setStateInfo('/jointset/ground_pelvis/pelvis_list/value', ...
    [-9.5*pi/180, 9.5*pi/180]);

problem.setStateInfo('/jointset/ground_pelvis/pelvis_tx/value', [-2, 2]);
problem.setStateInfo('/jointset/ground_pelvis/pelvis_ty/value', [0.75, 1.25]);
problem.setStateInfo('/jointset/ground_pelvis/pelvis_tz/value', [-1, 1]);

problem.setStateInfo('/jointset/hip_l/hip_flexion_l/value', [-30*pi/180, 60*pi/180]);
problem.setStateInfo('/jointset/hip_l/hip_adduction_l/value', [-30*pi/180, 30*pi/180]);
problem.setStateInfo('/jointset/hip_l/hip_rotation_l/value', [-30*pi/180, 30*pi/180]);

problem.setStateInfo('/jointset/hip_r/hip_flexion_r/value', [-30*pi/180, 60*pi/180]);
problem.setStateInfo('/jointset/hip_r/hip_adduction_r/value', [-30*pi/180, 30*pi/180]);
problem.setStateInfo('/jointset/hip_r/hip_rotation_r/value', [-30*pi/180, 30*pi/180]);

problem.setStateInfo('/jointset/knee_l/knee_flexion_l/value', [-60*pi/180, 9.5*pi/180]);
problem.setStateInfo('/jointset/knee_l/knee_adduction_l/value', [-15*pi/180, 10*pi/180]);
problem.setStateInfo('/jointset/knee_l/knee_rotation_l/value', [-5*pi/180, 20*pi/180]);

problem.setStateInfo('/jointset/knee_r/knee_flexion_r/value', [-60*pi/180, 9.5*pi/180]);
problem.setStateInfo('/jointset/knee_r/knee_adduction_r/value', [-15*pi/180, 10*pi/180]);
problem.setStateInfo('/jointset/knee_r/knee_rotation_r/value', [-5*pi/180, 20*pi/180]);

problem.setStateInfo('/jointset/ankle_l/ankle_angle_l/value', [-20*pi/180, 29*pi/180]);
problem.setStateInfo('/jointset/ankle_r/ankle_angle_r/value', [-20*pi/180, 29*pi/180]);
problem.setStateInfo('/jointset/subtalar_r/subtalar_angle_r/value', ...
    [-20*pi/180, 29*pi/180]);
problem.setStateInfo('/jointset/subtalar_l/subtalar_angle_l/value', ...
    [-20*pi/180, 29*pi/180]);

problem.setStateInfo('/jointset/back/lumbar_extension/value', [-20*pi/180, 20*pi/180]);
problem.setStateInfo('/jointset/back/lumbar_bending/value', [-20*pi/180, 20*pi/180]);
problem.setStateInfo('/jointset/back/lumbar_rotation/value', [-20*pi/180, 20*pi/180]);

% To modify solver settings uncomment
% solver = study.initCasADiSolver();
% To print the Moco setup uncomment
% study.print('Moco_setup.xml');

% Solve the problem
% ==================================
Moco_solution_MoCap = study.solve();
solution_file = strcat('Moco results\walk_', num2str(walk), '\Moco_solution_InCap_', ...
    num2str(walk), '.sto');
Moco_solution_MoCap.write([main_path, solution_file]);
```

GRF extraction

```
import org.opensim.modeling.*;

% Create MocoStates file
% ==================================
[data, names, table] = readMOTSTOTRCfiles_v1([main_path, ...
    strcat('Moco results\walk_', num2str(walk), '\')], ...
    strcat('Moco_solution_InCap_', num2str(walk), '.sto'));

C2 = [repmat({''}, 120,1) ;repmat({'/fiber_length'},78,1)];
names_new = strcat(names', C2);
names_new(121:145,:) = [];
names_new(end-3:end, :) = [];
names_new = names_new';
data_new = data;
data_new(:, 121:145) = [];
data_new(:, end-3:end) = [];
states_file = strcat('Moco results\walk_', num2str(walk), '\Moco_states_InCap_', ...
    num2str(walk), '.sto');
generateMotFile_v2(data_new, names_new, [main_path, states_file]);

model = Model(model);
model.initSystem();
statesTraj = StatesTrajectory();
states = statesTraj.createFromStatesStorage(model, [main_path, states_file]);

% Extract ground reaction forces
% ==============================
contact_r = StdVectorString();
contact_l = StdVectorString();
contact_r.add('SmoothSphereHalfSpaceForce_s1_r');
contact_r.add('SmoothSphereHalfSpaceForce_s2_r');
contact_r.add('SmoothSphereHalfSpaceForce_s3_r');
contact_r.add('SmoothSphereHalfSpaceForce_s4_r');
contact_r.add('SmoothSphereHalfSpaceForce_s5_r');
model = modelProcessor.process();
```

```
externalForcesTableFlat = opensimMoco.createExternalLoadsTableForGait(model, ...
    states, contact_r, contact_l);
GRF_file = strcat('Moco results\walk_', num2str(walk), '\Moco_solutionGRF_InCap_', ...
    num2str(walk), '.mot');
STOFileAdapter.write(externalForcesTableFlat, [main_path, GRF_file]);
```

CoP evaluation with ZMP method

```
clear data; clear data_new;
clear names; clear names_new;
clear table; clear C2;

% Load Moco estimated forces and calcn pointKinematics file
% ==============================
calcn_origin = 'HS1_PointKinematics_calcn_origin_pos.sto';
GRF = strcat('Moco_solutionGRF_InCap_', num2str(walk), '.mot');

[data, names, table] = readMOTSTOTRCfiles_v1([main_path, strcat('Moco results\walk_', ...
    num2str(walk), '\')], GRF);
[data1, names1, table1] = readMOTSTOTRCfiles_v1([main_path, strcat('Moco results\walk_', ...
    num2str(walk), '\calcn_origin\')], calcn_origin);

data_norm = TimeNormalize_v1(data, 101);
data1_norm = TimeNormalize_v1(data1, 101);

% ZMP method to evaluate px, pz and My
% ==============================
data_norm(:, 5) = (data_norm(:, 16)./data_norm(:,3)) - ...
    (data1_norm(:, 3).*(data_norm(:, 2)./data_norm(:,3))) + ...
    data1_norm(:, 2); %px
data_norm(:, 7) = (data_norm(:, 14)./data_norm(:, 3)) - ...
    (data1_norm(:, 3).*(data_norm(:, 4)./data_norm(:, 3))) + ...
    data1_norm(:, 4); %pz
data_norm(:, 15) = data_norm(:, 15) + ...
    ((data_norm(:, 4).*data_norm(:, 16))./data_norm(:, 3)) + ...
    ((data_norm(:, 2).*data_norm(:, 14))./data_norm(:, 3)); %My
data_norm(:, 14) = 0; %Mx
data_norm(:, 16) = 0; %Mz

data_norm(isinf(data_norm)|isnan(data_norm)) = 0;

% Filter data
% ==============================
fs=100;
[b, a] = butter(2, 20/fs*2, 'low');
data_filt = filtfilt(b, a, data_norm);
```

```
% Generate the complete estimated GRF file
% ==============================
generateMotFile_v1(data_filt, names, [main_path, strcat('Moco results\walk_', ...
    num2str(walk), '\estimated_GRF_InCap_', num2str(walk), '.mot')]);
```

Appendix B: MocoTrack MoCap results

1. Kinematics: reference vs. MocoTrack MoCap

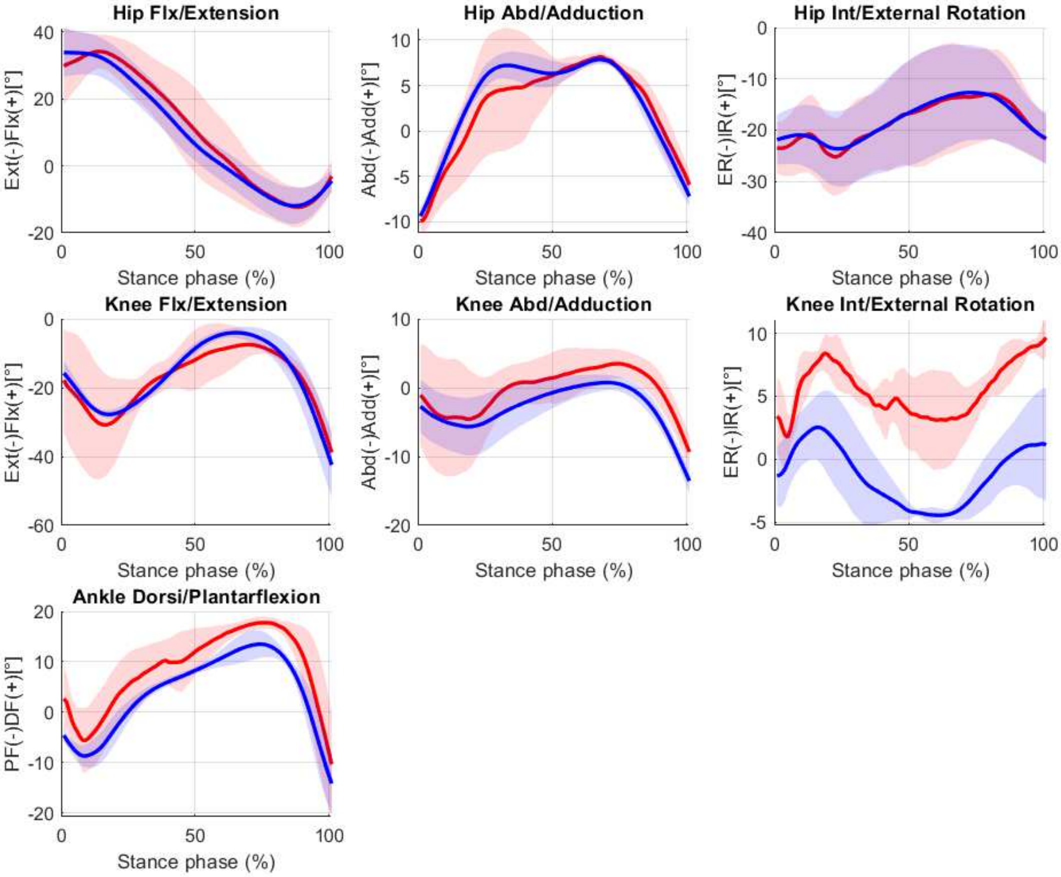


Figure 43: Kinematics comparison: in red are the MoCap data, while in blue are the tracked Moco results obtained with MoCap data as input. The continue lines represent the mean (± std) across the 2 subjects both for InCap and Moco results.

	mean ± std		
	Correlation coefficient	**RMSE [°]**	**Maximum difference [°]**
Hip FE	0.95 ± 0.068	4.91 ± 4.73	17.47 ± 20.15
Hip AA	0.86 ± 0.170	2.65 ± 2.62	9.84 ± 10.15
Hip IE	0.98 ± 0.001	1.33 ± 0.37	4.76 ± 0.83
Knee FE	0.84 ± 0.184	8.15 ± 3.03	26.79 ± 18.52
Knee AA	0.78 ± 0.188	4.01 ± 0.61	10.73 ± 5.23
Knee IE	0.66 ± 0.386	6.64 ± 3.73	9.78 ± 5.15
Ankle DF	0.92 ± 0.072	5.32 ± 3.85	13.59 ± 5.32

Table 9: Average value (± std) of the correlation coefficient (Pearson value), root mean square error (RMSE), and maximum of the difference between each joint angle of the reference (MoCap approach) and MocoTrack results. It is the average between subjects (**inter-subjects**), each one characterized by the average of their walking trials (**intra-subjects**).

2. Ground reaction forces: measured vs estimated MoCap

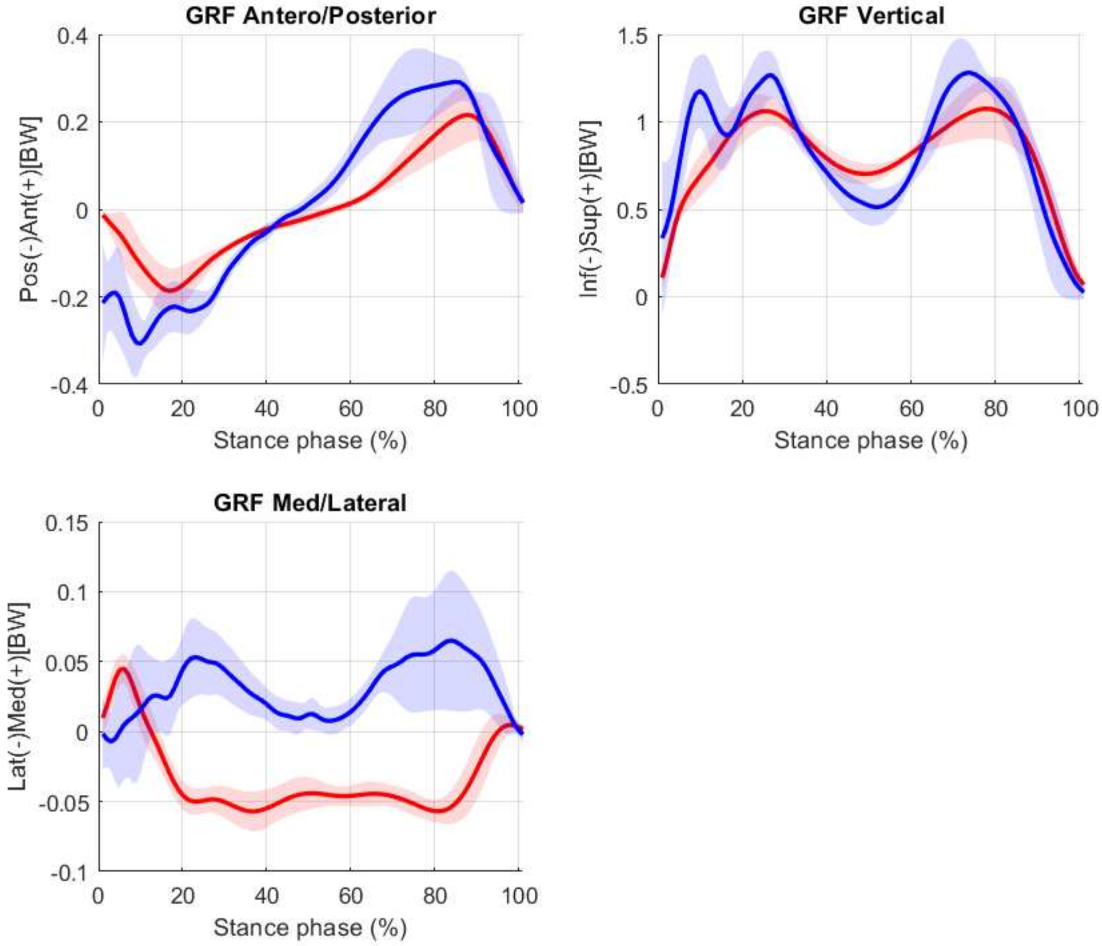


Figure 44: GRFs comparison: in red are the MoCap measured values, while in blue are the Moco estimated results obtained with MoCap data as input. The continue lines represent the mean (± std) across the 2 subjects for both measured and estimated results.

	mean ± std		
	Correlation coefficient	**RMSE [BW]**	**Maximum difference [BW]**
AP Force	0.89 ± 0.015	0.11 ± 0.014	0.34 ± 0.084
V Force	0.74 ± 0.077	0.25 ± 0.015	1.05 ± 0.234
ML Force	-0.42 ± 0.06	0.07 ± 0.031	0.15 ± 0.042

Table 10: Average value (± std) of the correlation coefficient (Pearson value), root mean square error (RMSE), and maximum differences between the measured GRFs by the MoCap system (particularly, by the force plates) and the estimated one by using the contact geometry model and motion capture data. It is the average between subjects (**inter-subjects**), each one characterized by the average of their walking trials (**intra-subjects**).

3. Center of pressure: measured vs. estimated MoCap

	mean ± std	
	Correlation coefficient	**RMSE [m]**
CoP AP Force	0.66 ± 0.45	0.029 ± 0.030
CoP ML Force	0.34 ± 0.89	0.019 ± 0.020

Table 11: Average value (± std) of the correlation coefficient (Pearson value), and root mean square error (RMSE) between the measured center of pressure by the MoCap system (particularly, by the force plates) and the ZMP evaluated by using MoCap estimated results over all subjects. It is the average between subjects (**inter-subjects**), each one characterized by the average of their walking trials (**intra-subjects**).

4. Dynamics: reference vs. MocoTrack MoCap

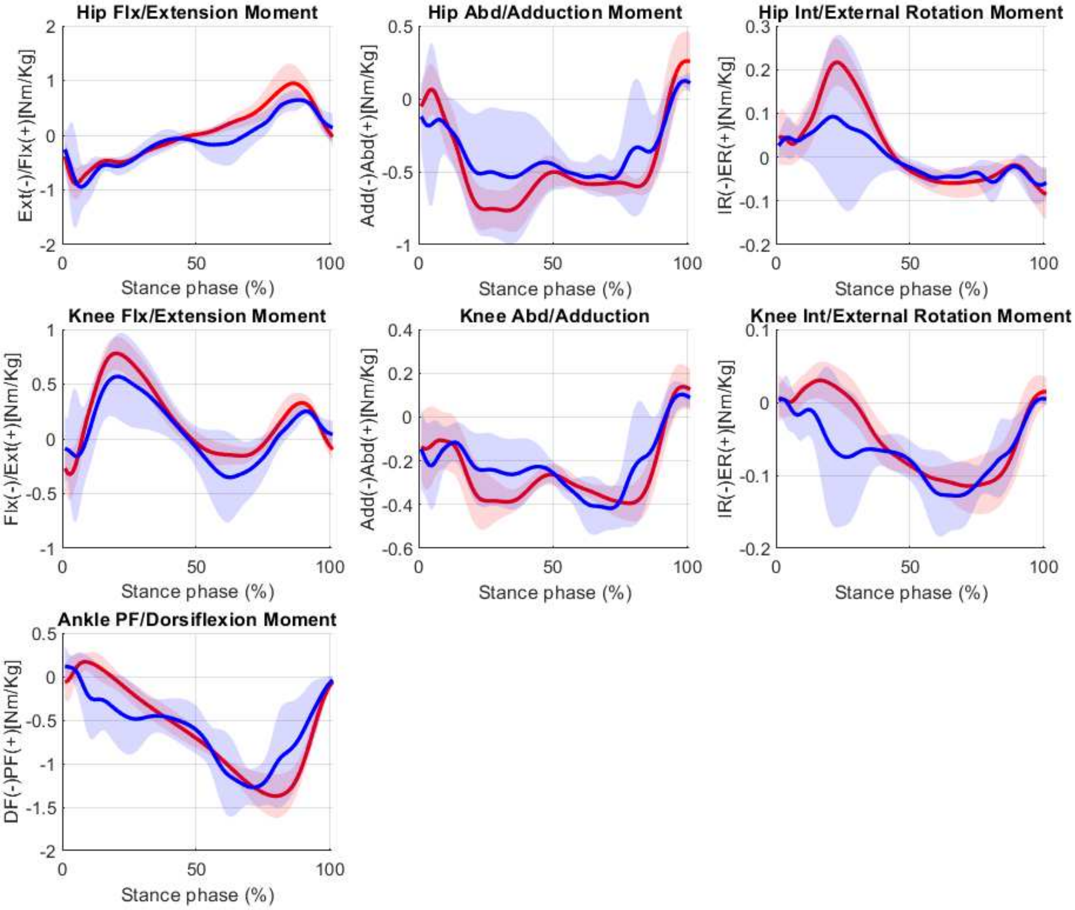


Figure 45: Dynamics comparison: in red are the MoCap system results, while in blue are the optical-based ones using estimated GRFs. The continue lines represent the mean (± std) across the 2 subjects both for MoCap and Mocap + estimated GRFs results.

	mean ± std	
	Correlation coefficient	**RMSE [Nm/kg]**
Hip FE	0.76 ± 0.330	29.59 ± 7.05
Hip AA	0.60 ± 0.564	26.44 ± 8.55
Hip IE	-0.04 ± 0.002	5.04 ± 0.31
Knee FE	0.75 ± 0.340	22.85 ± 4.19
Knee AA	0.65 ± 0.490	16.31 ± 0.58
Knee IE	0.58 ± 0.581	5.28 ± 1.94
Ankle DF	0.65 ± 0.493	44.92 ± 3.85

Table 12: Average value (± std) of the correlation coefficient (Pearson value), and root mean square error (RMSE) between each joint moment of the reference (MoCap approach) and MoCap+ estimated GRFs result. It is the average between subjects (**inter-subjects**), each one characterized by the average of their walking trials (**intra-subjects**).

Bibliography

Papers

[1] A. Ancillao et al. “Indirect Measurement of Ground Reaction Forces and Moments by Means of Wearable Inertial Sensors: A Systematic Review”. In: *Sensors* 18 (2018), p. 2564.

[2] Z. Sawacha. “Lectures Notes of Bioengineering of Movement and Rehabilitation course”. In: *University of Padua* (2020).

[4] C. L. Dembia et al. “OpenSim Moco: Musculoskeletal optimal control”. In: *PLOS Computational Biology* 16 (2020). Ed. by K. S. Campbell, e1008493.

[5] J.K. Aggarwal and Q. Cai. “Human Motion Analysis: A Review”. In: *Computer Vision and Image Understanding* 73.3 (1999), pp. 428–440.

[8] A. D. Grant. “Gait Analysis: Normal and Pathological Function”. In: *JAMA* 304.8 (Aug. 2010), pp. 907–907.

[11] C. L. Vaughan. “Theories of bipedal walking: an odyssey”. In: *Journal of Biomechanics* 36.4 (2003), pp. 513–523.

[13] M. Q. Liu et al. “Muscle contributions to support and progression over a range of walking speeds”. In: *Journal of Biomechanics* 41 (2008), pp. 3243–3252.

[14] S. Joshi and A. Gupta. “Conceptual Design of an Active Transtibial Prosthesis Based on Expected Joint and Muscle Forces in a Unilateral Transtibial Amputee: A Modelling Study”. In: *Volume 3: Biomedical and Biotechnology Engineering* (2015). (Visited on 03/05/2023).

[15] A. Cappozzo et al. “Human movement analysis using stereophotogrammetry”. In: *Gait & Posture* 21 (2005), pp. 186–196.

[17] C. A. Fukuchi, R. K. Fukuchi, and M. Duarte. “A public dataset of overground and treadmill walking kinematics and kinetics in healthy individuals”. In: *PeerJ* 6 (2018), e4640. (Visited on 03/10/2020).

[18] P. Picerno. “25 years of lower limb joint kinematics by using inertial and magnetic sensors: A review of methodological approaches”. In: *Gait & Posture* 51 (2017), pp. 239–246.

[24] M. M. van der Krogt, L. H. Sloot, and J. Harlaar. “Overground versus self-paced treadmill walking in a virtual environment in children with cerebral palsy”. In: *Gait & Posture* 40 (2014), pp. 587–593.

[25] A. H. Abdul Razak et al. “Foot Plantar Pressure Measurement System: A Review”. In: *Sensors* 12 (July 2012), pp. 9884–9912.

[26] E. Shahabpoor and A. Pavic. “Measurement of Walking Ground Reactions in Real-Life Environments: A Systematic Review of Techniques and Technologies”. In: *Sensors (Basel, Switzerland)* 17 (2017).

[27] Y. Ohtaki, K. Sagawa, and H. Inooka. “A Method for Gait Analysis in a Daily Living Environment by Body-Mounted Instruments”. In: *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing* 44 (2001), pp. 1125–1132.

[28] J. M. Neugebauer, D. A. Hawkins, and L. Beckett. “Estimating Youth Locomotion Ground Reaction Forces Using an Accelerometer-Based Activity Monitor”. In: *PLoS ONE* 7 (2012), e48182.

[29] E. Yang and M. Mao. “3D analysis system for estimating intersegmental forces and moments exerted on human lower limbs during walking motion”. In: *Measurement* 73 (2015), pp. 171–179.

[30] A. Karatsidis et al. “Estimation of Ground Reaction Forces and Moments During Gait Using Only Inertial Motion Capture”. In: *Sensors* 17 (2016), p. 75.

[31] Y. Lin and M. G. Pandy. “Three-dimensional data-tracking dynamic optimization simulations of human locomotion generated by direct collocation”. In: *Journal of Biomechanics* 59 (2017), pp. 1–8.

[33] M. Kelly. “An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation”. In: *SIAM Review* 59 (2017), pp. 849–904.

[36] J. A. E. Andersson et al. “CasADi: a software framework for nonlinear optimization and optimal control”. In: *Mathematical Programming Computation* 11 (2018), pp. 1–36.

[37] A. Wächter and L. T. Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical Programming* 106 (2005), pp. 25–57.

[38] P. E. Gill, W. Murray, and M. A. Saunders. “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization”. In: *SIAM Review* 47 (2005), pp. 99–131.

[40] A. Falisse, M. Afschrift, and F. De Groote. “Modeling toes contributes to realistic stance knee mechanics in three-dimensional predictive simulations of walking”. In: *PLOS ONE* 17 (2022). Ed. by G. S. Cymbalyuk, e0256311.

[41] S. R. Hamner, A. Seth, and S. L. Delp. “Muscle contributions to propulsion and support during running”. In: *Journal of Biomechanics* 43 (2010), pp. 2709–2716.

[42] C. C. Raasch et al. “Muscle coordination of maximum-speed pedaling”. In: *Journal of Biomechanics* 30 (1997), pp. 595–602.

[43] F. De Groote et al. “Evaluation of Direct Collocation Optimal Control Problem Formulations for Solving the Muscle Redundancy Problem”. In: *Annals of Biomedical Engineering* 44 (2016), pp. 2922–2936.

[46] D. Gupta, Cyril J. Donnelly, and Jeffrey A. Reinbolt. “Physics-Based Guidelines for Accepting Reasonable Dynamic Simulations of Movement”. In: *IEEE Transactions on Biomedical Engineering* 69 (2022), pp. 1194–1201.

Books

[9] J. B. Webster and B. J. Darter. *Principles of Normal and Pathologic Gait.* 2019.

[10] D. C. Rizzo. *Fundamentals of Anatomy and Physiology.* 2015.

[12] J. M.D. Perry. *Gait Analysis: Normal and Pathological Function.* 1992.

[16] J. Richards. *The comprehensive textbook of clinical biomechanics.* 2018.

[21] V. M. Zatsiorsky. *Biomechanics of the Human Musculoskeletal System.* Fizkultura i Sport, 1981.

[23] A. Ancillao. *Modern Functional Evaluation Methods for Muscle Strength and Gait Analysis.* 2018.

[34] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I.* Springer Science & Business Media, 2008.

Websites

[3] *OpenSim Moco*. opensim-org.github.io. URL: https://opensim-org.github.io/opensim-moco-site/.

[6] *Hip Joint - an overview | ScienceDirect Topics*. www.sciencedirect.com. URL: https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/hip-joint.

[7] *Joint Kinematics - an overview | ScienceDirect Topics*. www.sciencedirect.com. URL: https://www.sciencedirect.com/topics/materials-science/joint-kinematics.

[19] *(PDF) Human Gait and Clinical Movement Analysis*. ResearchGate. URL: https://www.researchgate.net/publication/301935875_Human_Gait_and_Clinical_Movement_Analysis.

[20] *The Pros, Cons and Uses of Different Gyroscopes*. CEVA's Experts blog, 2018. URL: https://www.ceva-dsp.com/ourblog/exploring-the-application-of-gyroscopes/.

[22] *OpenSim Documentation - Global Site*. Stanford.edu, 2017. URL: https://simtk-confluence.stanford.edu:8443/display/OpenSim/OpenSim+Documentation.

[32] *Shooting method*. Wikipedia, 2019. URL: https://en.wikipedia.org/wiki/Shooting_method.

[35] *Webinar - OpenSim Moco: Software to optimize the motion and control of OpenSim models*. www.youtube.com. URL: https://www.youtube.com/watch?v=IYYZgyE33pU&t=790s.

[39] *Xsens Products*. URL: https://www.xsens.com/.

[44] *Vicon Documentation*. docs.vicon.com. URL: https://docs.vicon.com/dashboard.action.

[45] *Skeleton Tracking - NaturalPoint Product Documentation Ver 2.2*. v22.wiki.optitrack.com. URL: `https://v22.wiki.optitrack.com/index.php?title=Skeleton_Tracking`.

List of Figures

List of Tables