UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITY OF PADUA

DEPARTMENT OF INFORMATION ENGINEERING

MASTER'S DEGREE IN BIOENGINEERING

"Estimation of ground reaction forces with applications for ecological monitoring of joint loading: a combined musculoskeletal and optimization-based proof of concept"

Supervisor:

Prof. Zimi Sawacha, PhD

Candidate:

Cristina Imbesi

2036976

Co-Supervisors:

Prof. Ilse Jonkers

Dr. Bryce A Killen

Dr. Giacomo Di Raimondo

Academic Year 2022 – 2023

October 19th, 2023

INDEX

Abstract

This thesis has been carried on during Erasmus Studio+ project resulting from the collaboration between the research group of the Laboratory of Human Movement Bioengineering at the University of Padua (Italy) and the research group of Human Movement Biomechanics of the Faculty of Movement and Rehabilitation Sciences at the Katholieke Universiteit in Leuven (Belgium).

The goal of this project is to create a new approach for the assessment of ground reaction forces, one of the fundamental parameters of gait analysis. At the clinical level, human motion analysis, can provide useful information for a variety of applications, from identifying functional abnormalities associated with pathological gait patterns, to rehabilitation, for example, of the knee after injury. Three-dimensional motion capture (3D MoCap) is the a widely accepted technique as the state of the art for the study of human motion: it consists of a series of precise measurements, which are processed by biomechanical models to produce curves related to indirect kinematics and dynamics, joint angles, forces and relative moments. The gold standard for measuring ground reaction forces (GRFs) is based on force platforms (FPs) often integrated with these MoCap systems in motion analysis laboratories.

Generally, these results are considered reliable and used to make clinical decisions on specific interventions to restore a more functional gait for pathological individuals, from rehabilitation treatments to surgery. 3D MoCap requires a controlled environment and highly skilled operators; this involves a significant economic expense, so the goal is to look for cheaper surrogate techniques. An additional issue to consider is that when subjects walk in a controlled environment (e.g., a treadmill) they may not have a spontaneous gait, and this may alter the results of the analysis performed. As an alternative to using a laboratory environment, several methods have been developed to analyze movement, such as using a simple smartphone in conjunction with OpenCap, an open-source software package developed by Stanford University.

In this study, particular attention is given to the use of mobile devices, as an alternative to FP, to measure GRFs: for example, methods based on pressure insoles that directly measure GRFs, or methods based on wearable inertial sensors (IMUs) that measure the movement of individual body parts, approximated to segments, and estimate GRFs using musculoskeletal models, and finally methods that exploit machine leaning [1]. Musculoskeletal modeling and simulation

provide an ideal framework for studying in silico quantities such as musculoskeletal load, muscle forces, and joint contact forces that cannot be measured in vivo [2].

OpenSim, an open-source software that enables modeling, analysis, and simulation of the musculoskeletal system, was used to carry out this project, along with MATLAB, a programming and numerical computing platform. The aim of this thesis is to optimize the GRFs derived from the contact geometry belonging to a musculoskeletal model so that they are as close as possible to the values obtained from the force platforms in the laboratory.

Sommario

Questa tesi è stata sviluppata durante il progetto Erasmus Studio+ come risultato della collaborazione tra il gruppo di ricerca del Laboratorio di Bioingegneria del Movimento Umano dell'Università di Padova (Italia) e il gruppo di ricerca di Biomeccanica del Movimento Umano della facoltà di Scienze del Movimento e della Riabilitazione della Katholieke Universiteit di Leuven (Belgio).

L'obiettivo del suddetto progetto è quello di creare un nuovo approccio per la valutazione delle forze di reazione al suolo, uno dei parametri fondamentali dell'analisi del cammino. A livello clinico, l'analisi del movimento umano, può fornire informazioni utili per diverse applicazioni, dall'identificazione delle anomalie funzionali associate a modelli di andatura patologici, alla riabilitazione, per esempio, del ginocchio dopo un infortunio. La cattura tridimensionale del movimento (3D MoCap) è la tecnica ampiamente accettata come lo stato dell'arte per lo studio del movimento umano: consiste in una serie di misurazioni precise, che vengono elaborate da modelli biomeccanici per produrre curve relative alla cinematica e alla dinamica indiretta, di angoli articolari, forze e momenti relativi. Il gold standard per misurare le forze di reazione al suolo (GRF) si basa su piattaforme di forza (FP) spesso integrate con questi sistemi MoCap nei laboratori di analisi del movimento.

Questi risultati sono generalmente considerati affidabili e utilizzati per prendere decisioni cliniche su interventi specifici, per ripristinare un'andatura più funzionale degli individui patologici, dai trattamenti di riabilitazione alla chirurgia. La 3D MoCap richiede un ambiente controllato e operatori altamente qualificati, ciò comporta un notevole dispendio economico, pertanto l'obiettivo è di cercare tecniche sostiutive più economiche. Un ulteriore aspetto da valutare è quello relativo al fatto che i soggetti quando camminano in un ambiente controllato (ad esempio, un tapis roulant) possono non avere un andamento spontaneo e ciò può alterare i risultati dell'analisi effettuata. In alternativa all'uso di un ambiente di laboratorio, sono stati sviluppati diversi metodi per analizzare il movimento, ad esempio con l'uso di un semplice smartphone insieme a OpenCap, un pacchetto software open-source sviluppato dall'Università di Stanford.

In questo studio viene rivolta particolare attenzione all'uso di dispositivi mobili, come alternativa alle FP, per misurare le GRF: ad esempio, metodi basati su solette di pressione che misurano direttamente le GRF, o metodi basati su sensori inerziali indossabili (IMU) che misurano il movimento di singole parti del corpo, approssimate a segmenti, e stimano le GRF utilizzando modelli muscoloscheletrici, e infine metodi che sfruttano il machine leaning [1]. La

modellazione e la simulazione muscoloscheletrica forniscono un quadro ideale per studiare in silico grandezze come il carico muscoloscheletrico, le forze muscolari e le forze di contatto articolare che non possono essere misurate in vivo [2].

Per lo svolgimento di questo progetto è stato utilizzato OpenSim, un software open-source che consente la modellazione, l'analisi e la simulazione del sistema muscoloscheletrico, insieme a MATLAB, una piattaforma di programmazione e calcolo numerico. L'obiettivo di questa tesi è l'ottimizzazione delle GRF ricavate dalla geometria di contatto appartenenti ad un modello muscoloscheletrico, in modo tale che siano il più possibile prossime ai valori ottenuti dalle piattaforme di forza in laboratorio.

1. HUMAN MOTION ANALYSIS

1.1. Introduction

Motion is a variation of position during time. Human motion analysis refers to the study and description of the human body during motion. The aim of motion analysis is to give quantitative and qualitative information about musculoskeletal system mechanics during movement, how it is controlled, and how it responds to different stimuli. This involves the use of techniques from various fields such as biomechanics, computer vision, and machine learning, to analyze, model, and understand human motion [3]. This analysis can be used in several 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:

- Obtaining the motion that could be done on selected point, body parts, joints, or the entire human body;
- Deriving muscular activity;
- Knowing the loads acting on tissues during specific motor tasks.

Moreover, with this approach it is possible to design therapeutic routes, to assess functional limitations, to track the effectiveness of rehabilitation therapy, to diagnose eventual motor disorders, and to document the usage of aids and prosthesis, in the clinical field. In the same way, this methodical approach applies to evaluate the performances of an athlete, to optimize them and to prevent eventual injuries.

In order to better understand 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 illustrated in [Fig. 1]:

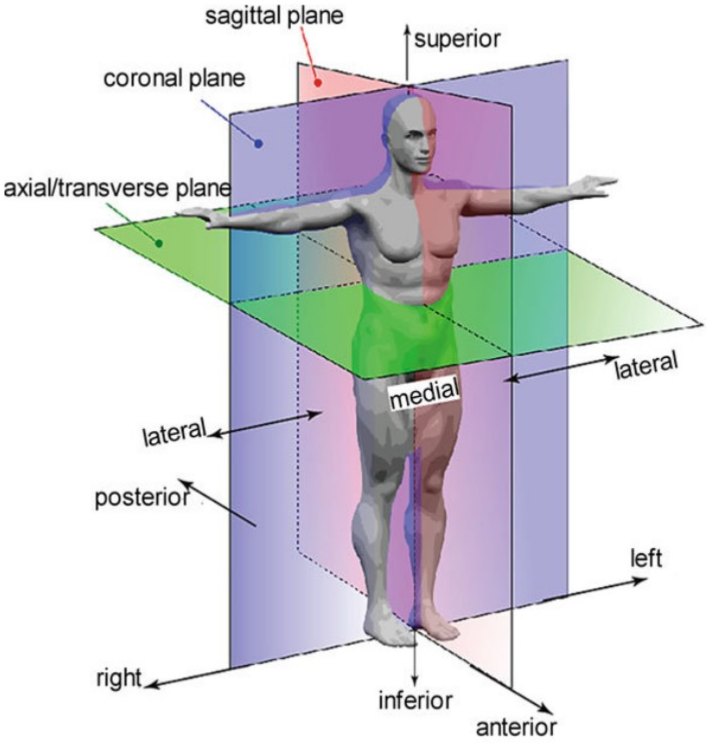


Figure 1: Anatomical planes and position definition. [4]

As it can be seen from the [Fig.1], the main planes are [5]:

- Sagittal plane (X, Y), which divides the human body into two symmetrical halves (right and left) and median plane refers to the midline while paramedian plane refers to subdividing one half;
- Frontal or coronal plane (Y, Z), which divides the body into two asymmetrical halves (anterior and posterior);
- Transverse plane (X, Z), which divides the body into two asymmetrical halves (upper and lower).

While the main axes are:

- X, the sagittal, or antero-posterior axis, which passes horizontally from anterior to posterior and it is formed by the intersection of the sagittal and transverse planes;
- Y, the vertical axis, which passes vertically from superior to inferior and it is formed by the intersection of the sagittal and coronal planes;
- Z, the coronal, or medio-lateral axis, which passes horizontally from left to right and it is formed by the intersection of the coronal and transverse planes;

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: Flexion refers to decreasing a joint angle, and extension to increasing the joint angle back to resting anatomical position;

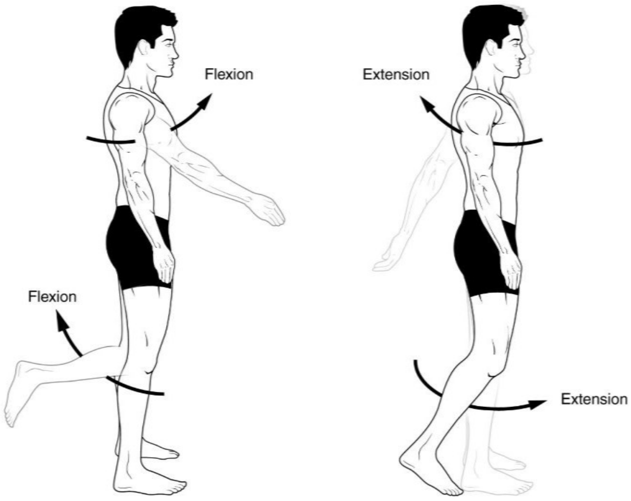


Figure 2: Flexion and extension movements of the shoulder and the knee.

- Adduction/abduction, around the X axis in the frontal plane: Abduction is moving a body part away from its resting anatomical position in the coronal plane; adduction is returning it to its normal resting position;

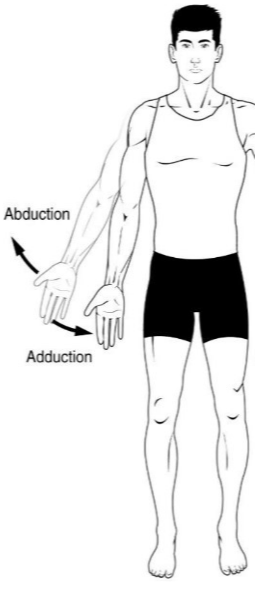


Figure 3: Abduction and adduction movements of the shoulder.

- Internal/external (or medio/lateral) rotation, around the Y axis in the transverse plane: Internal rotation is rotating a joint towards the midline and external rotation is rotating a joint away from the midline.

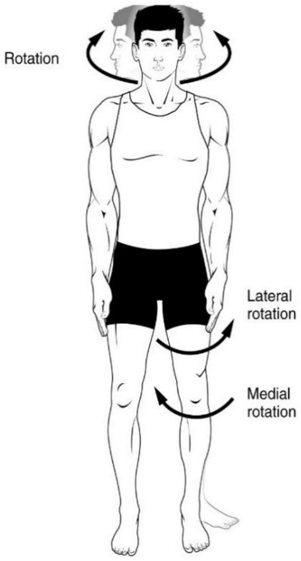


Figure 4: Internal (medial) rotation and external (lateral) rotation of the hip.

Other movements are possible and result from the combination of the latter.

1.2. Gait analysis

Human gait is a periodic motion of body segments. The analysis of this periodic motion and related studies is termed gait analysis, that is focused on 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 [6]. 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 [7]. 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 [8]. 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 [8]. Gait analysis has gained much popularity because of its applications in clinical diagnosis (identifying abnormalities or deviations from normal gait that could mean a MSK disorder, or an injury), rehabilitation methods, gait biometrics, robotics, sports, and biomechanics. Traditionally, subjective assessment of gait was conducted by health experts; however, with the advancement in technology, gait analysis can now be performed objectively and empirically for better and more reliable assessment.

Determination of gait functionalities, irregularities, and classifications are the main target of gait analysis so that better treatment is possible to provide for no ambulatory patients and also to provide gait enhancement facilities for military or disaster recovery applications. Because of the complex structure of the human musculoskeletal system, gait analysis becomes the preliminary requirement in understanding the complex dynamics of its locomotion strategy. Several techniques are used in gait analysis such as optical system, inertial system, force plate mechanism, force shoes, foot plantar pressure sensor, electromyography, goniometric measurement system, magnetic system, acoustic tracking system, medical imaging technique, and uses of portable media devices such as smartphones, and some of them are going to be mentioned in this chapter.

1.2.1. Human gait cycle

The characteristics of human walking can be defined as the alternation of the same type of elementary movements of legs and feet [Fig.5].

The movements of the legs for a single step can be categorized into two phases: Double Support Phase (DSP) and Single Support Phase (SSP). DSP starts at the moment of heel-contact or heel-strike of one foot on the ground and ends at the toe-off situation of the rear foot (0 – 10% of gait cycle). The end of DSP is the start of SSP (10 – 50% of gait cycle) or swing phase. The conclusion of the swing phase occurs at the beginning of DSP i.e., heel-strike. So, a cycle of a single step begins with the heel-strike (0% of the gait cycle) and ends up with the following heel-strike (100% of the gait cycle) of the same foot [9].

The gait cycle of a single leg also can be characterized by two other phases: Stance Phase and Swing Phase. The Swing Phase and the SSP represent the same manner of movement pattern and occur during 40% (approximately) of a gait cycle. The rest 60% (approximately) of a gait cycle represents the Stance Phase of the same foot. During the Stance Phase, DSP occurs at the beginning and the end of the phase. Stance phase is one of the major parts of a gait cycle because during this time, the body is balanced on the single leg, contact limb supports total weight, body propelled forward, and also maintains the direction of swing leg movement [10].

So, the stance phase is the period in which the foot is in contact with the ground, and it could be divided into several sub-phases [7]:

1. Heel Strike (HS) – Initial contact (0% of gait cycle): This is the moment when the heel of the leading foot makes initial contact (from here the name) with the ground. The rest of the foot is off the ground.

2. Foot flat (FF) – Loading response (0 – 10% of gait cycle): After the heel strike, comes the moment in which the foot becomes flat on the ground, so all the foot touches the ground.

3. Midstance (10 – 30% of gait cycle): During this moment, the body's weight is positioned over the supporting leg, maintaining stability. The opposite leg swings forward in preparation for the next step.

4. Heel off – Terminal stance (30 – 50% of gait cycle): At this moment, since the body moves forward, the heel starts to lift off the ground. The toes remain in contact with the ground, and the body weight is transferred to the forefoot.

5. Toe off – Pre-Swing (50 – 60% of gait cycle): After the heel is off the ground, the toes follow it, propelling the body forward. The foot leaves the ground and the stance phase ends.

While the swing phase, is the period in which the foot is in air, and it could be divided into these different sub-phases:

1. Initial swing – Acceleration (60 – 73% of gait cycle): During this moment, the leg swings forward and the foot gradually increases its speed.

2. Mid – Swing (73 – 87% of gait cycle): In this moment, the leg continues to swing forward, and the foot reaches the highest part in the swing arc. The knee starts to extend in preparation for the next stance phase.

3. Terminal Swing – Deceleration (87 – 100% of gait cycle): This is the moment in which the swing phase ends, in fact the foot begins to decelerate because it is reaching the ground preparing for the new heel strike.

During ambulation, the main active muscles are [11]:

- Gastrocnemius and soleus: these muscles are located in the calf and work together to plantarflex the ankle joint during the toe-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;

- 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; also active during the toe off phase but they contract to extend the knee joint;

- 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 (is active together with the gastrocnemius) 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;

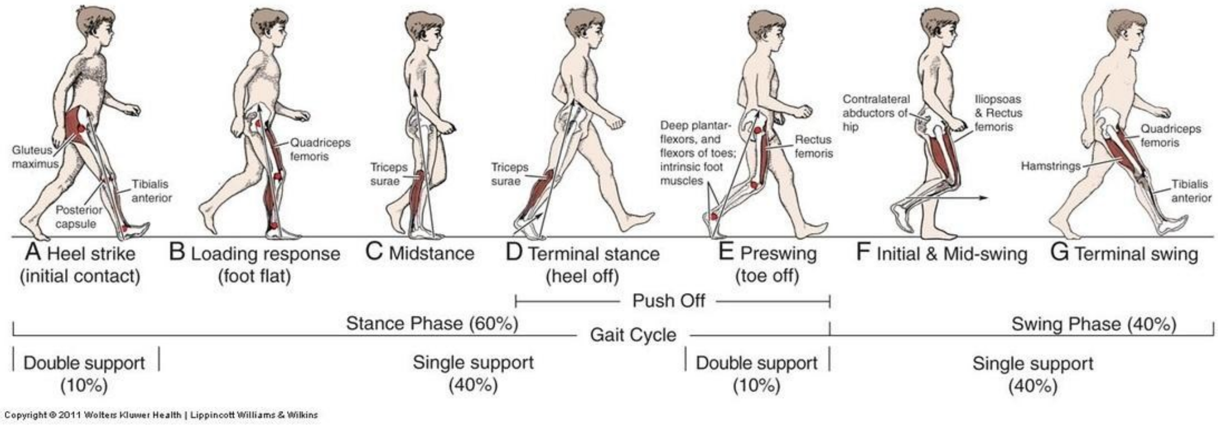
- Gluteus Medius and gluteus minimus, that form the abductor muscles, are active during the stance phase to maintain stability. The Gluteus Medius muscle is located in the side of the hip and plays an important role in stabilizing the pelvis during single-leg stance. As the weight of the body shifts over the stance leg, this muscle 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;

- Hamstrings: they 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, so controlling the leg, 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;

- Iliopsoas: It belongs to the hip flexors, and it is active during the swing phase assisting the lifting of the leg and moving it forward;

- Adductors, that help with the control of the leg movements during the stance phase;

- 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 helping control the lowering of the foot, and peroneals. 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.



Copyright © 2011 Wolters Kluwer Health | Lippincott Williams & Wilkins

Figure 5: Gait analysis phases and muscles involved.

Gait analysis involves the evaluation of an individual's gait pattern, such as some space-time parameters that are calculated 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 must be identifiable [2]. The temporal parameters are :

- Cadence [steps/min]: this is the number of steps per minute;

- Stance and swing phases duration [s]: they are the times between heel strike and toe off and between toe off and heel strike of the same foot respectively;

- Single and double support duration [s]: they are the times in mono and bi podalic support respectively;

- Step duration [s]: this is the time between ipsilateral and contralateral heel contact;

- Stride duration [s]: this is the temporal interval between consecutive heel strikes of the same foot;

While the spatial parameters are:

- Step length [m]: this is the distance between consecutive heel strikes of the same foot;
- Advancement length [m]: this is the distance covered during a complete gait cycle of one leg;
- Step speed [m/s]: this is the linear velocity along walking direction measured on one step;
- Walking speed [m/s]: this is the linear velocity along walking direction measured on one or more steps.

1.3. Motion capture instrumentation

One of the established techniques of gait analysis is the use of motion capture system [2], abbreviated as mocap. Although a number of researchers are trying to establish various strategies in representation, characterization, and recognition of human gait, it is really a challenging job to extract targeted and suitable features from image sequence. Targeted features could be joint positions, joint motion trajectories, and joint angle variations during walking. Gait motion can be analyzed with or without having any marker attached to the human body.

In 3D mocap with markers, the subject is fitted with markers and/or IMU sensors, which can be attached directly to specific anatomical landmarks on the body (including pelvis, thighs, shanks, feet, and upper body, or placed on a specialized suit, and they help to track the subject's movements accurately. These markers can be reflective spheres (most common), LED markers, or other types of markers that can be easily detected by the camera.

For 3D analysis, more than one camera is needed with complex setup as the observation should focus on all planes of movements, because the 3D mocap captures not only the movements of a subject but also the three-dimensional position and orientation of the body parts. As 3D analysis requires reconstruction of the points of interest on the subject for all the time, the points should be visible for at least two cameras for every moment. Before any recording, the cameras or sensors have to be fixed and the origin of the laboratory system must be defined. Moreover, the mocap system needs to be calibrated, to establish the relationship between the marker positions and the subject's joint angles and segment lengths. This calibration process involves recording specific poses, such as the static position, to align the virtual character with the subject's physical movements. Then, once the system is calibrated, the subject performs the exercises required in front of multiple cameras that record and capture the movements from different angles.

The exercises are quite standard, and they are walking at normal speed, walking faster, usually also squatting, and going up and down the stairs. This captured data is processed using dedicated software, such as Vicon [12] or Xsense, that analyzes the positions, orientations and depth information of the markers/sensors and creates a 3D digital representation of the subject's movements. This software analyzes the data and reconstructs the subject's movements, providing information on joint angles and torques, segmental positions, and the other kinematic parameters related to walking. This data can be further refined and edited if it is necessary. With the reconstructed data, it could be possible to calculate the various gait parameters such as the ones cited in the previous paragraph (step length, cadence, walking speed, stance phase duration etc.) and temporal-spatial parameters such as foot contact times and swing times.

1.3.1. Optoelectronic and stereophotogrammetric systems

To capture 3D motion data is used the stereophotogrammetry [18], that combines the principles of stereoscopic vision and photogrammetry. Stereophotogrammetry works using multiple photographs of the subject and its movements, taken from different positions and angles to ensure that they cover all the scenes. First of all, there should be defined internal parameters (focal length, principal point coordinates and distortion coefficients) and external parameters (position of the camera referment system with respect to the absolute referment system) to calibrate the stereophotogrammetric system. Then, when a part of the scene is acquired, a two-dimensional photograph of a three-dimensional space, to describe the mathematical relationship between these two systems, is used in the pinhole camera model. The concept of a pinhole camera is a closed box into which a single tiny hole is made with a pin, through which light may enter and hit a photosensitive surface. The pinhole camera model mimics the geometrical projection carried out by a pinhole camera. The entire optics and aperture of a camera are reduced to a single point, that is called center of projection. The photosensitive surface is assumed to be planar and is represented by the image plane. To determine where a 3D point is depicted in the image, it is sufficient to consider a straight line from that 3D point and go through the center of projection. The intersection between this straight line and the image plane gives the desired image point of the 3D point. Instead, the line passing through the center of projection that is orthogonal to the image plane is called the optical axis and their intersection is the principal point.

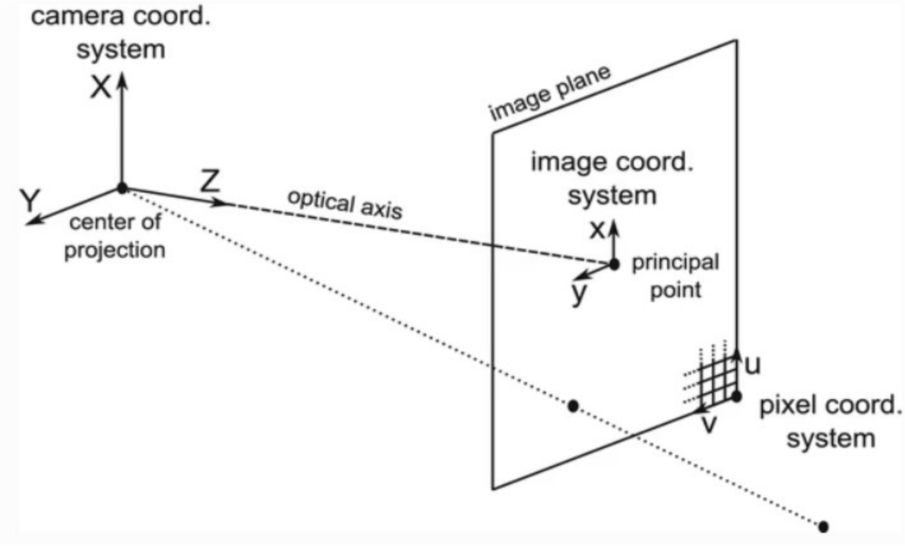


Figure 6: Pinhole camera model. The distance between the center of projection and the image plane is the focal length f [19].

To get the 2D image point's coordinates in the image coordinate system (x, y), having the coordinates of a 3D point expressed in the camera coordinate system (X, Y, Z), it should be applied these formulas [19]:

$$x = f\frac{X}{Z} \; ; \; y = f\frac{Y}{Z}.$$

In order to model cameras in motion or multi-camera systems, one needs to describe the position and orientation of a camera. To do this, the laboratory coordinate system is considered as the world coordinate system. Let **t** be the coordinates of the center of projection in the world coordinate system and let the rotation matrix **R** represent the camera's orientation. Then, a 3D point is mapped from the world to the camera coordinate system, as:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} \mathtt{R} & -\mathtt{R}\mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{pmatrix} \begin{pmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{pmatrix}$$

Once getting all the photographs from the multi-camera system, they are analyzed and corresponding points or features in the images (using for example SIFT algorithm) are identified. To get the points/features position in 3D space, is used the triangulation method [19].

These 3D points are then processed and used to create a dense point cloud representation of the scene. This point cloud consists of a large number of interconnected points, each representing a specific location in 3D space. From this point cloud is created a surface mesh to represent the object's geometry. The captured photographs are projected onto the surface of the 3D mesh, helping to enhance the visual appearance of the 3D model subject.

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. However, despite its potential benefits, stereophotogrammetry has some limitations: the technique can be time-consuming and very expensive, and requires specialized equipment and expertise to implement effectively [18]. 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 [18].

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. There errors can be classified into [2]:

- Instrumental errors: they could be systematic (i.e., an inaccurate calibration) or casual (i.e., electronic noise). The first case occurs when there are bad estimations of the model parameters, the measurement system model has limited validity, or the entity of the error depends on the amplitude of the visual field and from the marker position. The second case occurs during the digitization process of the image or during its elaboration. To minimize these errors, low pass filtering or the self-tuning of the cut frequency is used.
- Wrong positioning of markers on the anatomical landmarks: identifying the landmark can be complicated and requires specialized and experienced staff.
- Soft tissue artifacts (STA): these artifacts are caused by sliding of the skin, deformations, movement of the underlying muscles, and the presence of adipose tissue.

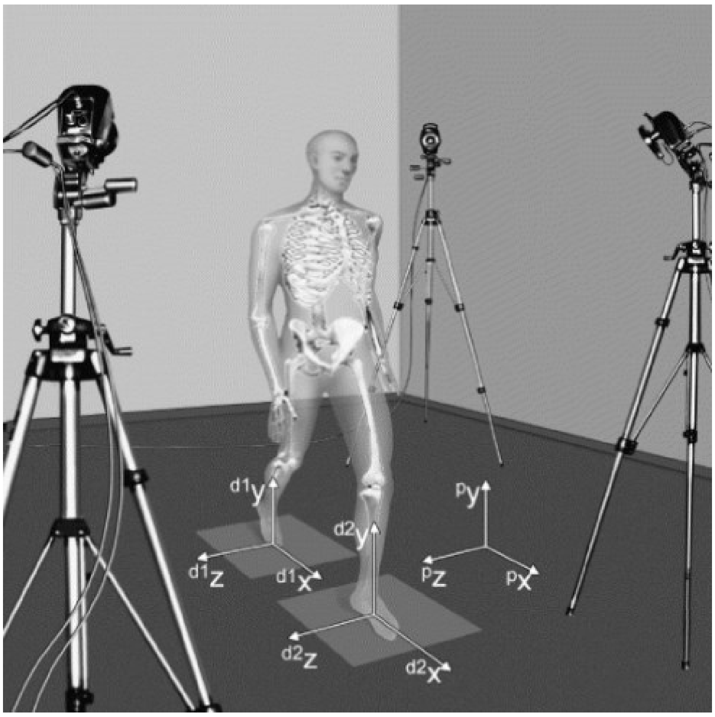


Figure 7: The human movement analysis laboratory. Basic measurement instruments are depicted together with their systems of axes (p: photogrammetry; d: dynamometry) [18].

Another technique of human gait analysis is well known where active or passive markers are attached with the point of interest of the human body. These marker points are traced by an optoelectronic system that converts the emitted or reflected light signals from the markers into electrical signals. These electrical signals are used to construct models for gait analysis. The active markers emit light signals where small LEDs could be used to attach the point of interest of the subject. On the other hand, the passive markers reflect the lights coming from the optoelectronic device where Infra-Red (IR) light is used to illuminate the markers. Although the sampling rate of an optoelectronic system varies within the range of 50 Hz to 1 KHz, it does not increase the accuracy significantly for a quicker movement rather than slow walking. Some optoelectronic devices, like the Kinect sensor, use an array of camera and depth sensors to produce 3D virtual models of skeleton for gait analysis [18].

1.3.2. Force plates systems

This system consists of metal flat and rigid plates having load cells (most commonly) at each corner of the plates. This mechanism is used to measure the GRF caused by standing or moving subject of experiment, including vertical force (magnitude of weight-bearing forces), anterior-posterior force (forward/backward forces during gait or balance tasks), medial-lateral force (sideways forces during walking), center of pressure (location of the applied forces on the plate), torque and timing of force application. The shape of the force plate can vary based on the design such as square, rectangular, or triangular [2].

Sensors used in force plates could be resistive, capacitive, or piezoelectric, piezoresistive, strain gauge, etc. and they also could vary based on pressure range, sensitivity, and linearity. Force plate mechanism provides the facility to measure the force induced on the plate and its directions. Force transductors based on strain gauges have a spring element on which are applied the forces to measure. When solicited, this spring element generates a deformation on the surface, converting the forces to measure in a as more reproducible and linear deformation as possible. Strain gauge is a measure grid of conductive material obtained by photo etching on a polyimide carrier (the carrier is glued to the spring component, and this gives the electrical insulation). Generally, there are four strain gauges installed in a way that two are compressed and two are stretched by the applied force and they are linked in a circuit called Wheatstone bridge, alimented by a voltage [15].

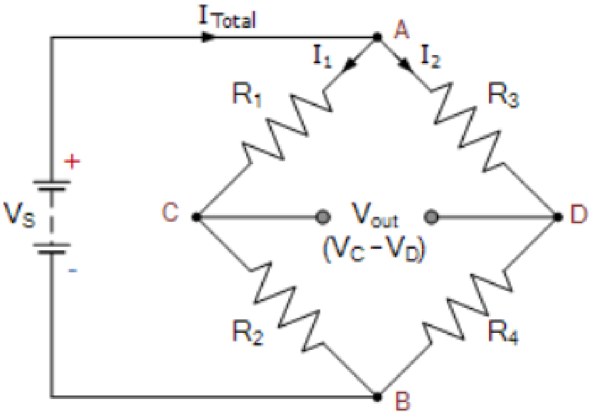


Figure 8: The Wheatstone bridge.

The strain gauge is a conductor, so it is valid that R = ρ $\frac{L}{A}$where ρ is the material resistivity, L is the length and A is the cross section of the conductor. If the conductor is deformed

$$\frac{\Delta R}{R} = \frac{\Delta\rho}{\rho} + \frac{\Delta L}{L} - \frac{\Delta A}{A}$$

The conductor also follows the elasticity rules (ε = $\frac{\sigma}{E}$ and ε_t= -εν, where ε is the deformation, σ is the strength, ν is Poisson's coefficient and E is the Young modulus)

$\frac{\Delta L}{L} = \varepsilon, \frac{\Delta A}{A} = \frac{2\Delta d}{d} = 2\varepsilon_t = -2\varepsilon\nu \Rightarrow \frac{\Delta R}{R} = \frac{\Delta\rho}{\rho} + \varepsilon + 2\varepsilon\nu = \frac{\Delta\rho}{\rho} + (1+2\nu)\ \varepsilon.$

Then it could be introduced the gauge factor K (constant) as K ≡ $\frac{\frac{\Delta R}{R}}{\varepsilon} = (1+2\nu) + \frac{\frac{\Delta\rho}{\rho}}{\varepsilon}$

The Wheatstone bridge is used for precision measures of unknown resistance and for measuring the resistance variance [15]. Then, the four triaxial strain gauges/bridges, with its resistance, formed a twelve channels system in which it is possible to calculate the intensity of the external force, that is obtained with the sum of the forces measured with the single bridges, and the torque respect to the force plate center.

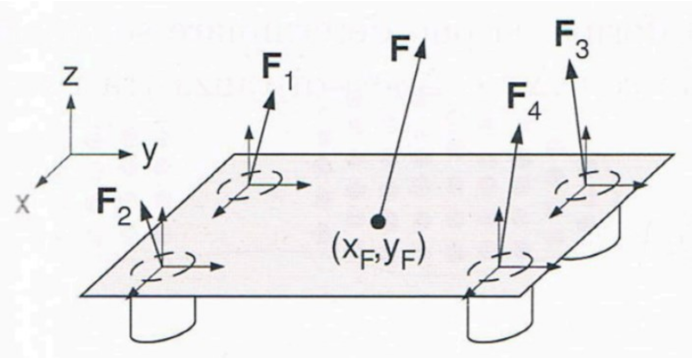


Figure 9: A force plate.

When a person stands or moves on the force plates, the sensors can detect the applied forces and generate analog signals proportional to those forces. These signals are then converted into digital data using an analog-to-digital converter (ADC) and can be sampled at high frequency for precise force measurement.

As the force plates are placed in a fixed location on the ground, the COP of the subject body can easily be calculated but for a long track it may lead to increased cost. Moreover, the suitable position of placing the footsteps on the plate may not be followed properly which could lead to the wrong calculation of COP point. Data captured from foot plates should combine with limbs kinematics information to analyze the principles of gait.

1.3.3. Electromyography (EMG)

Electromyography (EMG) is a diagnostic technique, functional type, used to measure, register, and analyze, the small electric signals produced by muscles during contraction while walking or other locomotion [16].

The fundamental unit for the muscle contraction process is the motor unit, that is a set of the motoneuron, and the muscular fibers innervated by the motor neuron itself. The motor neuron generates a signal, called motor unit action potential (MUAP), that causes the contraction of all, if the impulse is sufficiently wide, or none of the fibers. Depending on how many and which motor units are recruited, the intensity (strength) and duration of the contraction can vary greatly. In muscles used for non-precise movements, such as posture or walking, motor units usually consist of hundreds or even thousands of muscle fibers, otherwise the number of fibers is considerably less. The repeated activation of the motor unit causes a MUAP train. The two most important mechanisms that influence the amplitude and density of the signal are the MUAP recruitment and the firing frequency [17].

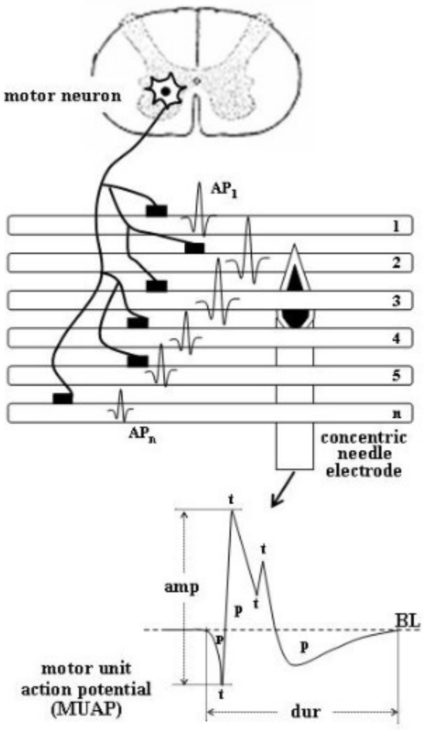

motor neuron
AP1
1
2
3
4
5
n
APn
concentric needle electrode
t
t
amp
p
t
BL
p
p
t
motor unit action potential (MUAP)
t
dur

Figure 10: Schematic representation of a motor unit with n muscle fibers. The algebraic summation of the action potentials (AP) of all the single fibers present in the recording uptake area of the electrode ($AP_1 + AP_2 + \ldots + AP_n$) generates the MUAP. The main parameters of the MUAP waveform are indicated: amp = amplitude; dur = duration; p = phase; t = turn; BL = baseline. [17]

EMG has two types [2]:

1. Surface EMG (sEMG) that is not invasive.
2. Intramuscular EMG (iEMG), that is invasive.

sEMG is much easier to use compared with iEMG as the first one uses electrodes on the skin and the second one uses fine wires inserted into the deep muscle. But sEMG produces less efficient electric signals than iEMG. There are two types of sEMG: passive and active. Active sEMG provides signal amplification at the electrode side that reduces signal noise. To measure these signals, the surface electrodes or fine needle electrodes are placed or inserted into the skin over the muscle of interest. The surface electrodes are placed directly on the skin and are used for the superficial muscles; for deeper muscles are necessary the needle electrodes with which it is possible the direct contact with the interested muscular group.

The most commonly used surface electrodes are those of the Ag-AgCl type, gel, adhesive and disposable. They typically have a circular shape with a single or dual button, depending on the distance at which they are to be positioned. The conductive area, made up of the AgCl coated silver disk, should be about 1 cm or less. They can be further divided into passive and active electrodes: passive electrodes need an external amplification circuit; while the active electrodes have already integrated the pre-amplification circuit, which already allow to reduce the input noise, such as parasitic voltages due to capacitive couplings or to the movement of the electrodes. Based on the positioning surface, three configurations can be observed:

1. Monopolar, in which a single signal detection electrode and a reference electrode are used. This method is used due to its simplicity but is not recommended as it detects all electrical signals in the vicinity of the detection surface.
2. Bipolar, in which two detection electrodes are used, on the surface of interest, and a third one for reference, in an area not active with respect to the investigation. It is the most frequent configuration.
3. Multipolar, in which are used several equidistant detection electrodes plus the ground one. It is used to further reduce the crosstalk between signals and noise, thus obtaining a more significant EMG signal.

Needle electrodes are made of thin, strong, flexible wires, typically alloys of platinum, silver, nickel, and chromium. Depending on the muscle to be analyzed, they will be more or less long (from 25 to 70 mm) and thick (average diameter that varies between 0.30 and 0.45 mm) [16].

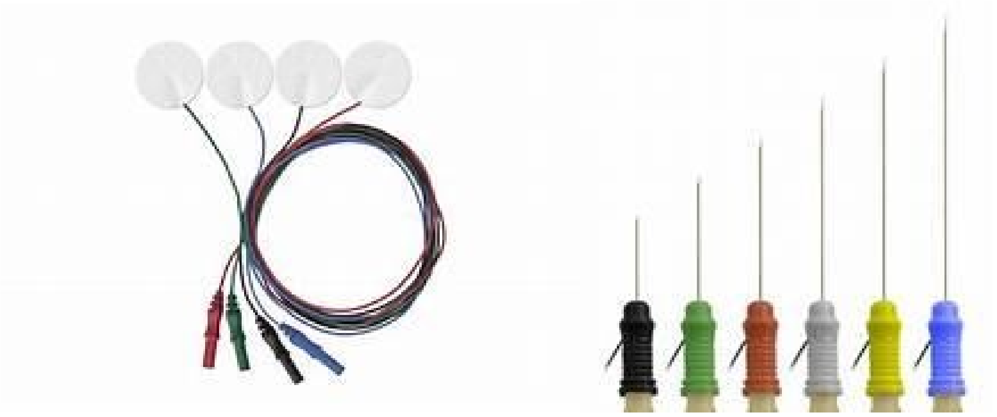


Figure 11: Types of electrodes: Surface electrodes on the left; Needle electrodes on the right.

The electrodes pick up the electric signals, known as electromyographic signals, generated by muscle fibers when they contract and produce electrical impulses. These detected electromyographic signals are weak and require amplification to make them measurable. In fact, the EMG signal is input to an instrumentation amplifier. The instrumentation amplifier must have a high input impedance (ideally tending to infinity) and a low output impedance (ideally tending to zero) and is typically made using 3 operational amplifiers.

Then, the EMG signal is filtered and processed to remove noise and unwanted artifacts [2], ensuring the accuracy of the recorded data that could be then analyzed using dedicated software to extract the parameters of interest, such as the muscle length or strength. High frequency noise can be caused by signal conduction in nerves and by interference to electronic devices such as radios, computers or cell phones and can be removed with a low pass filter. Low frequency noise can be caused by amplifier DC offset, movement of the sensor on the skin and temperature variations and can be removed using a high pass filter. The cutoff frequency for the low-pass filter (LPF) is around 400-450 Hz, instead for the high-pass filter (HPF) is 15-20 Hz. Additionally, if information about the motor unit firing frequency is to be removed from the EMG signal, the cutoff frequency can be shifted between 25 and 30 Hz.

Before the analysis using appropriate hardware/software of the EMG signal, it has to be converted into a digital signal so it would be compatible with these digital instruments. The tool used for this process is the analog-to-digital converter (ADC). The ADC therefore does nothing but convert the analog signal into bits through two processes: sampling and quantization. Sampling consists in transforming the signal from the continuous-time domain to the discrete-time domain. The sampling frequency must be chosen in such a way as to respect the Nyquist theorem ($f_c \geq 2f_{max}$, f_c = sampling frequency, f_{max} = maximum frequency contained in the signal), therefore typically it is around 1000 Hz or even, to avoid loss of signal, 1500 Hz. If this does not occur in the frequency domain, the phenomenon of "aliasing", i.e., overlapping of the spectrum, would occur. Quantization will subdivide the sampled analog signal into a set of levels which each identify a different range of voltage, where a binary coding is associated to each level. The number of quantization bits used is called resolution and for most EMG applications it is 12 bits (2^{12} levels) [16].

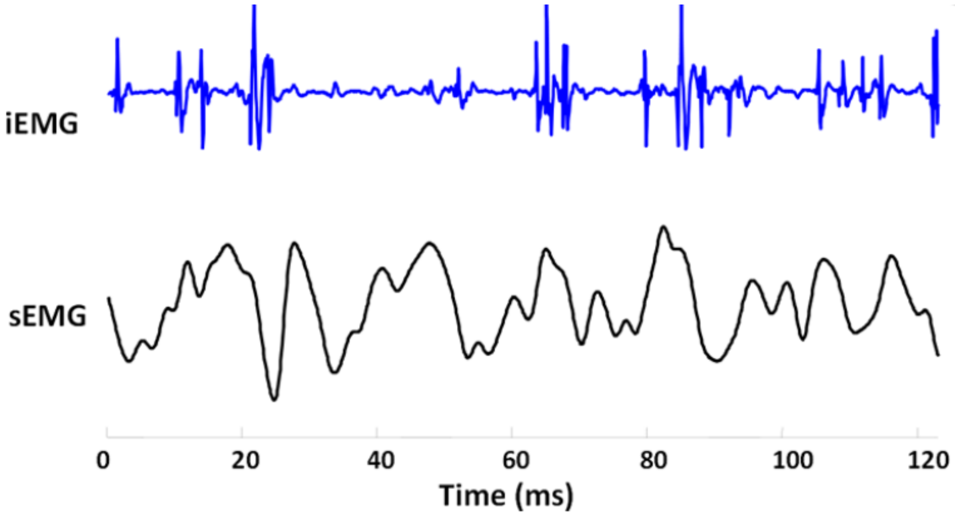


Figure 12: Example of intramuscular (blue curve) and surface (black curve) EMG signals.

1.3.4. Inertial systems

Inertial systems, also known as inertial measurement units (IMUs), are devices that utilize sensors to measure and track an object’s motion and orientation in 3D space. 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 [13].

The inertial system accumulates accelerometers and gyroscopes together and works on the principle of inertial measurements. 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, 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 [13].

Accelerometer and gyroscope provide the data of acceleration and orientation of the attached point through which segment acceleration, segment orientation, and joint position can be achieved for gait analysis. The sampling rate both for accelerometer and gyroscope are the same and vary from 100 Hz to 10 KHz. The tri-axial acceleration and gyration capabilities provide the facilities to analyze human locomotion in a 3D environment. The sensors are small, lightweight, and capable of detecting a large range of angular velocity and acceleration. Inertial sensors have different components: One proof mass that generates the inertial force from an acceleration; An elastic spring, to give a mechanical support to the mass maintaining the neutral position; One dashpot, that controls the proof mass motion and to get the best features of the frequency response; A method to get the electrical output, that is the measurement of the proof mass motion (mechanical measurement) converted into the electrical measurement [2].

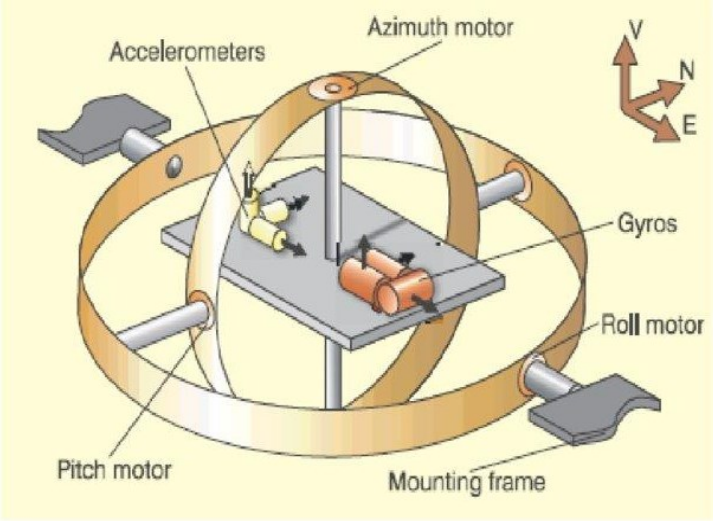


Figure 13: Inertial Measurement Unit (IMU).

To design an accelerometer, it is important to control the proof of inertial mass movement. This mass should be rigid to act in a proper way and there should be a bigger stiffness than the one of the springs along the axis of sensitivity. The spring is important for the kinematic movement of the mass along the axis of sensitivity, which elastic constant depends on the properties and the geometry of the material. The mass should produce a relevant signal to avoid crosstalk with the other axis. The gyroscopes [13] are constituted of rotating mass in a rotating support inside a container that, when the gyroscope rotates around an axis perpendicular to the axis in which the mass rotates around, generates an angular moment around the orthogonal axis of the two that could be measured by a moment sensor. This is explained thanks to the Coriolis force [2], that is an apparent force applied to a body, when the motion is observed from a referral system that is in circular motion with respect to an inertial reference system. This force produces vibrations that are acquired by capacitive or piezoresistive sensors, so the transducer of the gyroscope should oscillate to work as a sensor.

The system of the inertial sensors is formed by three accelerometers and three gyroscopes built orthogonally to measure the three linear and angular accelerations in an independent way, a hardware to process the signals, a battery, a transmitter of the signals and a container.

The disadvantage of using this system is the skin movement artifacts which can affect the reading of acceleration and gyration. It is also a challenging task to identify the segment length and exact rotational axis. Moreover, the acceleration is relative to the position of the Inertial Measurement Unit (IMU) system.

The inertial sensors could give some errors based on repeatability, the capacity of giving the same measurement if it is repeated in time in the same conditions, stability, the capacity of giving the same output if it receives the same constant input, or drift, the variation of the signal output in time. IMU sensors always measure the force linked to acceleration gravity, which induces an error because it is detected as a constant acceleration by the accelerometer. For this reason, before any measurements, the calibration is done in static conditions. Double integration induces quadratic (accelerometers) or cubic (gyroscope) errors that depend on sample frequency. To reduce the error, the Kalman filter is used [2].

For human movement analysis, there are various alternatives like inertial sensors that use technologies like accelerometers, magnetometers, and gyroscopes to estimate joint kinematics. The advantages of this technology consisted in portability, accuracy, and reliability, even though they remain expensive and require particular algorithms. Moreover, the inertial sensors are linked to a PC allowing a wide variety of applications, in fact, this system could be used in every environment. In addition, each recent smartphone has an inertial measurement unit (IMU), so it is useful for long term analysis of the activity, but it is demonstrated that the resulting data have a low reliability [14].

1.4. Gait-analysis protocols

The markers are placed in the body according to different protocols that have been developed, in order to reduce as much as possible, the errors introduced in the previous paragraph, by the misplacement of markers on landmarks; 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. 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].

First of all, in a protocol, it is necessary to describe the movement of the musculoskeletal model during a specific task, such as walking. This implies the definition of a global and/or local set of axes. The global referment system in the laboratory is given by the stereophotogrammetric system and the calibration procedure used and the local referment system is associated with the analyzed task. The local referment system requires the definition of a technical referment system associated with the anatomical segment of interest and also an anatomical reference system. These systems are defined according to the anatomical axes and planes indicated in [Fig.1].

The technical referment systems can be generated by one technical cluster. This type of system is used to describe the movement of a segment considering the instantaneous position of at least three not aligned markers associated with the anatomical segment. Their trajectories are reconstructed by a stereophotogrammetric system. The three markers are technical markers, and they are placed considering the visibility requirement from at least two cameras and in order to contain the movement with respect to the below bone. This configuration makes the technique cluster with an arbitrary position and orientation, and it is dependent on the collocation of markers and from the analytic procedure. This technical referral system could require the definition of virtual markers, that is that they do not correspond to any anatomical point, and they can be constructed geometrically by other markers of the technical cluster [2].

At the end there is also the possibility to define another technical referment system, that is called morphological technical referment system. As it is understood by the name, this system is linked to the morphology of the bone and to the task, and it is not mandatory to define the marker set. So, for the definition of a protocol for kinematic analysis it is necessary to define a marker set for the cluster technical reference system and a set of anatomical points for the anatomical reference system. It should be respected the conventions of three dimensionality, minimum three markers per segment, visibility of the markers, maximum affability of the definition of the anatomical planes and axes, repere points easily identifiable, high accuracy, easily applicability of the markers, reduction of the time in patient preparation, reduction of the instruments, reduction of the time for the data collection, ensure the minimum variability of intra/inter operator and inter subject results [20]. The following paragraphs are going to describe the most used gait protocols by researchers.

1.4.1. Helen Hayes – Davis protocol

The Davis et al. protocol (1991) [21] is the most used protocol in movement analysis internationally. 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. It involves the detection of the subject's anthropometric parameters (height, weight, distance between the anterior iliac crests and thickness of the pelvis, leg length, distance between the femoral condyles or diameter of the knee, distance between the malleoli or diameter of the ankle), used to estimate the position of the joint centers. Subsequently it is proceeded with the positioning of 22 markers in specific landmarks, such as [24] [Fig.14]:

- acromion and seventh cervical vertebra (C7)
- iliac crests and sacrum
- greater trochanter, lateral femoral epicondyle and a marker positioned on a bar placed approximately mid-thigh
- head of the fibula, malleolus and a marker placed on a bar positioned approximately halfway up the tibia
- fifth metatarsal and heel (only in the standing phase)

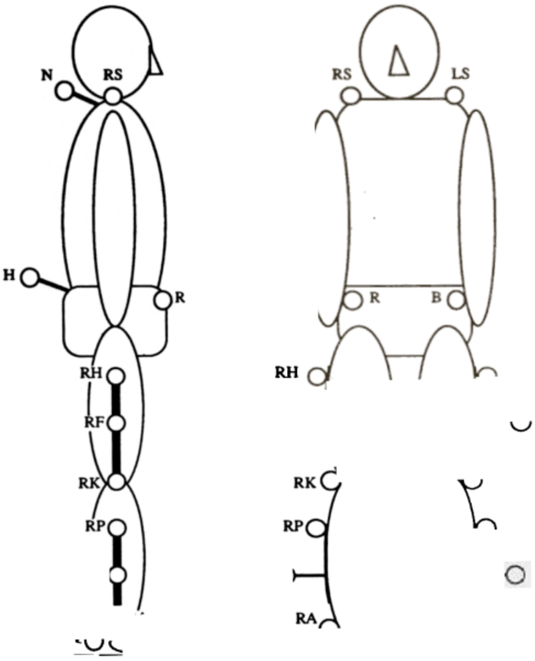


Figure 14: Davis Marker set configuration.

Once the markers have been placed, it has been proceed with a static acquisition in an upright position which allows us to record the initial positions and define the reference systems as well as the joint centers. The last step consists in making the subject walk, with a speed considered normal by the subject itself, within the acquisition volume for a number of times necessary to obtain at least three good walks. Finally, from the trajectories of the markers it is possible to obtain the displacement, speed and acceleration of the body segments and the joint angles [20].

The anthropometric parameters, measured thanks to the embedded coordinate system, may be determined for any body segment (assumed to be rigid) that has at least three non-colinear markers attached to it. The location of hip, knee, and ankle joint centers are calculated relative to the associated embedded coordinate system origin, i.e., the hip center location relative to the origin of the pelvic embedded coordinate system in pelvic coordinates, the knee center location relative to the origin of the marker-based thigh embedded coordinates system in thigh coordinates, and the ankle center location relative to the origin of the marker-based shank embedded coordinates system in shank coordinates [24].

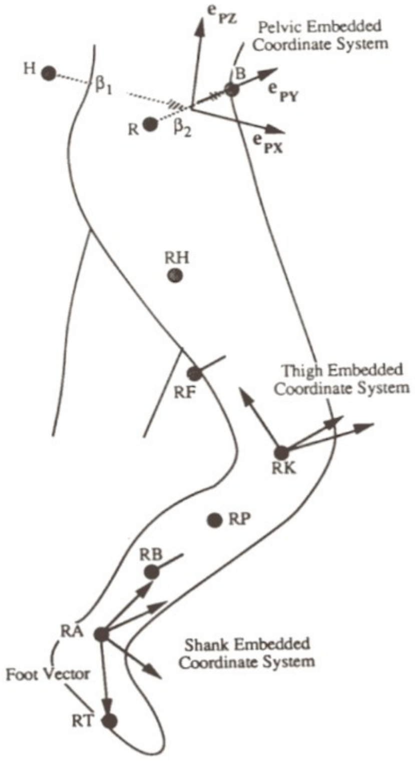

e_PZ
Pelvic Embedded Coordinate System
H
β1
B
e_PY
R
β2
e_PX
RH
Thigh Embedded Coordinate System
RF
RK
RP
RB
RA
Shank Embedded Coordinate System
Foot Vector
RT

Figure 15: Lower extremity, embedded coordinate system used to compute the joint angles and as frames of reference for the joint moment reactions.

The limb rotation algorithm is based on the determination of Euler angles with an y-x-z axis rotation sequence that produces as output a transformation matrix consisting of three joint angles in the three directions. Then according to Davis protocol, it is possible to have a scheme to evaluate the kinetics data for dynamics calculations: the net 3D joint moments at the hip, knee, and ankle are computed via Newtonian mechanics through the application of Newton's Second Law and Euler's equations of motion:

$M_x = I_{xx} + \alpha_x + (I_{zz} – I_{yy})\ \omega_y\omega_z$

$M_y = I_{yy} + \alpha_y + (I_{xx} – I_{zz})\ \omega_z\omega_x$

$M_z = I_{zz} + \alpha_z + (I_{yy} – I_{xx})\ \omega_x\omega_y$

Where, (M_x, M_y, M_z) = components of the sum of the external moments applied to the limb segment; (α_x, α_y, α_z) = components of the absolute segmental angular acceleration; (ω_x, ω_y, ω_z) = components of the absolute segmental angular velocity; (I_{xx}, I_{yy}, I_{zz}) = principal mass moments of inertia of the segment; x,y,z = body-fixed coordinate axes, defined as the principal axes and located at the center of mass of the segment [24].

The joint power relative to each segment axis is determined from the product of the joint moment vector component and the associated relative angular velocity component. The segmental mass, mass center location and mass moment of inertia are approximated based on the relationship of Dempster et al. (1959) [24].

The Helen Hayes marker set, only for the lower part of the body (15 markers), is represented in the figure below:

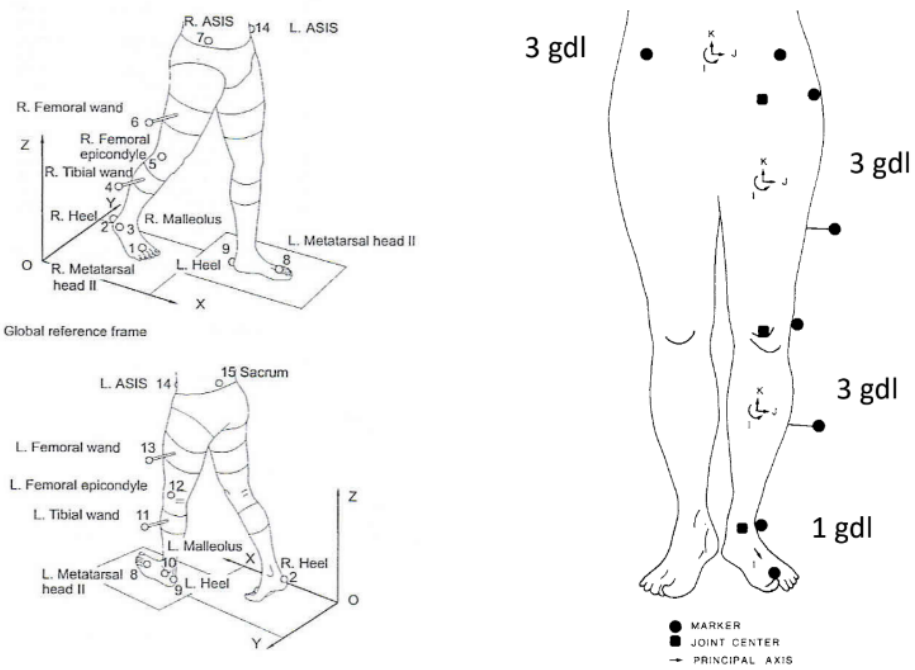


Figure 16: Helen Hayes marker set (1990). The 15-marker system that uniquely defines the position of each segment in 3D space. Anterior and posterior view on the right; Marker configuration and embedded coordinate systems on the left [21].

From Helen Hayes marker set derive other two marker sets: GaitLab and Cleveland Clinic, respectively on the left and on the right of the following figure:

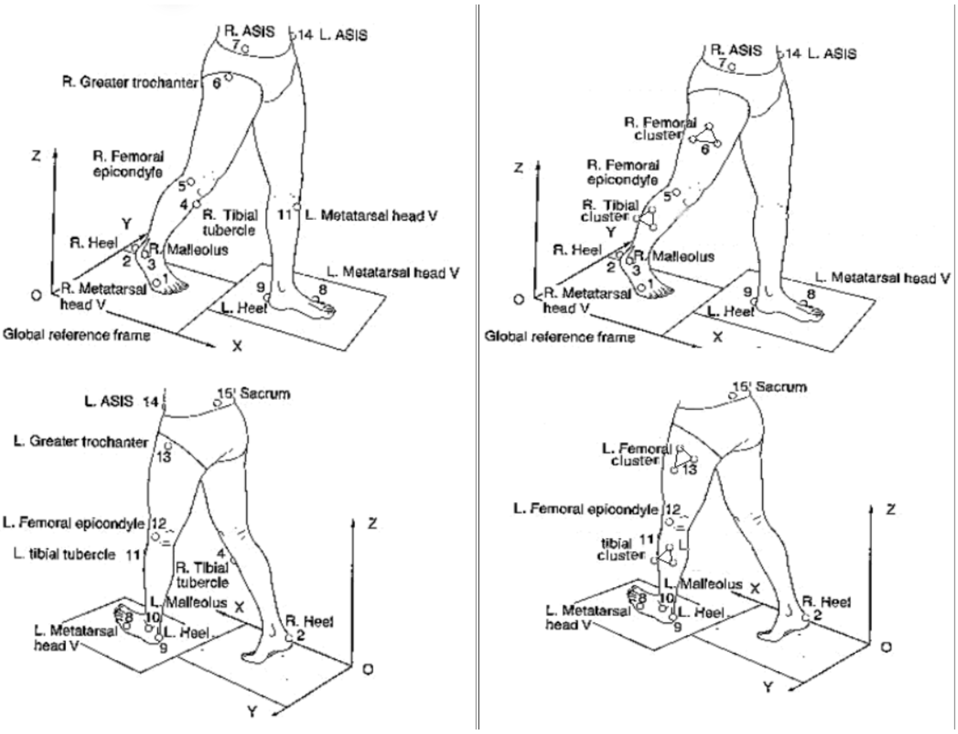


Figure 17: The 15-marker system that uniquely defines the position of each segment in 3D space for GaitLab and with clusters for Cleveland clinic (anterior and posterior view).

1.4.2. C.A.S.T. protocol

C.A.S.T. (Calibrated Anatomical System Technique) protocol (1995) also known as the "Calgary Cambridge Foot and Ankle Protocol" [22], 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. This protocol was developed to define anatomical and technical reference systems starting from the idea of solving soft tissue artifacts, a problem that arises when markers are applied to the skin.

In this protocol the technical and anatomical reference systems are defined through an intermediate procedure called anatomical calibration. Anatomical calibration can be accomplished in two ways. If the anatomical landmark is easily palpable on the skin, a marker is placed on that point and thus the soft tissue artifact (STA) can be overlooked. Subsequently, the positions of the technical cluster (technical reference system) and the marker placed on the anatomical landmark are acquired. This procedure is then performed for each anatomical point. The second way to carry out anatomical calibration consists in directing a wand, made up of markers placed at known distances, to the anatomical point of interest. This allows the position of the point to be detected with respect to the technical reference system and this procedure must then be repeated for all the repere points [20].

After obtaining information on the position of the technical reference system with respect to the global reference system and on the position of the reference point with respect to the technical system, it proceeds with the dynamic acquisition of the technical reference system and finally, with a reconstruction algorithm, the global position of the anatomical landmarks is reconstructed.

The CAST protocol has the advantage of minimizing STA; however, it requires a high number of calibrations due to the 33 markers used.

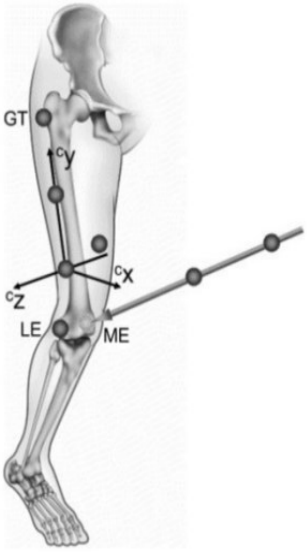


Figure 18: Anatomical calibration using a wand.

1.4.3. IOR – GAIT protocol

IOR-Gait protocol (Leardini et al. 2007) [23], also known as the "Istituto Ortopedico Rizzoli Gait 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.

This protocol was born as a union of the protocols described previously, combining the need to use a limited number of markers with the aim of obtaining accurate data. This was achieved by considering a few markers from the Davis protocol and adopting the calibration procedure characteristic of the C.A.S.T. protocol. In the IOR-Gait protocol it is preceded with the calibration of six markers, three for each limb; however, the estimate of anatomical landmarks is reduced due to the adoption of only the anatomical reference system and not the technical one.

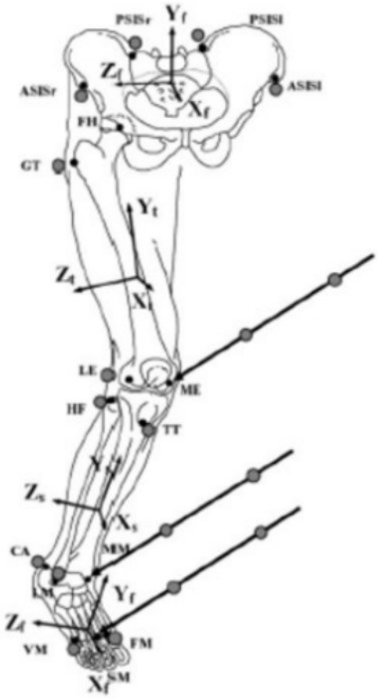


Figure 19: IOR-Gait protocol marker set (gray points) and anatomical repere points (black points); the wand indicates the calibration points.

2. MUSCULOSKELETAL (MSK) MODELING

2.1. Introduction

The musculoskeletal (MSK) system is the combination of the muscular and skeletal systems in the human body. It provides structural support, facilitates movement, protects vital organs, and enables various physiological functions. It consists of bones, muscles, tendons, ligaments, and connective tissues. The computational approach used to simulate and analyze the interactions between the tissues of the MSK systems is Musculoskeletal modeling.

Musculoskeletal models are typically built using anatomical data obtained from medical imaging. These data are used to create three-dimensional representations of the MSK system. These models include information about joint kinematics, muscle architecture and physiological properties. The muscles in these models are represented as actuators that can contract and produce forces, in fact the muscle modeling involves characterizing muscle properties allowing the estimation of muscle forces, joint torques and joint reaction forces during different activities, such as walking. The movement analysis, so for example walking, with MSK models, is done applying external forces and motion data as inputs and estimating how the parameters of the model change over time.

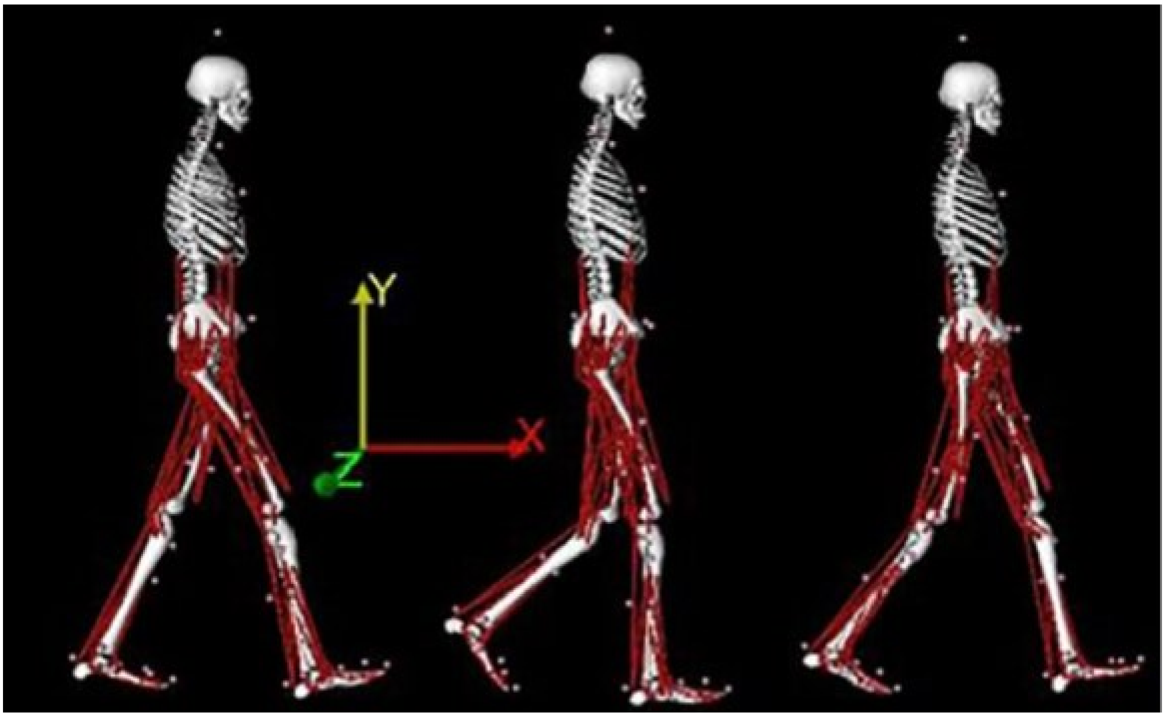


Figure 20: Example of a MSK model. In red are represented the muscles as actuators. The model is represented with respect to the global coordinate system. [25]

2.2. Musculoskeletal modeling analysis

2.2.1. Musculoskeletal model

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]. 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 [26]. Various software have been developed to model and analyze human movement (i.e., AnyBody, and OpenSim), and there is always an increase in literature reporting motion analyzes based on these software systems. 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. First of all, to create a motion simulation, the definition of a dynamic model of the musculoskeletal system and its interactions with the environment is necessary. Usually, a musculoskeletal model consists of [26]:

- 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 lack degrees of freedom of the model.

2.2.2. Scaling

In order to attribute to the model, the mass and anthropometric characteristics of a particular subject, the generic model is scaled 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.

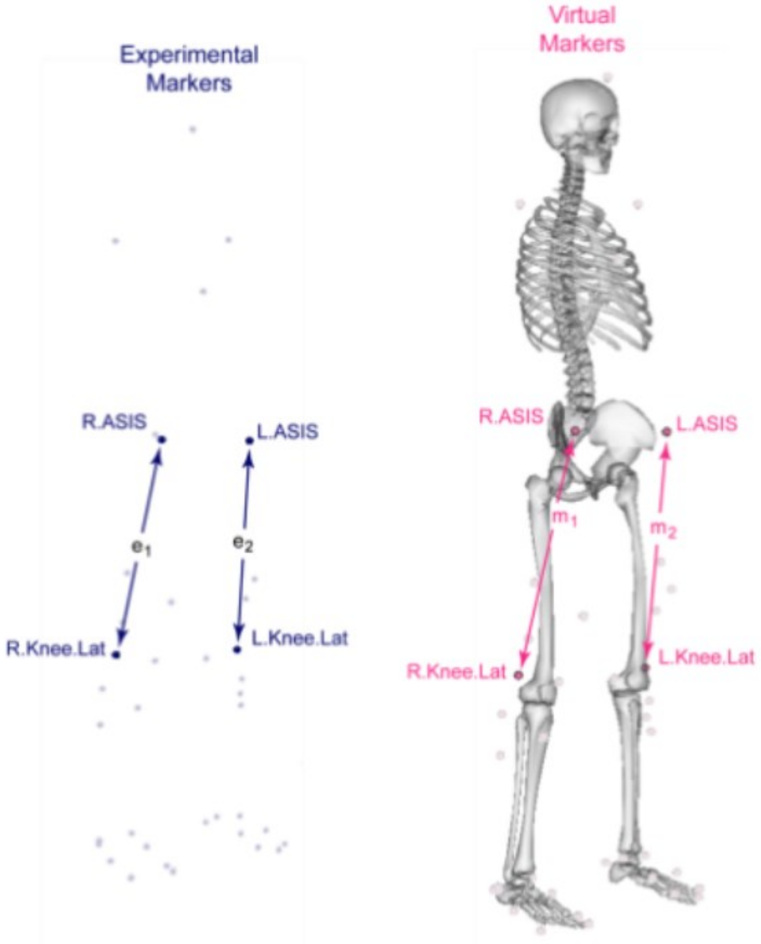


Figure 21: Experimental markers position (blue markers) measured with motion capture. Virtual markers (pink markers) are placed in correspondence on anatomical models. The distances between experimental markers (e_i) and the distances between virtual markers (m_i) are used to calculate scaling factors [20].

2.2.3. Inverse Kinematics (IK)

Kinematic analysis represents the position, motion, and trajectories of interest points (or anatomical segments or the whole body) for each instant of time during the task execution of the subject to describe the locomotion pattern where mass and force are not considered. Since the morphology of the anatomical segments is very complex, it is substituted by a simple morphology, that is the rigid body [27].

The inverse kinematics (IK) problem is the problem of finding a vector of joint variables which produce the known desired end effector location and orientation (relative to the base frame) of a serial chain manipulator. Also, all the geometric link parameters are known. An end effector, in robotics, is the device at the end of a robotic arm, designed to interact with the environment. In human biomechanics analysis, the end effector is the human arm or the lower limb. An inverse kinematics problem could be well posed that means that exists a unique vector of joint angles that correspond to the desired end effector location, but it could be also ill posed because, for a given end effector position (x, y), there could be either no solution (in this case the target location is infeasible, that means, out of the reachable workspace), or because there are many solutions (one, two or multiple) [28]. In general, ill-posedness of the latter type can be interpreted as arising from two distinct phenomena. First, the inverse is non unique in a global sense but perhaps still well-defined locally because of the existence of multiple solution branches. This occurs for manipulators both with and without excess degree of freedom (DOF). Second, the inverse kinematics problem for a manipulator with redundant DOF is locally ill-posed in that each solution branch contains an infinite number of solutions. In general, the set of solutions to the inverse kinematics problem for a redundant manipulator consists of a finite number of nontrivial self-motion manifolds (the solution branches) in the configuration space.

The general problem of IK is to find a solution or multiple solutions when a 4 × 4 homogeneous transformation matrix is given [29]:

$$H_n^0 = [R_n^0 \; o_n^0 \; 0 \; 1\,] \in R^3$$

The above homogeneous transformation matrix provides the end-effector orientation (3×3 matrix) R_n^0 and position (a 3×1 vector providing the coordinates of the end-effector origin o_n^0) with respect to the base frame. From n-DOF manipulator relationship,

$$\mathrm{H} = T_n^0 = \mathrm{A}_1(\mathrm{q}_1)\cdot\ldots\cdot\mathrm{A}_n(\mathrm{q}_n)$$

it can be possible to find the solution for joint variables q_1, q_2, q_3,…, q_n, unless the problem is ill-posed. The position of the end-effector (X) is a function of joint variables q_i ($X = f(q_i)$), then solving for IK, required to solve for the joint variables when the end-effector's position is provided by:

$$q_i = f^{-1}(X) \qquad (1)$$

To solve joint variables, there are two distinct methods [29]:

1. Analytical Method:

The analytical method solves Equation (1) analytically to produce all possible solutions for a given inverse kinematic problem. Depending on the DOF of a robot manipulator, the number of solutions varies. Again, For manipulators with a fewer DoF than that of the end-effector, there are possibilities of no solution; in cases where there are too many degrees of freedom, there can be infinite solutions. The analytical method is further divided into two methods called geometric and algebraic methods, based on how the equations are. As it moves ahead to solve the IK problem for different robot manipulators using the analytical method, it could be observed that as the number of joints increases, the complexity of solving inverse kinematic using the analytical method becomes more unfeasible.

a. Geometric Method:

The geometrical approach to solving IK decomposes the geometry of the robot manipulator, consisting of linkage and joints, into multiple planar problems. Converting a 3D problem to several 2D problems is quite straightforward for certain configurations of robots. The equations found after solving each 2D problem are then used to solve joint variables using algebraic manipulation. An example uses the geometry of the manipulator to form a trigonometric relationship between the desired end-effector position and joint values.

b. Algebraic Method:

The algebraic approach to solving IK uses the forward kinematics solution for a robot manipulator ($X = f(q_i)$) and uses the homogeneous transformation matrix T_n^0 to find the necessary joint equations and solve them algebraically. There is no systematic way to solve multiple solutions like in the geometrical approach. It does not include observing the geometrical aspects of the robot manipulator to find the equations. When comparing with the geometric approach, the algebraic approach relies on the solution of forward kinematics whereas the geometrical approach only relies on the geometry of the robot manipulator (obviously both require the end-effector position in the form of a 4×4 homogeneous transformation matrix).

2. Numerical Iterative Method:

Numerical solutions, unlike analytical solutions, are robot independent and may thus be used for any kinematic structure. Only when the polynomial equations have four or fewer degrees may analytical solutions be found. The analytical solution fails to provide a solution if the degree of the polynomial is higher. As a result, even if the DOF is less than or equal to 6, many manipulator geometries are not solvable. In general, higher numbers of non-zero geometric parameters correspond to a higher degree of polynomials, which makes the analytical approach unfeasible. It uses a numerical approach to solve such manipulator geometries. There are different ways to solve IK using the numerical method; the Newton-Raphson method is one of the faster ways to find solutions for nonlinear equations.

a. Newton-Raphson method:

It is known that a robot manipulator's end-effector position (X) can be defined as a differentiable function of joint variables. Assuming that the robot manipulator is with only revolute joint, it can be written that:

$$X = f(\theta)$$

Let x_d and θ_d be the desired end-effector position and joint values to reach the desired joint variable. Thus, the inverse kinematic can be defined as:

$$x_d - f(\theta_d)=0$$

This equation represents the Newton-Raphson method, and the goal is to find θ_d.

According to the Newton-Raphson method, nonlinear polynomials can be solved by making an initial guess and using that to converge towards a root/solution (it doesn't necessarily solve for the closest root).

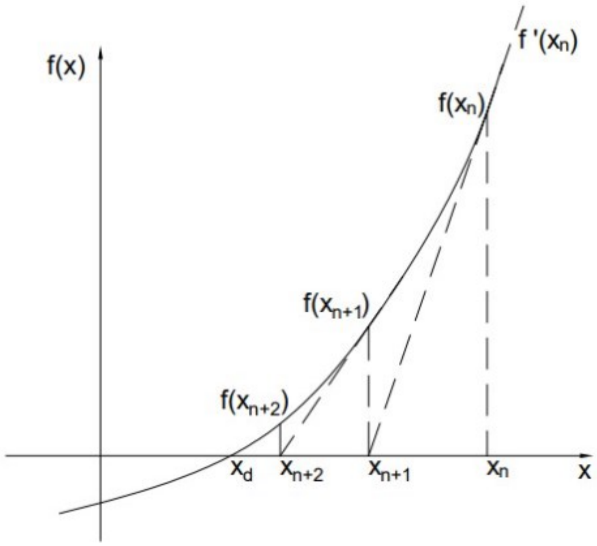


Figure 22: Solution illustration (polynomial curve) for the numerical iterative method [29].

The Newton-Raphson method begins with the selection of an initial guess (x_n), which is closer to the solution x_d, from where we find the slope f‘(x_n). When extended it can be seen that the slope intersects at the x-axis giving a new point (x_{n+1}). Then, it is evaluated the slope f‘(x_{n+1}) at point x_{n+1}. The slope then intersects at a new point x_{n+2} and the process repeats converging towards the solution x_d.

It is preferable to derive the formula for a better approximation (x_{n+1}) according to the figure above. The tangent f′(x_n) can be defined as

$$y = f'(x_n)(\mathbf{x} - \mathbf{x}_n) + f(x_n)$$

The x-intercept (y = 0) of the tangent f′(x_n) will give the point x_{n+1}. Therefore,

$$0 = f(x_n) + f'(x_n)(x_{n+1} - \mathbf{x}_n) \therefore x_{n+1} = x_n - (\frac{df(x)}{dx})^{-1} f(x_n)$$

Since it has been considered a vector in IK, x is equal to

$$\vec{x} = [\theta_1 \vdots \theta_n]$$

Thus,

$$x_{n+1} = x_n - (\frac{\partial F(x)}{\partial x})^{-1}|_{xn} F(x_n)$$

Where, $\frac{\partial F(x)}{\partial x}$ is the Jacobian matrix.

The iteration will continue till it has reached the end-effector position. Sometimes, the function curve is such that it doesn’t reach the end-effector position even after many iterations (the approximation doesn’t converge towards the solution). In such cases, it is taken a new initial guess (x_n) and starts new iterations [29].

2.2.4. Ground reaction forces (GRFs)

The branch of mechanics that studies the effects of forces on motion is known as kinetics. Understanding the mechanisms of joint injuries and musculoskeletal diseases requires kinetics analysis of human motion. Walking is a fundamental and frequently utilized motion, and thus can be used to understand the mechanical pathways of musculoskeletal problems of lower limb joints. Kinetic analysis uses mass and force resulting from the movement and impact of the musculoskeletal system of the human body. Impact force transmission of human body lower limbs is important to analyze to find out the condition of catastrophic failure or damages during walking [30]. The impact force can be defined as the force that occurs at the contact moment between heel and ground.

Kinetic analysis of walking typically requires the coordinates of body movements and ground reaction forces (GRF). Ground reaction forces are the forces that could be measured based on the interaction of the body (in particular, the feet) with the ground, according to Newton's third law: "For every action, there is an equal and opposite reaction" [31]. A person standing motionless on the ground exerts a contact force on it, equal to the person's weight, and at the same time, an equal and opposite ground reaction force is exerted by the ground on the person. In order to move, the person puts force into the ground and the ground applies forces back (GRF) onto him/her which then propels the subject forward. There is no GRF during the swing phase.

The GRF has a point of application on the foot, a magnitude, and a direction, or line of action. In static situations these all remain constant, with the magnitude equal to body weight. However, in dynamic situations such as locomotion they typically vary in a repetitive trend. When the line of action of the GRF lies at a distance from the center of rotation of a joint, it creates an "external moment." The greater the perpendicular distance of the line of action of the GRF to the joint center (the "lever arm"), the greater the external moment will be. If the GRF is aligned close to the joint center, the external moment is smaller. If the GRF passes through the joint center, then no external moment is generated. The presence of an external moment tends to cause motion at a joint and usually requires the generation of an opposing internal muscle moment to create equilibrium or to control this motion. The greater the external moment, the greater the required internal moment likely will be. In static situations, the internal and external moments will be equal and opposite and therefore in equilibrium. However, in dynamic situations, such as locomotion, it may be necessary for the moments generated by the muscles to be slightly greater or less than the external moments to control angular motion at the joints. During the stance phase, the GRF alignment may pass from one side of the joint center to the other, “switching” moments from, for example, flexion to extension.

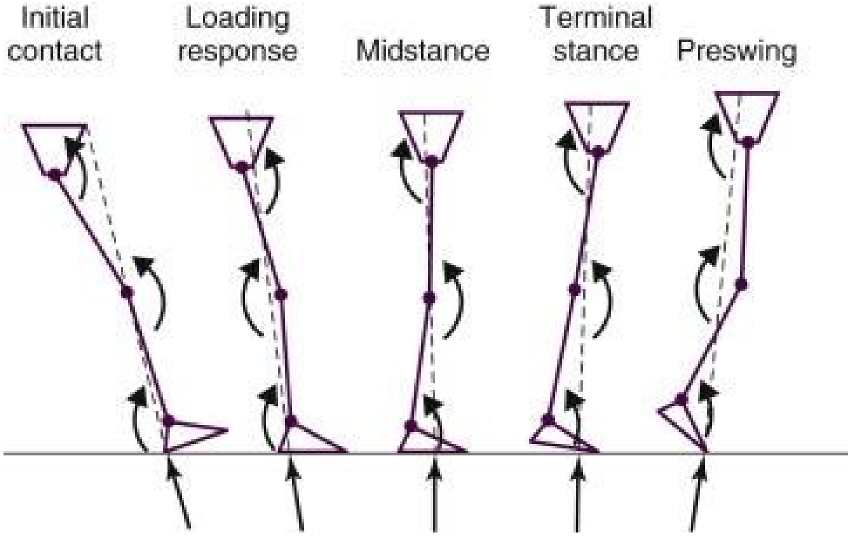


Figure 23: GRF trend during stance phase.

There are three components of induced GRF [2]:

1. Anteroposterior GRF (AGRF, PGRF), this is the force exerted in the front-to-back direction that is the x axis. When walking forward, the ground exerts an anterior force (AGRF) on the foot, which helps to move forward; vice versa, when the walking arrives to a stop, the ground exerts a posterior force (PGRF) on the foot which helps to slow down, and then to stop the motion;

2. Vertical GRF (VGRF), this is the force exerted in the up-down direction that is the y axis. When jumping 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. Mediolateral GRF (MGRF, LGRF), this is the force exerted in the side-to-side direction that is the z axis. When changing directions during a motion (walking or running), the ground exerts a medio-lateral force on foot, which helps to move in the desired direction (left or right).

In addition to these forces along the three orthogonal axes, there is also a moment along the vertical axis at the center of the pressure (COP) placed on the foot sole. The GRFM is measured using force plates. GRFM prediction is of great interest for simulating walking, and they can be estimated using equations for inverse dynamics of human motion.

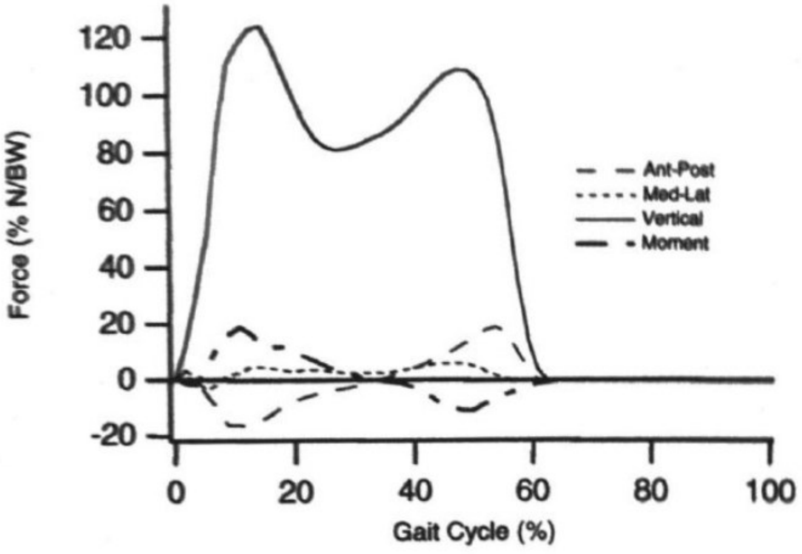


Figure 24: Ground reaction forces and moment (GRFM) in typical walking. [32].

2.2.5. Center of pressure (COP)

The center of pressure (COP) is the point of application of resultant ground reaction force (vertical, anterior-posterior, and medio-lateral components) acting on a subject's feet during standing or other activities like walking (COM). If both feet are in contact with the floor, the COP is a point between them.

According to D.A. Winter [33], when the body is in the standing position, the chain that describes the biomechanics of the body could be considered as an inverted pendulum to the ankle with a one degree of freedom on the sagittal plane. COM reflects on the effective movements of the body, while COP reflects on the muscular forces. With respect to the ankle, COM is the distance from the gravity force vector and COP is the distance from the ground reaction force vector. The equation that describes the inverse pendulum is [2]:

$$COP - COM = k*COP$$

That is, horizontal distance between COP and the center of gravity projection is inverse proportional to the acceleration of the COP movement. To maintain a correct stability, there is the need of moving the body mass as little as possible. Instability derives from the difference COM – COP that, if it is positive, the COM acceleration is positive, otherwise is negative. To avoid the fall, there are two mechanisms of control:

1. Acting on COP controlling the muscular activation of the ankle.
2. Acting on COM with relative movements of body parts.

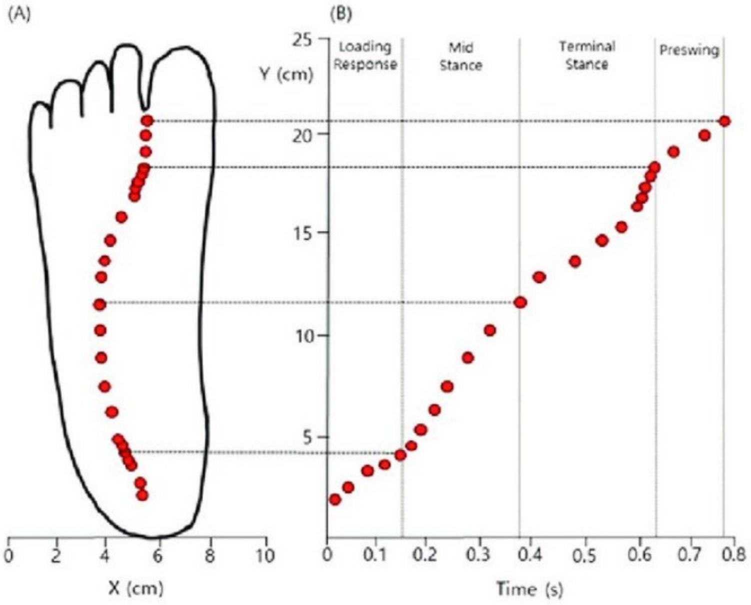


Figure 25: The trajectory of the center of pressure (COP) in the foot. (A) The COP trajectory in X-Y in the stance phase. (B) The COP movement is shown along the Y-axis versus time. [34]

The COP is measured with the force plates or statokinesiogram, and it is an important parameter in assessing postural control and balance indicating how the subject adjusts their body position to maintain stability during standing or dynamic activities. Deviations in the COP over time curve indicate issues with balance or postural control [2]. The COP movement trajectories is an important point also to consider explaining the behavior of walking gait. It helps in understanding how the body distributes and manages forces during dynamic activities and can provide information about gait stability and eventually gait abnormalities.

2.2.6. Inverse Dynamics (ID)

Simulations of human movement dynamics are useful for multiple applications such as the prediction of injury risks, musculoskeletal disease progression, clinical decision making, selecting effective patient treatment or the design of wearable robots like exoskeletons or arm prosthesis.

In the field of biomechanics, inverse dynamics analysis is commonly used to investigate aspects of mechanics, energetics, and control of movement. An inverse dynamics analysis is typically based on measurement of the kinematics of the body segments, often complemented with measurement of selected external forces (e.g., the ground reaction force), required for a desired motion; but kinematics is in general inconsistent with measured external forces, i.e., result in *nonzero* residual forces and torques.

Inverse Dynamics is a function that takes joint orientations (and their time derivatives) as inputs and joint moments and forces as outputs. Not only does Inverse Dynamics give the joint moments required to make a certain rotation, but it also gives the forces and torques from the link. Inverse dynamics relies on the motion of the subject and a body model to compute the forces that were necessary to produce this movement [35].

Using these data, the net joint moments, and net joint reaction forces are calculated using Newton-Euler equations of motion applied to a model containing a (chain of) rigid segments. These equations are [36]:

- Newton's (linear) equation: F = $m \cdot a$;
- Euler's (angular) equation: M = $I \cdot \alpha$;

where F is the force, m is the mass, a is the linear acceleration, M is the moment, I is the mass moment of inertia and α is the angular acceleration.

These equations describe the behavior of a mathematical model of the limb called a link-segment model, and the process used to derive the joint moments at each joint is known as inverse dynamics. Classically, these equations are solved consecutively for each body segment, starting at a segment for which the number of unknowns matches the number of equations. This is easily done when the motion is an open chain, with no resistance to motion at the terminal segment, since all the kinematic variables are known from motion analysis.

When there is contact of the limb with another object, such as the ground, the forces between the limb and the obstructing object in this closed chain must be measured. This is usually arranged by the technique of strain-gauging, described in the previous paragraphs used in force plates used to measure the ground reaction force during walking and running [36]. There is a potential issue with inverse dynamics though when the articulated body has a closed loop. At this point, the lower limbs form a loop with the supporting plane. There are workarounds to this issue. For example, for biped locomotion, you can adopt a simple approximate solution. The ground force can be distributed to each foot according to the percentage of body support on each leg. If a percent of the upper body weight is supported by the left leg and the remaining percent is supported by the right leg, then the left hip gets a percent of the force and torque from the pelvis and the right hip gets the remaining percent [35].

A model is based on several assumptions, e.g.:

- that the joints are frictionless pin-joints;

- that the segments are rigid with mass concentrated at their centers of mass (COMs);
- that there is no co-contraction of agonist and antagonist muscles;
- that air friction is minimal.

Joint Reaction Forces

For the calculation of this type of forces, it is used the Newton equation; the sum of Horizontal Forces, $\sum F\text{x} = \text{m}\cdot\text{a}_x$ is calculated as [36]:

$$R_{xp} = m\cdot a_x - R_{xd}$$

where p = proximal, d = distal joint, a_y = acceleration of segment center of mass, COM, in y direction; note that d = force platform when p = ankle.

And the sum of Vertical Forces, $\sum F_y = m\cdot a_y$ as:

$$R_{yp} = m\cdot a_y + m\cdot g - R_{yd}$$

These formulas have to be considered for each body segment of the limb.

Joint moment

Finally, it could be possible to calculate the joint moments, the aim of inverse dynamics. This time, from Euler equation, the sum of Moments, $\sum M_z = I \cdot \alpha$, is calculated as [36]:

$$M_{zp} = I_z\cdot\alpha - M_{zd} - R_{xp}\cdot r_p\cdot\sin\theta + R_{yp}\cdot r_p\cdot\cos\theta + R_{xd}\cdot r_d\cdot\sin\theta - R_{yd}\cdot r_d\cdot\cos\theta$$

where r_p = distance from segment COM to proximal joint; θ = angle of segment to the right-hand horizontal. Or, using motion coordinates (more usual), the sum of moments is obtained with:

$$M_{zp} = I_z\cdot\alpha - M_{zd} - R_{xp}\cdot (y_p-y_{CoM}) + R_{yp}\cdot (x_{COM}-x_p) + R_{xd}\cdot (y_{COM}-y_d) - R_{yd}\cdot (x_d-x_{COM})$$

where (x_{COM}, y_{COM}) are the coordinates of the center of mass of the segment, (x_p, y_p) the coordinates of the proximal joint (x_d, y_d) the coordinates of the distal joint. For the understanding of these formulas, see the figure below.

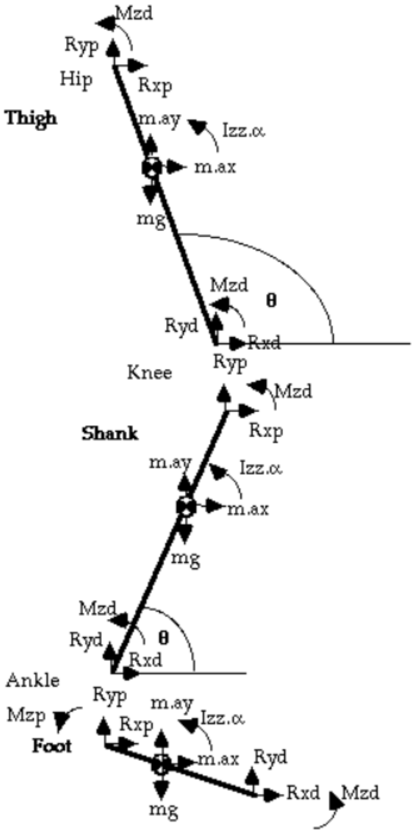


Figure 26: A limb-segment model with all the vectors cited in the formulas [36].

It will be noted that several body segment parameters are required to be known, such as the relative masses and mass moment of inertia (or radius of gyration) of each segment, the positions of their centers of mass. These are determined mostly from published cadaver studies, though there now exist automatic techniques for their estimation in vivo.

However, kinematics and forces are usually not consistent due to incorrect modeling assumptions and measurement errors. This is commonly resolved by introducing ‘residual forces and torques’ which compensate for this problem, but do not exist in reality.

Inverse dynamics is a very powerful technique for understanding movement, but it does have some inherent limitations [36]:

- it relies on assumptions that are not always valid - specifically:
    - there may be friction at the joint (e.g., in osteoarthritis);
    - the distribution of mass in the segment is not uniform, and certainly not concentrated at one point;
- estimating the joint center of rotation is prone to error (Holden & Stanhope, 1998):

  - the typical models (e.g., Helen Hayes) used rely heavily on anthropometry to define the hip joint center (because it is deep and so can't be directly defined by a marker);
  - the joint center of rotation may also (and often does) move during motion, especially at the knee;
  - some models (e.g., Cleveland model and six degree of freedom model used at NIH using marker triads) make less assumptions in this respect;
- measurement error (Holden et al, 1997 [37]):
  - the worst of these tends to be inaccuracies in co-alignment of the force platform and motion analysis system;
  - marker motion on the skin, especially "wand" type markers on sticks;
  - motion at the skin-bone interface;
  - marker tracking is sometimes contaminated by errors due to interpolation when markers go missing and data from some frames is lost;
- body segment parameters (anthropometry) are approximations and generalizations:
  - very thin or overweight people, children and patients with wasted legs may have different proportions;
  - note that this will mainly affect the swing phase; during stance the ground reaction forces are dominant, and accelerations are minimal;
  - special consideration must be given to amputees, in order to use values appropriate to the prosthesis components;
- error propagation (the errors of the distal joint calculations affect those at more proximal calculations).
- it can only determine the net moment:
  - co-contraction of antagonistic muscles will cancel out (important in spastic conditions, such as cerebral palsy and stroke);
- it cannot differentiate between different muscles:
  - e.g., we can determine that the joint moment is flexor, but not the relative activity of each flexor muscle;

3. CONTACT MODELS

3.1. Description

When two or more bodies collide, contact occurs between two surfaces of the bodies so that they cannot overlap in space. Contact mechanics is the study of the deformation of solids that touch each other at one or more points. In a broader sense, contact is a common and important aspect of mechanical systems, where multiple parts are assembled to compose the system. The objective of contact analysis is to investigate four different aspects:

1. Whether two or more bodies are in contact;
2. Where the location or region of contact is;
3. How much contact force or pressure occurs in the contact interface;
4. If there is a relative motion after contact in the interface.

Contact is categorized as nonlinearity [38]. This can be explained in two aspects. Firstly, if two separate bodies come into contact, the contact force stays at zero until the bodies are separate and increases vertically after the two bodies come into contact. A similar phenomenon happens in the tangential direction under friction, where two bodies are stuck together until the tangential force reaches a threshold, after which continuous sliding occurs without further increasing the tangential force. Such an abrupt change in contact force and slip makes the problem highly nonlinear. Secondly, in order to be a well-posed problem in mechanics, either displacement (kinematics) or force (kinetics), but not both, must be given for every material point. In fact, a central distinction in contact mechanics is between stresses acting perpendicular to the contacting bodies' surfaces (known as normal stress) and frictional stresses acting tangentially between the surfaces (shear stress).

Normal contact mechanics or frictionless contact mechanics focuses on normal stresses caused by applied normal forces and by the adhesion present on surfaces in close contact, even if they are clean and dry. Frictional contact mechanics emphasizes the effect of friction forces [39].

There are different types of contact body models:

- Elastic
- Plastic
- Visco-Elastic

In static or dynamic contact.

In order to describe contact mechanics, contact force formulations like the Hertz theory and elastic foundation modeling (EFM) have been used [40].

Hertzian contact theory is a classical theory of contact mechanics and even though the derivation of this theory is relatively difficult, the final solution is a set of simple analytical equations relating the properties of the system to the developed stress. Hertz contact theory is derived from the analytical solution of elasticity theory equation under half-space (i.e., either of the two parts into which a plane divides the three-dimensional Euclidean space) approximation [41]:

1. Surfaces are infinitely large half-spaces.
2. Pressure profile is parabolic (which assumes that the shape of the bodies in contact can also be approximated well with parabolic shapes, e.g., sphere, ellipse, or a cylinder)
3. All the assumptions of the classical theory of elasticity apply (small strain, homogeneous material).

Elastic Foundation Modeling (EFM) is a technique used to analyze the behavior of structures resting on elastic or viscoelastic foundations. The technique involves developing a mathematical model for the foundations of a system and considering the interaction between the foundation and the system. These models range from simple to complex and have been applied in structural mechanics, bio-devices, and composite structures. The Winkler model is one of the simplest presentations for an elastic foundation. It was first proposed by Winkler to model a Euler–Bernoulli beam on an elastic foundation. The model assumes that the foundation behaves like a series of independent linear springs supporting the structure [42].

The analytical method described by Hertz assumes that the size of the contact area is small when compared to the curvature radii of the surfaces in the initial contact point. It is a good approximation for small strains and non-conforming solids but is limited to simple geometric objects like spheres or cylinders. Alternatively, EFM assumes that contacting solids may be considered rigid bodies but with a thin layer of elastic material at the surfaces. The geometry of each surface, which can be arbitrarily complex, is approximated with a triangular mesh. A spring is placed at the centroid of each triangle, discretizing the contact areas, and enabling the computation of contact forces for each discrete element, independent of each other [40].

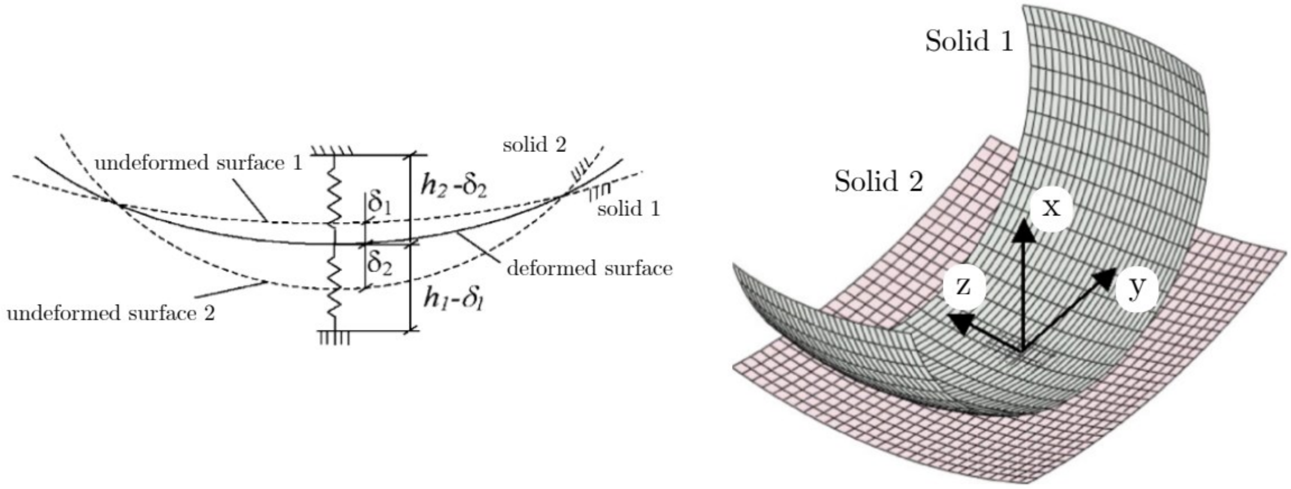


Figure 27: Elastic Foundation method formulation [40].

Based on Hertz elastic contact theory, Hunt and Crossley [43] developed a contact force model in 1975 very used in a wide series of applications through the years. The popularity of the Hunt and Crossley contact force model is owed to its simplicity and suitability to treat impacts in dynamical systems, since it directly depends on the material, geometric and kinematic properties of the colliding bodies. Moreover, this model avoids the computation of significant contact forces, either repulsive or attractive, when the penetration is close to zero [44]. However, it can be a source of computational difficulties, such as the generation of attractive forces during the impact period, which are not acceptable from the physical point of view. This occurs when the contact forces are not the only ones governing the motion of the colliding bodies. In addition, the Hunt and Crossley contact force model does not behave properly for collisions with high impact velocity. In fact, as mentioned in the original paper, this model was developed for low impact velocities. Another limitation associated with the Hunt and Crossley solution deals with the estimation and identification of the contact parameters.

3.2. Contact spheres models

3.2.1. Impact between a sphere and a half-space

An elastic sphere of radius R incidents an elastic half-space understood as ground, where total deformation is d, causing area a of radius [39]:

$$\mathrm{a} = \sqrt{r \cdot d}$$

The applied force F is related to the displacement d by:

$$F = \frac{4}{3} E\, r^{\frac{1}{2}} \cdot d^{\frac{3}{2}}$$

Where E comes from:

$$\frac{1}{E} = \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2}$$

And E_1, E_2 are the elastic moduli ν_1, ν_2 the Poisson's ratios associated with each body. The distribution of normal pressure in the contact area as a function of distance from the center of the circle is

$$p(r) \ = \ p_0(1-\frac{r^2}{a^2})^{\frac{1}{2}}$$

where p_0 is the maximum contact pressure given by

$$p_0 \ = \ \frac{3F}{2\pi a^2} = \frac{1}{\pi}(\frac{6FE^2}{r^2})^{\frac{1}{3}}$$

The radius of the circle is related to the applied load F by the equation

$$a^3 = \frac{3Fr}{4E}$$

Thus, the total deformation d is related to the maximum contact pressure by

$$d \ = \ \frac{a^2}{r} = (\frac{9F^2}{16E^2r})^{\frac{1}{3}}$$

The maximum shear stress occurs in the interior at 0.49a for ν = 0.33.

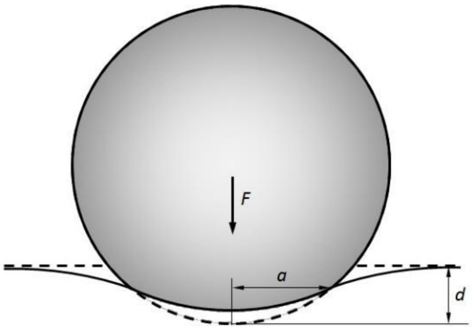


Figure 28: Contact between a sphere and a half-space [39].

3.2.2. Impact between two spheres

For this purpose, let consider the representation of [Fig.29], in which the two spheres 1 and 2 have masses *m₁* and *m₂*, respectively. In order to keep the analysis simple, let assume that the two spheres describe a one-dimensional movement with constant velocities and without any external applied force. Thus, before the impact, the velocity of sphere 1 is higher than the velocity of sphere 2, which means that sphere 1 impacts sphere 2. After the termination of the impact, the velocity of sphere 1 is lower than the velocity of sphere 2, which implies that the two spheres separate from each other when the impact ends. During the impact process, the transition between pre- and post-impact velocities is considered to be smooth. Furthermore, the duration of the impact is assumed to be finite to facilitate the representation of the kinematic and dynamic quantities [44].

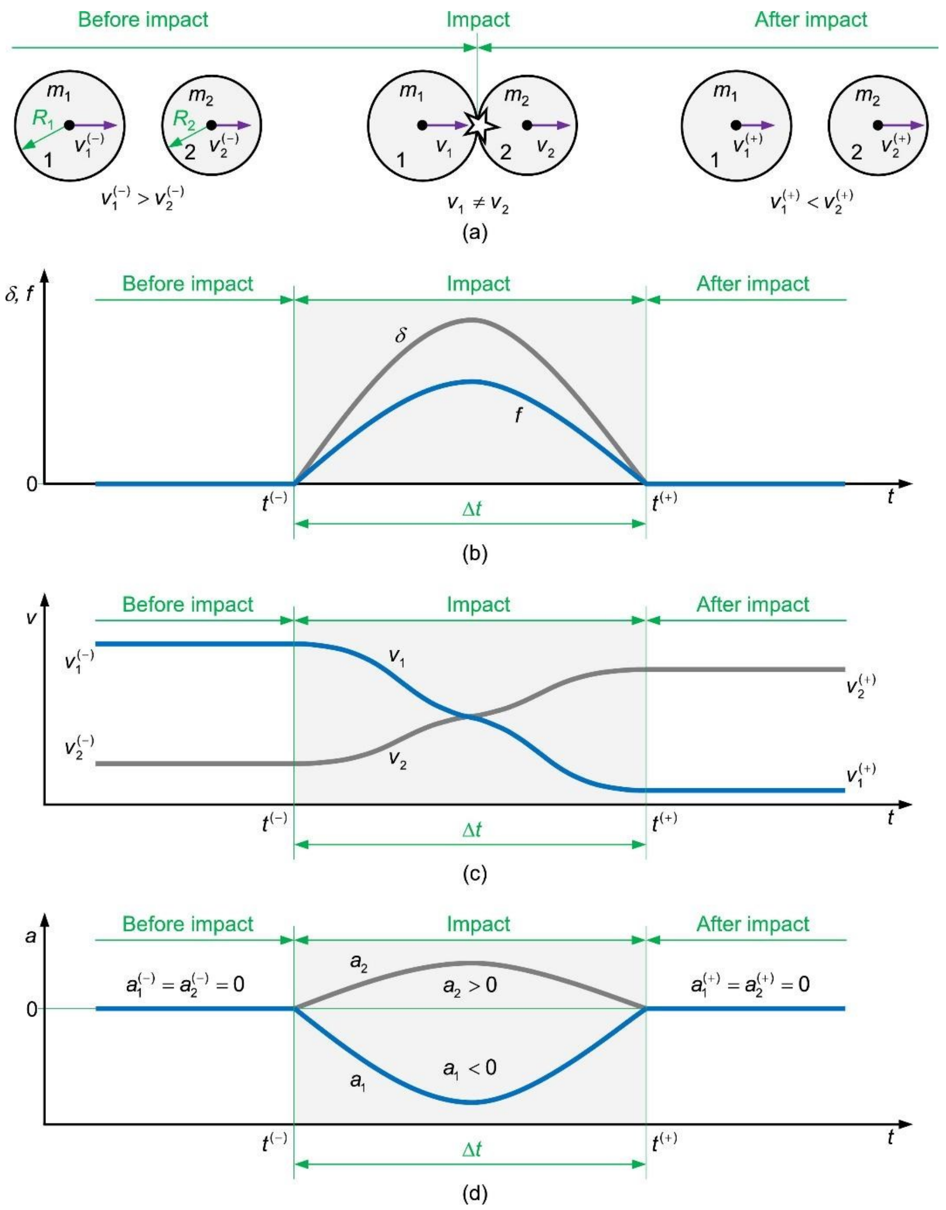


Figure 29: a) One-dimensional central impact between two solid spheres; b) Deformation and contact force evolutions during the impact; c) Velocities of the spheres before, during, and after the impact; d) Accelerations of the spheres before, during, and after the impact [44].

When the two spheres, represented in [Fig.29], are in contact, deformation (penetration or indentation) occurs in the vicinity of the contact zone, resulting in reaction contact forces that act over the impact period. The geometric condition that permits evaluating the relative penetration between the two spheres can be expressed as [44]

$$\delta = R_1 + R_2 - (x_2 - x_1)$$

where x_1 and x_2 represent the horizontal position of the spheres 1 and 2, and R_1 and R_2 denote the corresponding radii.

One of the most popular contact force models developed to handle the impact between two solid elastic and isotropic spheres is the one based on the hertzian contact theory. The Hertz's contact force model can be expressed as

$$F = K\delta^n$$

in which K represents the contact stiffness parameter, δ is the local relative penetration between the surfaces of the two spheres, and n is the nonlinear power exponent determined from material and geometric properties of the local region of the contacting bodies. According to the hertzian contact theory, the power exponent n is equal to 3/2 for the case in which a parabolic distribution of contact stresses is considered. For two spheres of isotropic materials in contact, the contact stiffness parameter is a function of the radii of the spheres 1 and 2 and the material properties in the form

$$K = \frac{4}{3(\sigma_1 + \sigma_2)}\sqrt{\frac{R_1 R_2}{R_1 + R_2}}$$

where the material parameters σ_1 and σ_2 are given by

$$\sigma_i = \frac{1-v_i^2}{E_i} \; (i = 1{,}2)$$

and the quantities v_i and E_i represent the Poisson's ratio and Young's modulus associated with each sphere, respectively. For other contacting geometries, the contact stiffness is evaluated using different expressions, which always depend on the local geometry and material properties [44].

[Fig.27 b-c-d], show the evolution of the deformation and contact force at the impact duration, as well as the velocities and accelerations of each sphere before, during and after the collision. In these diagrams, $t^{(-)}$ represents the instant just before the impact, $t^{(+)}$ denotes the instant immediately after the impact, and Δt is the duration of the impact. In the impact process represented in [Fig.27], the relative approaching velocity, and the relative separating velocity are defined as, respectively

$$\dot{\delta}^{(-)} = v_1^{(-)} - v_2^{(-)} \; \& \; \dot{\delta}^{(+)} = v_1^{(+)} - v_2^{(+)}$$

The kinematic coefficient of restitution, which is based on Newton's impact theory, can be established as the quotient between the relative normal velocities of the colliding bodies just after and just before the impact. The Newton's hypothesis can be written as

$$c_r = -\frac{\dot{\delta}^{(+)}}{\dot{\delta}^{(-)}}$$

where $\dot{\delta}^{(-)}$ and $\dot{\delta}^{(+)}$ represent the relative normal velocities of the colliding bodies immediately before and after the impact, respectively. The coefficient of restitution constitutes the foundation of the impact models of mechanical systems. For a fully elastic collision, this parameter is equal to unity, while for a fully inelastic collision, the coefficient of restitution is null. The most general and predominant type of collision involves a coefficient of restitution, the value of which varies between 0 and 1 [44].

The Hunt and Crossley [43] contact force model is composed of two parcels, namely the elastic (conservative) and viscous (dissipative) components, and can be written as

$$F = K\delta^n + \chi\delta^n\dot{\delta}$$

where *K* represents the contact stiffness, δ is the penetration, *n* denotes the nonlinear exponent, χ is the hysteresis damping factor, and $\dot{\delta}$ is the penetration velocity. According to the Hunt and Crossley approach, the coefficient of restitution can be related to the impact velocity as

$$c_r = 1 - \alpha\dot{\delta}^{(-)}$$

in which α represents a parameter that depends on the material, geometry, and kinematics of contacting bodies, and $\dot{\delta}^{(-)}$ represents the velocity at the beginning of the contact. It should be noticed that, according to Hunt and Crossley work, the relation given by the last equation is valid for relatively low impact velocities. Hunt and Crossley also presented a formulation for the hysteresis damping factor as

$$\chi = \frac{3}{2}\alpha K$$

Thus, combining last two equations yields the following expression for the hysteresis damping factor

$$\chi = \frac{3(1-c_r)}{2}\frac{K}{\dot{\delta}^{(-)}}$$

which can easily be determined since the values of the coefficient of restitution and contact stiffness parameter are characteristics of the pair of contacting bodies. Finally, introducing the last expression of hysteresis damping factor into the equation of contact force, permits obtaining the Hunt and Crossley contact force model written in a convenient form as

$$F = K\delta^{n}[1 + \frac{3(1-c_r)}{2}\frac{\dot{\delta}}{\dot{\delta}^{(-)}}]$$

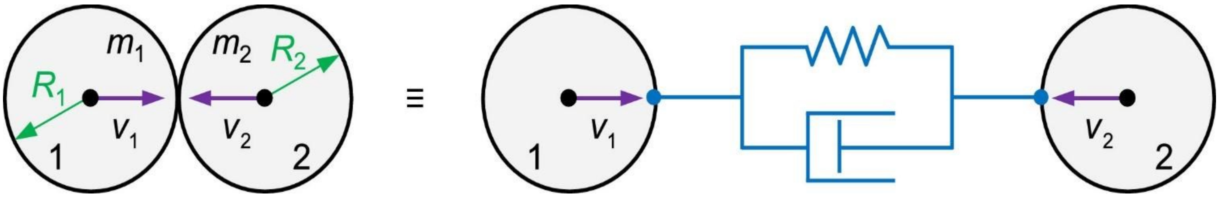


Figure 30: Central impact between two solid spheres modeled with the Hunt and Crossley contact force model [44].

3.3. Musculoskeletal models application

Musculoskeletal models can be divided into two main categories: finite elements (FE) models and multibody models [40].

3.3.1. Finite elements (FE) musculoskeletal models

Finite Element Analysis, or FEA, is the simulation of a physical phenomenon using a numerical mathematical technique called the Finite Element Method, or FEM, useful for problems with complicated geometries, loadings, and material properties where analytical solutions cannot be obtained [45]. The FEM is a general numerical method for solving partial differential equations in two or three space variables (i.e., approximate solution of boundary value problems). To solve a problem, the FEM subdivides a large system into smaller, simpler parts that are called finite elements. This is achieved by a particular space discretization in the space dimensions, which is implemented by the construction of a mesh of the object: the numerical domain for the solution, which has a finite number of points. The finite element method formulation of a boundary value problem finally results in a system of algebraic equations. The method approximates the unknown function over the domain. The simple equations that model these finite elements are then assembled into a larger system of equations that models the entire problem. The FEM then approximates a solution by minimizing an associated error function via the calculus of variations.

There have been some FE MSK models, especially models of the foot. FE analyses can predict the load distribution between the foot and supporting structures and provide information on the internal stress and strain states of the ankle-foot complex [40]. Simple foot FE models have been developed to study the stress distribution within the foot at different time points during stance phase [46] or to evaluate stress distributions in the plantar soft tissue [47]. These models described the foot as a unique segment and often used rough geometries of the bones. Other simple models were used to determine the stress distribution in the foot plantar soft tissue when using insoles. FE analysis has provided valid insights into different aspects of lower limb musculoskeletal biomechanics, e.g., foot joints stress distribution and plantar distribution on the foot soft tissue. However, FE musculoskeletal studies are limited to static or quasi-static evaluations due to the required demanding computational power.

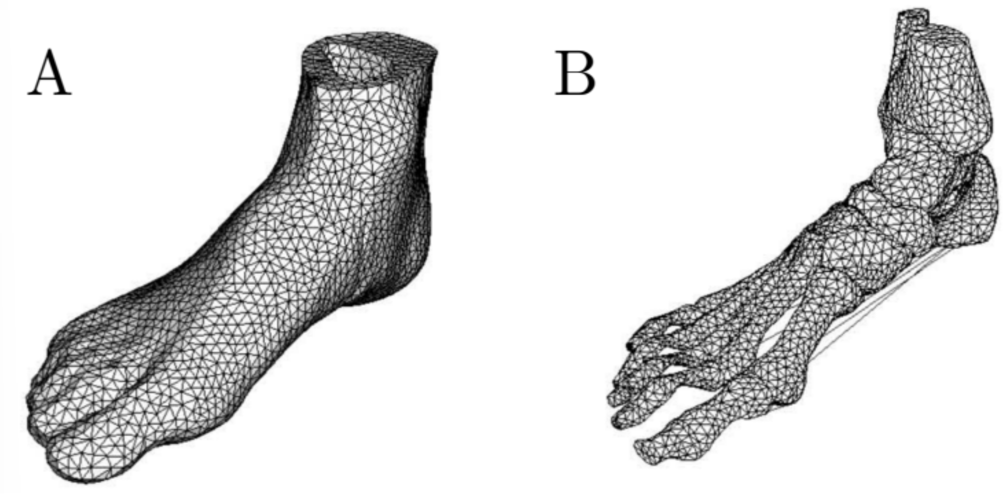


Figure 31: Finite Element (FE) model described in Cheung and Zhang (2005): (A) soft tissues; (B) bony and ligamentous structures [40].

When the physical phenomena is a contact between two or more bodies, the finite element equation solves for unknown information with given information. If the boundary value problem is, for example, the displacement of the bodies' boundary, and the displacement is given, reaction force should be calculated with FEM; on the other hand, on the traction boundary, if the applied force is given, the corresponding displacement is to be calculated. In case of contact, however, both displacement and contact force are unknown, except for very limited cases; that is, the contact boundary is a part of the solution. Therefore, the finite element analysis procedure must find if a material point in the boundary of a body is in contact with the other body and if it is in contact, the corresponding contact force must be calculated. Since the contact force at a material point can affect the deformation of neighboring points, this process needs to be repeated until finding the right states for all points that are possible in contact. For the case of an elastic system, equilibrium can be described as finding a displacement field that minimizes the potential energy. Contact can then be considered as a constraint of the optimization formulation, such that the potential energy is minimized while satisfying the contact constraint; that is, a body cannot penetrate the other body. The constrained optimization problem can be converted into an unconstrained one by using the penalty regularization or Lagrange multiplier methods. Therefore, most contact algorithms are derived based on these two methods.

To run an FEA simulation, a mesh is first generated that makes the object of interest discrete, containing millions of small elements that make up the overall shape. This is a way of transcribing a 3D object into a series of mathematical points that can then be analyzed. The density of this mesh can be altered based on how complex or simple a simulation is needed. Calculations are run for every element or point of the mesh and then combined to make up the overall final result for the structure. Since the calculations are done on a mesh rather than the entirety of a physical object, some interpolation needs to occur between the points. These approximations are usually within the bounds of what's needed. The points of the mesh where the data is known mathematically are referred to as nodal points and tend to be grouped around boundaries or other areas of change in an object's design [45].

A mesh is a representation of a larger geometric domain by smaller discrete cells. A mesh partitions space into elements (or cells or zones) over which the equations can be solved, which then approximates the solution over the larger domain. Element boundaries may be constrained to lie on internal or external boundaries within a model. Higher-quality (better-shaped) elements have better numerical properties, where what constitutes a "better" element depends on the general governing equations and the particular solution to the model instance.

The primary characteristics of a finite element are embodied in the element stiffness matrix. For a structural finite element, the stiffness matrix contains the geometric and material behavior information that indicates the resistance of the element to deformation when subjected to loading. Such deformation may include axial, bending, shear, and torsional effects.

In finite element analysis, solution accuracy is judged in terms of convergence as the element “mesh” is refined. There are two major methods of mesh refinement. In the first, known as h-refinement, mesh refinement refers to the process of increasing the number of elements used to model a given domain, consequently, reducing individual element size. In the second method, p-refinement, element size is unchanged, but the order of the polynomials used as interpolation functions is increased. The objective of mesh refinement in either method is to obtain sequential solutions that exhibit asymptotic convergence to values representing the exact solution.

3.3.1.1. Spheres' mesh

Most of the geometric objects like points, lines, triangles, planes, pyramids, and cubes, are perfectly representable in modern graphics hardware. When users face the problem of creating a mesh for a sphere, trade-offs must be made between quality and construction, memory and rendering costs. There are several methods to do this type of mesh, but there will be explained four methods, analyzing their characteristics and comparing them (O. S. Cajaraville, 2015 [48])

- Standard sphere – UV Sphere: This is the most common implementation of a sphere mesh and can be found in almost any 3D toolset. This method divides the sphere using meridians (lines from pole to pole) and parallels (lines parallel to the equator). It produces faces with a bigger area near the equator and smaller ones close to the poles. The faces are of triangles, at the poles, and quads.
- Normalized Cube: This method uses a uniformly subdivided cube where each vertex position is normalized and multiplied by the sphere radius. This creates a non-uniformly subdivided sphere where the triangles closer to the center of a cube face are bigger than the ones closer to the edges of the cube.
- Spherified Cube: This method is based on a subdivided cube as well, but it tries to create more uniform divisions in the sphere. The area of the faces and the length of the edges suffer less variation, but the sphere still has some obvious deformation as points get closer to the corners of the original cube.
- Icosahedron: An icosahedron is a polyhedron composed of 20 identical equilateral triangles and possesses some interesting properties: Each triangle has the same area, and each vertex is at the same distance from all its neighbors. To get a higher number of triangles there is the need to subdivide each triangle into four triangles by creating a new vertex at the middle point of each edge which is then normalized, to make it lie in the sphere surface. This breaks the initial properties of the icosahedron; the triangles are not equilateral anymore and neither the area nor the distance between adjacent vertices is the same across the mesh. An added problem with this method is that it can only increase the number of faces by four each time. But it is still a better approximation by almost any measure excluding its number of triangles.

The sphere mesh will be an imperfect representation as it is using triangles to approximate the surface of the sphere. The metrics used to compare the different implementations is the distance from a number of points on the sphere surface to the mesh created, and the ratio between the expected triangle area and the actual triangle area. It turned out that the UV sphere is the worst option in terms of accuracy for a given number of triangles, but it is still the easiest algorithm. But if there is the need to render a lot of spheres or there is the need of highly tessellated spheres, the other algorithms are better [48].

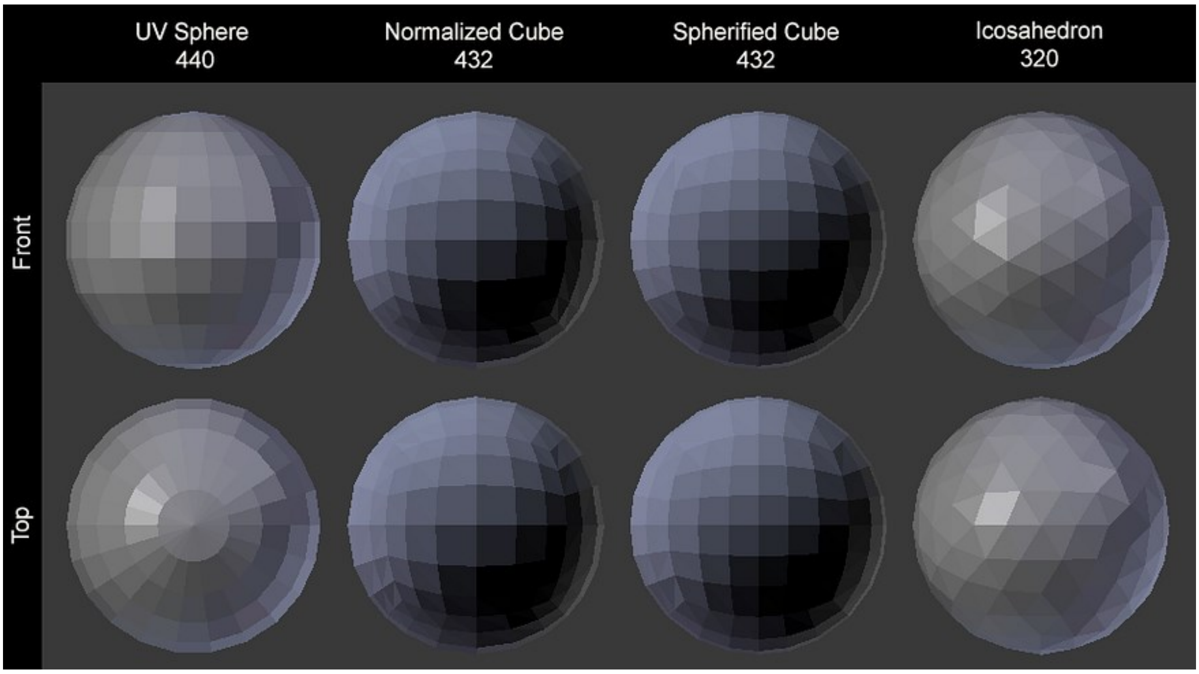


Figure 32: Comparison of the four methods to mesh a sphere illustrated. All around 400 triangles [48].

3.3.2. *Musculoskeletal multibody models*

Musculoskeletal multibody models are described by rigid segments (bones) connected through joints actuated by muscle tendon actuators [40]. A rigid body is a system of particles, where the distance between any two particles is fixed, also when it is in motion, and this distance is described by a shift vector that is in function only of the coordinates of the two particles, considered as points. In fact, all the particles of which the rigid body is constituted have a rigid motion, that is, they maintain the features, the distances, and a fixed respect to an external observer [49].

With an observer is intended an orthogonal cartesian system with a fixed orientation in which are defined an origin, a length measurement unit and three orthogonal axes i.e., x, y, z. The problem of rigid motion could be reconducted to the relative motion of two referral systems in the three-dimensional Euclidean space, that is the local system associated with the rigid body that moves with respect to the environment (for example the laboratory), represented by the global referral system. To define the position of a rigid local system with respect to a rigid global system, it is used a position vector and an orientation matrix (director cosines matrix). The fundamental formula of rigid motion, in vectorial form is:

$$\dot{\mathbf{P}} = \dot{\mathbf{O}} + \underline{\omega} \wedge (\mathbf{P} - \mathbf{O})$$

Where $\dot{\mathbf{P}}$ is the vector of coordinates of the velocity in the global system of the rigid body point; $\dot{\mathbf{O}}$ is the vector of coordinates of the velocity of the origin of the global system; $(\mathbf{P} - \mathbf{O})$ is equal to the product of position vector in the local system and the orientation matrix; $\underline{\omega}$ is the angular velocity. The rigid body has six degrees of freedom according to Euler-Cardan convention: 3DOF refers to the three rotations around the three axes and the other 3 are referred to the three shifts along the three distal axes. The two researchers, Grood and Suntay, defined a joint coordinate system that provided a simple geometric description of the three-dimensional rotational and translational motion between two rigid bodies.

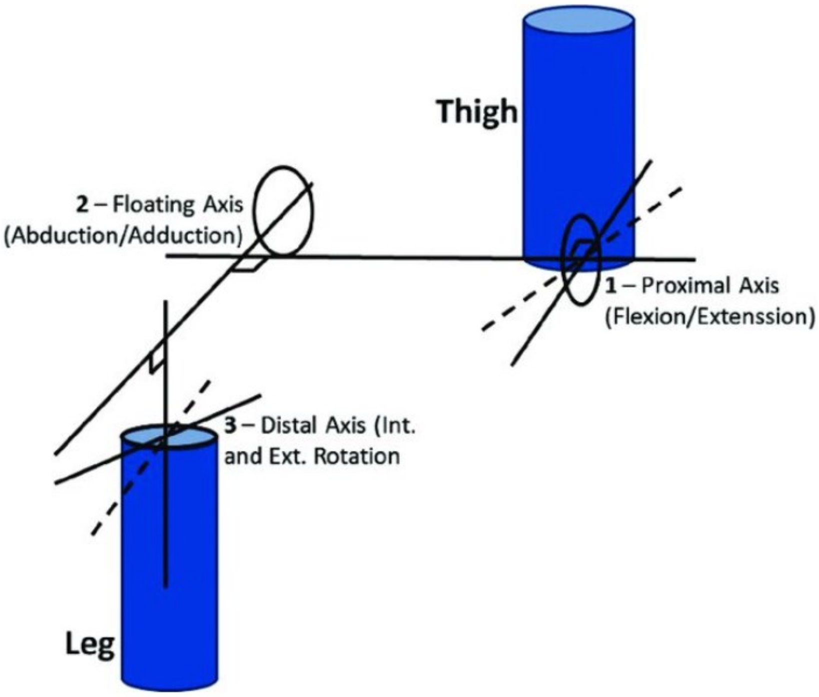


Figure 33: Joint coordinate system with floating axis applied to the knee by Grood and Suntay (1983). [50].

It is assumed that internal forces are generated to hold the relative positions fixed. These internal forces are all balanced out with Newton’s third law, so that they all cancel out and have no effect on the total momentum or angular momentum. The rigid body can actually have an infinite number of particles, spread out over a finite volume. The properties of shape, total mass, center of mass and rotational inertia tensor, for a rigid body, remain constant. The shape is defined by a closed triangular mesh of integer vertex indices.

3.3.2.1. AnyBody

The AnyBody modeling system (AMS) is a state-of-the-art musculoskeletal modeling and simulation software for biomechanical analysis founded by researchers from Aalborg University [51]. The AnyBody modeling allows users to create a full body detailed MSK model and simulate internal body loads as e.g., muscle activity, muscle forces and joint reaction forces. AMS has a graphical user interface (GUI) including several features such as the AnyScript editor, progress, and information window, and the model and chart view. AnyScript is an object-oriented programming language used within the AnyBody Modeling System. It is specially designed for describing the workings of the human body and its interactions with environmental objects such as exoskeletons, workplaces, etc. The elements of the models in AMS are mostly mechanical in nature and the most important ones are:

- Segments: representation of bones and other rigid elements of models.
- Joints: They connect segments and allow them to articulate with respect to each other.
- Drivers: Specification of the movement of the model should perform and provide power input as motors.
- Kinematics measures: Abstraction representation of kinematical constraints.
- Forces: Forces applied to the model.

AMS provides the user of an embedded chart view in which he/she could review analysis results e.g., individual muscle activity and joint reaction forces and get a graphical representation of the same. Moreover, AMS handles closed kinematic chains and it comes with inbuilt solvers for computing of forces and moments based on kinematics of the human body. AMS has the ability to predict ground reaction forces (GRF) so the user can make inverse dynamics models based on recorded motion without GRF force measurement.

A use of this server was for example, those of Kim, et al. (2019) [52]. The authors developed a five-segment musculoskeletal foot model by modifying the detailed foot model in the AnyBody Managed Modeling Repository and integrated it with a full-body musculoskeletal model for quantifying moments in foot joints. This five-segment foot model consists of five rigid segments (talus, calcaneus, midfoot, metatarsals, and toes) and five joints (tibiotalar, subtalar, Chopart's, Lisfranc, and MTP joints). The foot model has 9 rotational degrees of freedom

(DoFs), 21 muscles and 17 ligaments. The kinematics of the joints were estimated in the kinematics optimization routine in the AnyBody Modeling System. Scales of the five foot-segments and marker positions were adjusted to minimize the difference between the marker locations on the model. The GRF measured at the center of the foot pressure was divided by the ratio of the sum of the pressure under each foot segment. To input distributed GRFs into the foot segments, multiple nodes were attached to each foot segment (calcaneus: 31, midfoot: 36, metatarsals: 73, and toes: 30), and the total number of nodes was 170 as it can be seen in the figure [Fig.32]. The nodes in a segment could provide the GRF that was distributed to the segment using foot pressure data. Joint kinematics and distributed GRFs were input to the full-body musculoskeletal model with the authors' custom foot. An inverse dynamics-based optimization in the AnyBody Modeling System was performed at each instance of the input data to estimate the kinetics of the human musculoskeletal system, which included forces of the muscles and ligaments. The optimization minimized the maximum activation levels (force divided by strength) of the muscle actuators. The force between two adjacent bones was calculated at each joint during the kinetics estimation.

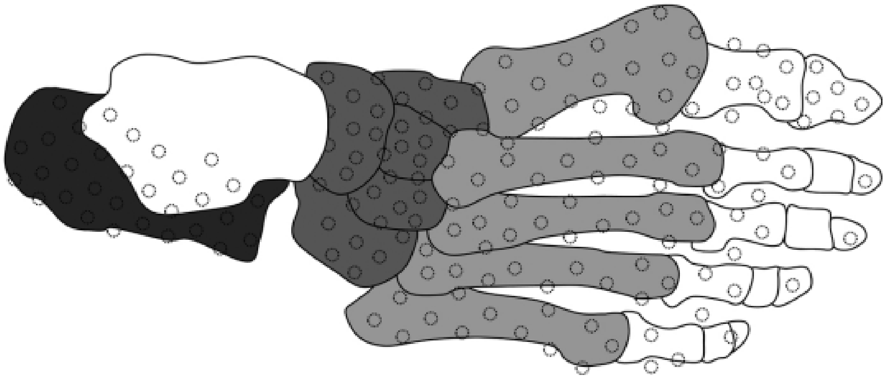


Figure 34: The 170 nodes in a five-segment MSK foot model that provide the ground reaction force (GRF) were distributed to this segment [52].

3.3.2.2. OpenSim

OpenSim is a freely available software package that enables the user to build, exchange, and analyze computer models of the musculoskeletal system and dynamic simulations of movement. OpenSim was introduced at the American Society of Biomechanics Conference in 2007. Successively, an application programming interface (API) was added, allowing researchers to access and customize OpenSim core functionality. In the last versions of the software, the API was extended to MATLAB and Python and a new visualizer was added. Since the initial release, thousands of people have used the software in a wide variety of applications, including biomechanics research, medical device design, orthopedics and rehabilitation science, neuroscience research, ergonomic analysis and design, sports science, computer animation, robotics research, biology, and education [53].

The software provides a platform on which the biomechanics community can build a library of simulations that can be exchanged, tested, analyzed, and improved through multi-institutional collaboration. The core software is written in C++, and the graphical user interface (GUI) is written in Java. OpenSim plug-in technology makes it possible to develop customized controllers, analyses, contact models, and muscle models among other things. The user can analyze existing models and simulations and develop new models and simulations from within the GUI. Moreover, the user with the GUI can import motion analysis data, scale a computer model of the musculoskeletal system, perform inverse dynamics analysis, and plot results all from the graphical interface.

OpenSim includes a wide variety of features. Some of the most useful features include taking pictures and videos of MSK models, plotting the results of the analysis; scaling the size of MSK models, performing inverse kinematics (IK) and dynamics (ID) but also forward dynamics and analyzing dynamic simulations.

In formulating the equations-of-motion (i.e., the system dynamics), OpenSim employs Simbody which is an open-source multibody dynamics solver. In Simbody and OpenSim, the body is the primary building block of the model. Each body in turn owns a joint that connects it to an existing parent body. The joint defines the coordinates and kinematic transforms that govern the motion of that body with respect to its parent body. Within the model all bodies are contained in a BodySet [53].

Thus, to start the model, there is the need to define a set of rigid bodies that represent the system. In the <BodySet> section, there are defined this group of bodies, with the name, mass properties, and visible objects associated with each body.

For Geometries, OpenSim supports more types/shapes such as Brick, Sphere, Cylinder, Cone, and Ellipsoid. In the model file it can be specified not only the shape geometry, but also the color, display preferences, and scale factors. These properties that specify how the geometry is displayed are grouped under the <Appearance> tag and can be specified for each Geometry object separately.

In addition to the set of rigid bodies, there is also the need to define the relationship between those bodies (i.e., joint definitions). In the figure below [Fig.], a joint (in red) defines the kinematic relationship between two frames (B and P) each affixed to a rigid body (the parent, P_o, and the body being added, B_o) parameterized by joint coordinates

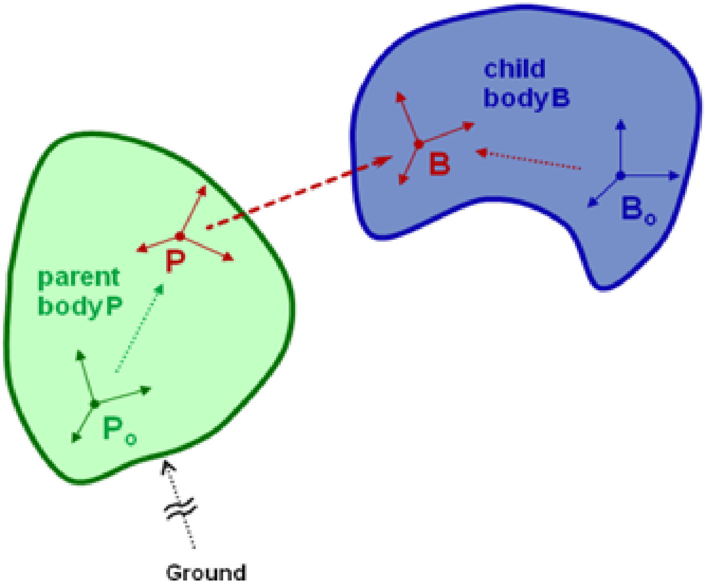


Figure 35: Parent (green) and child (blue) bodies kept together by the joint (red) [53].

A body is a moving reference frame (B_o) in which its center-of-mass and inertia are defined, and the location of a joint frame (B) fixed to the body can be specified. Similarly, the joint frame (P) in the parent body frame (P_o) can also be specified. Flexibility in specifying the joint is achieved by permitting joint frames that are not coincident with the body frame. In the last version of OpenSim, this flexibility was enhanced via the introduction of the Frame class hierarchy. There are three main types of Frames:

- Ground (each model starts with a ground frame)
- Body
- PhysicalOffsetFrame

All three of these frames are called PhysicalFrames because they either are a rigid body or are fixed to a rigid body. In this OpenSim version, a joint connects two PhysicalFrames (parent and child).

In order to actuate the model, there is the need to define the *forces* that will be applied to the model. Just like bodies are defined within the <BodySet> section, forces are defined in the <ForceSet> section of the model file. Forces come in two varieties:

- *passive* forces like springs, dampers, and contact;
- *active* forces like springs, idealized linear or torque actuators, and muscles;

Active forces that require input (controls) supplied by the user or by a controller are called Actuators and are a subset of the ForceSet. OpenSim has several built-in forces that include: PrescribedForce, SpringGeneralizedForce, BushingForce, as well as HuntCrossleyForce and ElasticFoundationForce to model forces due to contact.

A model may have some specific contact geometry that is associated with a model, in fact they are required when defined contact forces (for example, Hunt-Crossley Force) are. In OpenSim, contact geometry can be an analytical shape, such as a half-place, sphere, or cube, or a user-defined shape represented in a geometry file. The Sphere class reference is a contact geometry subclass representing a sphere centered at the origin, in which it could be defined as the radius besides the other features of contact geometry set class, i.e., location, orientation and/or filename [53].

3.3.2.3. OpenCap

Due to costs and the requirement of experts, despite the utility of the instruments of the optical motion capture, metrics of movement dynamics are not often used in clinical trials. For this reason, visual movement evaluations are used, that require basic instruments and there is no need for the laboratory environment to inform clinical decisions and as outcomes for clinical trials. With this aim, it is developing a new open-source platform, called OpenCap, that from video-based motion measures three-dimensional kinematics and kinetics of human movement with two or more smartphones and so without specialized hardware, software, or expertise [54]. OpenCap provides the use of two iOS devices (iPhone or iPad), as substitution of the video cameras that are the gold standard optical motion capture, to register the videos of the human movement. This approach offers alternatives to motion capture to estimate joint kinematics, but it offers no ready way to estimate ground reaction forces and moments, which are essential components for MSK modeling.

As [55] article reported, there are some alternatives to force plates. For example, there are the instrumented shoes and pressure insoles that again are expensive and not perfectly accurate or there are methods such as foot ground non-contact models, like the calculation of the distance between the feet and the center of pressure (COP) for the estimation of GRFM, but these methods do not allow the calculation of all components. Foot ground contact models that use visco-elastic force elements or muscle-like force elements have a high accuracy when they are subject-specific, so these methods estimate all components of GRFM using kinematic data.

3.4. Machine learning

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.
In general, machine learning algorithms are used to make predictions or classification. Based on some input data, which can be labeled or unlabeled, the algorithm will produce an estimate about a pattern in the data. There is an error function that evaluates the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model [56].

About Model Optimization Process, if the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this "evaluate and optimize" process, updating weights autonomously until a threshold of accuracy has been met. Machine learning models fall into three primary categories.

- Supervised machine learning

Supervised learning, also known as supervised machine learning, is defined by its use of labeled datasets to train algorithms to classify data (classification supervised learning) or predict outcomes (regression supervised learning) accurately. As input data is fed into the model, the model adjusts its weights until it has been fitted appropriately. This occurs as part of the cross-validation process to ensure that the model avoids overfitting or underfitting. Supervised learning helps organizations solve a variety of real-world problems. Supervised learning methods are for example Bayesian classifiers for classification; support vector machines (SVM), nearest neighbor (NN), Random forest, Ada-boost, and artificial neural networks (ANN) for regression.

- Unsupervised machine learning

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. This method's ability to discover similarities and differences in information makes it ideal for exploratory data analysis, cross-selling strategies, customer segmentation, and image and pattern recognition. It's also used to reduce the number of features in a model through the process of dimensionality reduction.

Principal component analysis (PCA) and singular value decomposition (SVD) are two common approaches for this. Other algorithms used in unsupervised learning include gaussian mixture models, k-means clustering, and probabilistic clustering methods.

- Semi-supervised learning

Semi-supervised learning offers a happy medium between supervised and unsupervised learning. During training, it uses a smaller labeled data set to guide classification and feature extraction from a larger, unlabeled data set. Semi-supervised learning can solve the problem of not having enough labeled data for a supervised learning algorithm. It also helps if it's too costly to label enough data [56].

3.4.1. Artificial Neural Network (ANN)

Artificial neural networks (or simply neural networks (NNs)) are a branch of unsupervised machine learning models that are built using principles of neuronal organization discovered by connectionism in the biological neural networks constituting animal brains [57].

ANNs are composed of artificial neurons which are conceptually derived from biological neurons. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron receives signals then processes them and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Each artificial neuron has inputs and produces a single output which can be sent to multiple other neurons. The inputs can be the feature values of a sample of external data, such as images or documents, or they can be the outputs of other neurons. The outputs of the final output neurons of the neural net accomplish the task, such as recognizing an object in an image.

To find the output of the neuron we take the weighted sum of all the inputs, weighted by the weights of the connections from the inputs to the neuron. It has been added a bias term to this sum. This weighted sum is sometimes called activation. This weighted sum is then passed through a (usually nonlinear) activation function to produce the output. The neurons are typically organized into multiple layers. Neurons of one layer connect only to neurons of the immediately preceding and immediately following layers. The layer that receives external data is the input layer. The layer that produces the ultimate result is the output layer. In between them are zero or more hidden layers. Single layer and unlayered networks are also used. Between two layers, multiple connection patterns are possible. They can be 'fully connected', with every neuron in one layer connecting to every neuron in the next layer. They can be pooling, where a group of neurons in one layer connects to a single neuron in the next layer, thereby reducing the number of neurons in that layer. Neurons with only such are known as feedforward networks. Alternatively, networks that allow connections between neurons in the same or previous layers are known as recurrent networks [57].

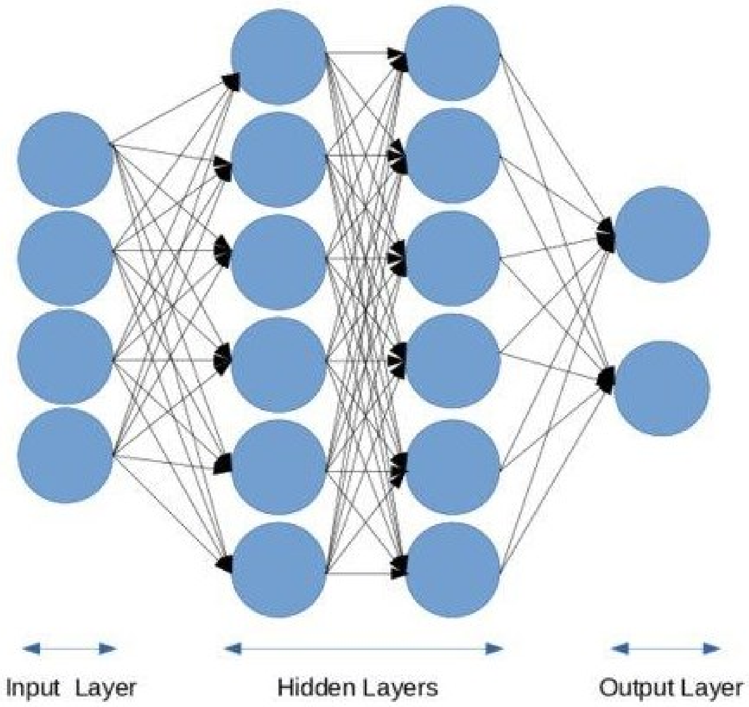


Figure 36: Example of Artificial neural network (ANN) architecture [58].

Artificial neural networks learn (or are trained) by processing examples, each of which contains a known "input" and "result", forming probability-weighted associations between the two, which are stored within the data structure of the net itself. The training of an artificial neural network from a given example is usually conducted by determining the difference between the processed output of the network (often a prediction) and a target output. This difference is the error. The network then adjusts its weighted associations according to a learning rule and using this error value. Successive adjustments will cause the neural network to produce output that is increasingly similar to the target output. After a sufficient number of these adjustments, the training can be terminated based on certain criteria. Then are going to add more examples, so artificial neural networks systems learn to perform any tasks without being programmed with task-specific rules [57].

ANNs are an extraordinarily flexible tool for nonlinear modeling and are especially useful in gait analysis. Many studies have shown that ANNs can be useful in predicting or distinguishing gait patterns. Recently, new studies predicting lower body joint loads using an ANN instead of an inverse dynamic approach have been reported. However, typically an ANN needs training data to make a prediction. Seung Eel Oh et al. (2013) study, developed an improved method for predicting GRFM for a normal gait with high accuracy and without the need for GRF-measuring devices during walking. The network utilized a back-propagation algorithm and the gradient steepest descent method (Schalkoff, 1997) [58]. The hidden layer with three nodes consisted of a bipolar sigmoid transfer function. In this study, the gait data are considered in the ANN training and also used as a test sample to validate the developed ANN. The results show that the predicted GRF is worse than the ones measured with force plates due to general limitations of an Artificial Neural Network, but if perfected, this method could give good results also in prediction of GRFM.

4. METHODS

4.1. Introduction

Human movement, especially the gait analysis, typically requires an optical motion capture system formed by infrared cameras for tracking marker trajectories to estimate kinematic data as well as multi-axial force plates (one or more) for the measurement of ground reaction forces and moments (GRFM) and the estimation of joint kinetics (joint moment and powers). Moreover, musculoskeletal (MSK) modeling and simulation tools combine the measures also of muscle activation from electromyography to do investigations of motor control and MSK loading. These measurement techniques are however not transferable outside the laboratory environment, especially in clinical applications, due to the excessive costs, time and expertise required. Further the use of these force plate data requires that each foot lands on a single force plate, creating a highly constrained environment.

The aim of this research study is to develop a method, based on optimization problem, for the estimation of GRFMs of three different trials of five healthy subjects, only from motion capture (MoCap) data, inverse kinematics (IK) simulations and a scaled MSK model with five spheres applied on the right foot. The estimation of GRFs is an important part of MSK analysis for trying to monitor the complex joint biomechanics, including joints moments and muscular activations. These measurements are used for clinical applications, where estimations of MSK loading is an important clinical need.

The next paragraphs of this thesis are going to describe the steps of the research study and the software that were used.

4.2. Data collection

The first step of this study was to obtain all the data in order to get the inputs of the optimization problem.

3D Motion Capture data were collected at the Movement and Posture Analysis Laboratory Leuven (MALL), at the Department of Movement Science of the KU Leuven, in Belgium. There were analyzed five healthy adults with no previous musculoskeletal or neurological disorders volunteered for the study (1 male and 4 females with average height of 1.68 ± 0.09 m and weight of 73.04 ± 11.97 kg). All subjects were performed in five different walking trials. The following table summarizes all the subject’s demographics.

Subject	Gender	Height [m]	Body weight [Kg]	BMI [Kg/m²]	Average gait speed [m/s]
H01	M	1.81	87.6	26.25	1.65
H02	F	1.69	77.5	26.96	1.22
H03	F	1.70	76.5	26.30	1.33
H04	F	1.62	55.6	20.96	1.34
H06	F	1.58	68.0	26.84	1.08

Table 1: Summary of 5 healthy subject demographics 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.

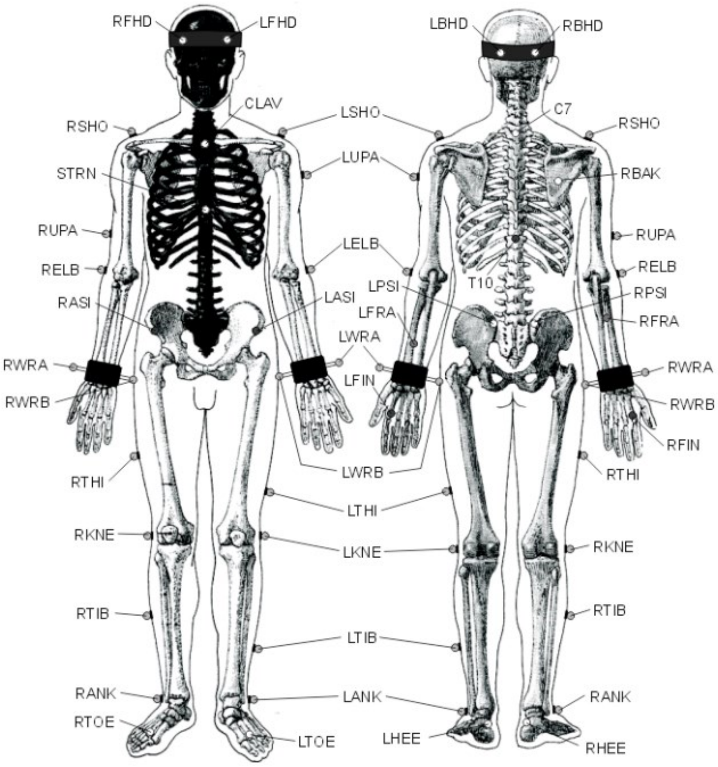


Figure 37: Plug-in-Gate marker placement.

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

It was required for each subject to do a static calibration trial, captured by the MoCap system to scale the generic MSK model. Then, the subjects performed several walking trials in order to calibrate the motion capture devices so that MoCap systems could capture ground reaction forces and moments as well as three-dimensional body movements simultaneously and synchronously.

4.3. Data elaboration workflow

4.3.1. Creation of the model workflow

The musculoskeletal model used in this study is a result of Falisse et al. work [60]. The authors of this article started from an existing model in the OpenSim library, known as the 'Full body running model' developed by Hamner et al. (2010) [76]. The Falisse et al. model has 31 degrees of freedom (DoFs) (pelvis-to-ground: 6 DoFs, hip: 3 DoFs, knee: 1 DoF, ankle: 1 DoF, subtalar: 1 DoF, metatarsophalangeal-toe: 1 DoF, lumbar: 3 DoFs, shoulder: 3 DoFs, and elbow: 1 DoF), 92 muscles actuating the lower limb and lumbar joints, 8 ideal torque motors actuating the shoulder and elbow joints, and six contact spheres per foot (three in the body calcaneus and the other three in the body toes) in order to simulate a movement and then estimate the parameters. Moreover, the knee flexion axis was fixed to its anatomical position to increase computational speed and passive stiffness (exponential) and damping (linear) were added to the lower limb and lumbar joint in order to model ligaments and other passive structures. To describe muscle excitation-activation coupling it is used the Raasch's model; instead, to describe muscle-tendon interaction and dependence of muscle force on fiber length and velocity it is used Hill-type muscle model. Then, with the Newtonian rigid body was modeled the skeletal motion and with Hunt-Crossley foot-ground contact, the authors described the smooth approximations [60].

This generic model used in this study, has been created doing some modifications on the model previously explained. Firstly, were added two more DoFs to the knee joint, allowing the joint to perform the movements of adduction-abduction and internal-external rotation in addition to flexion-extension, and to each joint, were added reserve actuators and torques in order to ensure the model was strong enough to perform all tasks. Secondly, since this study is interested in the behavior of the single right leg, the muscle group and the contact spheres were removed from the left leg and foot (incrementing the computational speed). In addition, for the finality of this study, according to the Plug-in-Gait marker set, markers were added only in the lower part (trunk and legs) of the model.

The main change was made to the spheres' configuration. This new configuration provides five spheres (and not six) designed (two in the body calcaneus and three in the body toes) so that the forces exerted by the ground during a task and recorded by the spheres themselves, could be better estimated.

In summary the generic model presents [61]:

- 35 degrees of freedom (DoFs) so arranged:
  - 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,
  - Elbow: 1 DoF.
- 49 muscles actuating the lower limb and lumbar joints defined as Thelen 2003 model placed only on the right side to simplify computational complexity;
- Reserve actuators applied to every joint to enable the simulation to run;
- 35 markers placed on the lower limbs and the trunk;
- 5 contact spheres implemented only on the right foot in order to reduce the computational complexity.

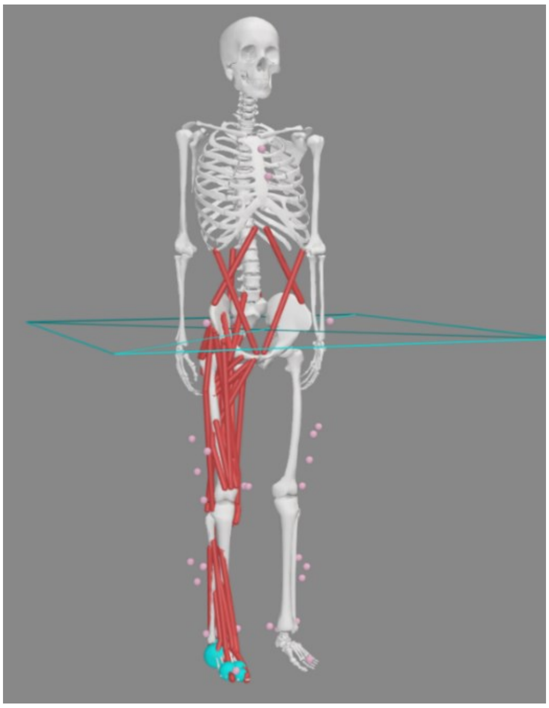


Figure 38: The generic model as a referment model of this study.

These spheres are part of the "Contact Geometry" class of the model, and they are very important for the extraction of the ground reaction forces and moments during the task. The spheres have a specific location, and radius as geometry parameters defined with respect to the right calcaneus (calcn_r body on OpenSim) frame. Each sphere is characterized by physic parameters:

- the constant stiffness (i.e., plain strain modulus): the rigidity of the sphere;
- the dissipation coefficient: the amount of energy that is dissipated or lost in a system due to friction or other forms of resistance;
- the static friction coefficient: the amount of force required to initiate sliding motion between two surfaces in contact with each other;
- the dynamic friction coefficient: the amount of force required to maintain relative motion between two surfaces in contact with each other;
- the 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);
- the transition velocity.

All of these parameters are described in the tables below with the exception of the location, which will be discuss in the next paragraphs:

	Geometric parameters	Physic parameters				
Spher e	Radius [m]	Stiffness [N/m^2]	Dissipation [s/m]	Static & Dynamic friction	Viscous friction	Transition velocity [m/s]
1	0.032	10000000	2	0.8	0.5	0.2
2	0.032	10000000	2	0.8	0.5	0.2
3	0.021	10000000	2	0.8	0.5	0.2
4	0.021	10000000	2	0.8	0.5	0.2
5	0.016	10000000	2	0.8	0.5	0.2

Table 2: Summary of the five spheres parameters.

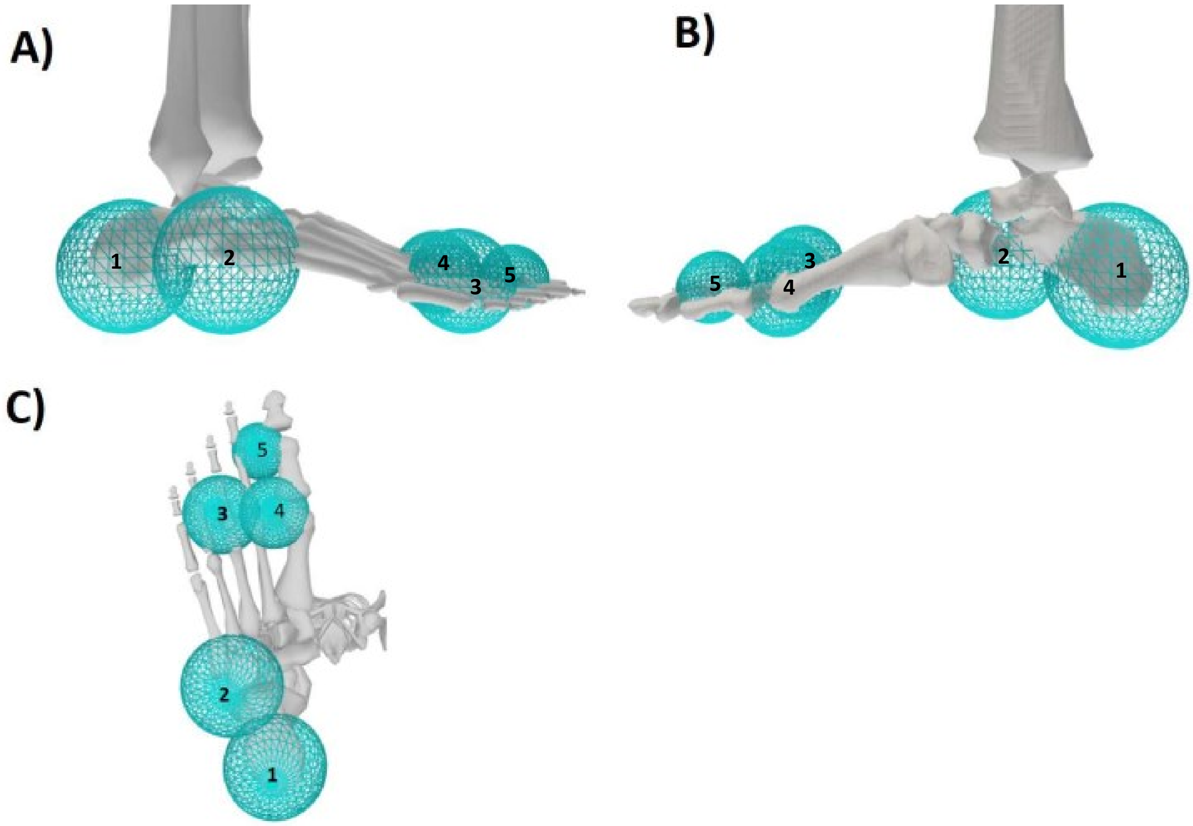


Figure 39: Contact sphere placement on the foot: A) lateral view; B) medial view; C) plantar view [61].

The described before was a generic model, used in order to generate five new musculoskeletal scaled models, one for each subject, based on the data from motion capture measurements (weight, marker placements, 3D body segments) .

These models were obtained thanks to OpenSim 4.4 Scaling tool [62]. It was necessary to add the mass of the subject of interest and the marker data in the settings section. The markers data were taken from a MoCap file during the static position, named T-pose (from the subject's position with the arms up); this file, used as reference, is called Tpose.trc.

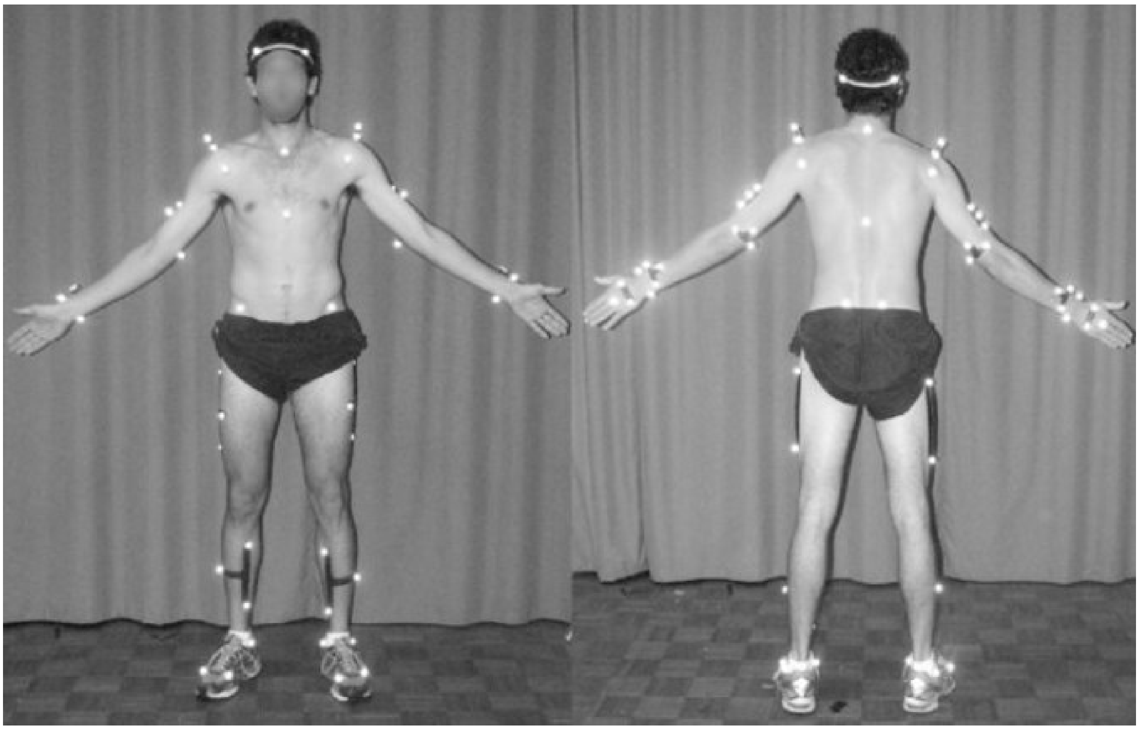
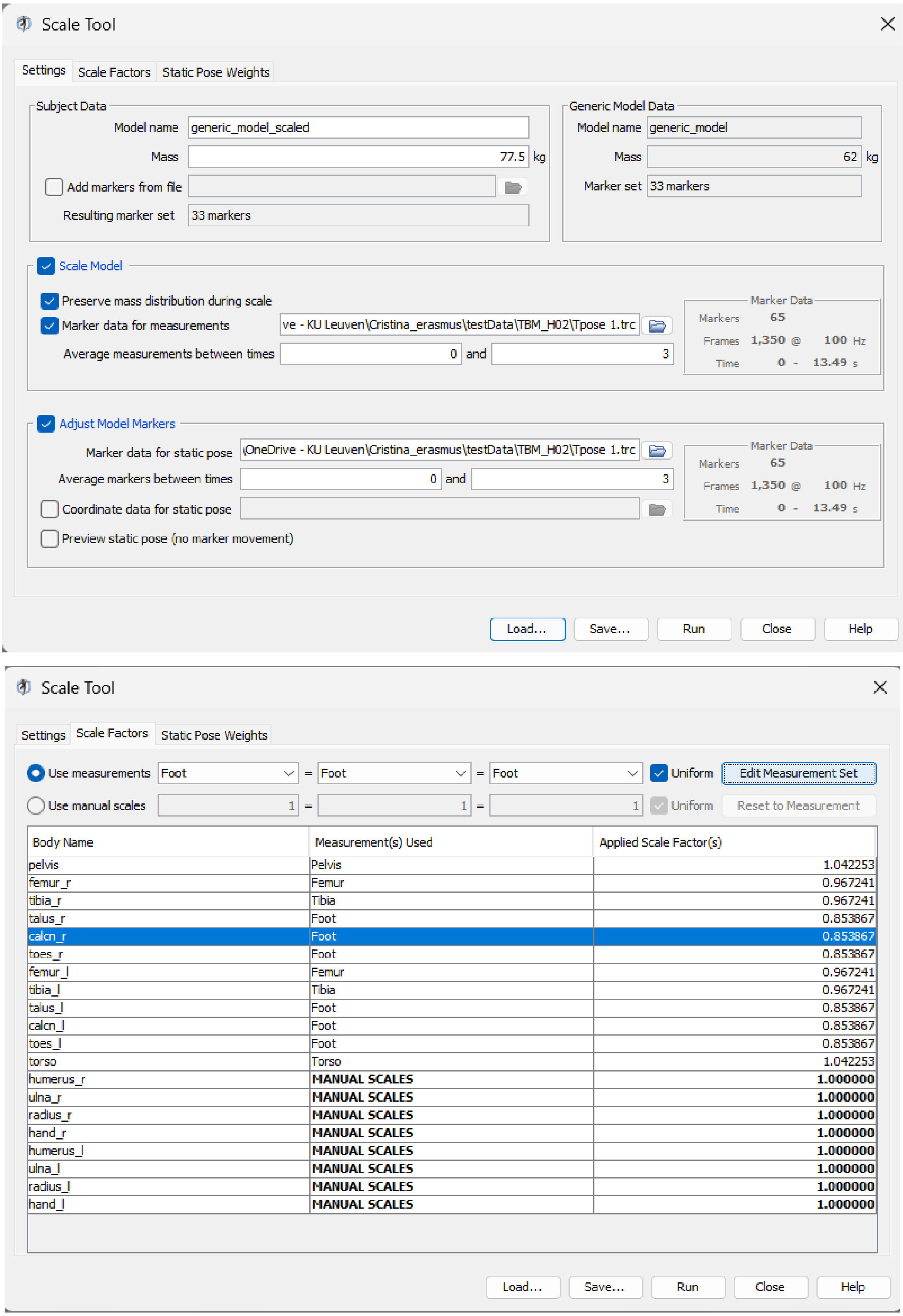


Figure 40: A man standing in an anatomical T-pose. [63].

This file was useful to have customized scale factors generated in the Scale Factors section in OpenSim, obtained by comparing distances between the markers of the generic model and the markers of the T-pose subject. It is important to underline that in the OpenSim tool was necessary to create the measurement set, that is the distances, between the marker pairs of the two models assigned to pelvis, torso and both legs to justify a symmetric behavior, as can be seen in the figure above [Fig.38]. Moreover, knowing the marker data, it was easier to assign less contribution to markers that are part of a cluster during the scaling operation and to the markers that are not important for this work (i.e., the markers of the upper body).



Scale Tool

Settings | Scale Factors | Static Pose Weights

Subject Data

Model name: generic_model_scaled

Mass: 77.5 kg

□ Add markers from file

Resulting marker set: 33 markers

Generic Model Data

Model name: generic_model

Mass: 62 kg

Marker set: 33 markers

☑ Scale Model

☑ Preserve mass distribution during scale

☑ Marker data for measurements: ve - KU Leuven\Cristina_erasmus\testData\TBM_H02\Tpose 1.trc

Average measurements between times: 0 and 3

Marker Data

Markers 65

Frames 1,350 @ 100 Hz

Time 0 - 13.49 s

☑ Adjust Model Markers

Marker data for static pose: \OneDrive - KU Leuven\Cristina_erasmus\testData\TBM_H02\Tpose 1.trc

Average markers between times: 0 and 3

□ Coordinate data for static pose

□ Preview static pose (no marker movement)

Marker Data

Markers 65

Frames 1,350 @ 100 Hz

Time 0 - 13.49 s

Scale Tool

Settings | Scale Factors | Static Pose Weights

● Use measurements: Foot = Foot = Foot ☑ Uniform

○ Use manual scales: 1 = 1 = 1 ☑ Uniform

Body Name	Measurement(s) Used	Applied Scale Factor(s)
pelvis	Pelvis	1.042253
femur_r	Femur	0.967241
tibia_r	Tibia	0.967241
talus_r	Foot	0.853867
calcn_r	Foot	0.853867
toes_r	Foot	0.853867
femur_l	Femur	0.967241
tibia_l	Tibia	0.967241
talus_l	Foot	0.853867
calcn_l	Foot	0.853867
toes_l	Foot	0.853867
torso	Torso	1.042253
humerus_r	**MANUAL SCALES**	**1.000000**
ulna_r	**MANUAL SCALES**	**1.000000**
radius_r	**MANUAL SCALES**	**1.000000**
hand_r	**MANUAL SCALES**	**1.000000**
humerus_l	**MANUAL SCALES**	**1.000000**
ulna_l	**MANUAL SCALES**	**1.000000**
radius_l	**MANUAL SCALES**	**1.000000**
hand_l	**MANUAL SCALES**	**1.000000**

Figures 41-42: Example of using OpenSim Scaling tool in this study [62].

The Scaling tool does not consider any contact geometry, so if the model contains these geometries, their relative position to the body to which they are assigned does not change. Since during scaling the body measurements change, the unchanged position of the contact geometry, that is in this case the spheres on the right foot, could be a source of errors for the extraction of ground reaction forces and moments.

For this reason, after the scaling of the generic model, the position of all the five spheres for each new scaled model was fixed with the received MATLAB custom function called "fixContactSpherePositionAfterScaling.m". This function accepts as inputs the generic model, as the model of spheres position referment, and the new scaled model, as the model with different anthropometry, and calculates new scaling parameters related to the spheres on the foot, thus adjusting their position. Then, the output of this function is the final musculoskeletal model used in this work, called scaled&fixed_model.osim.

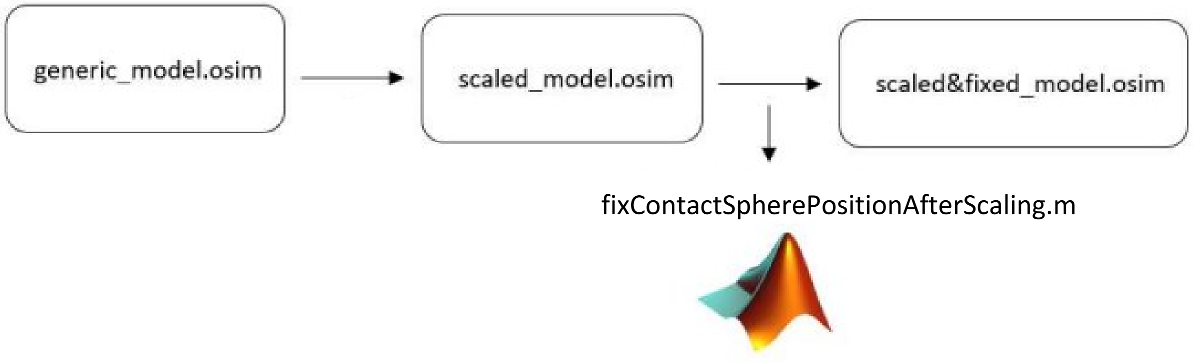
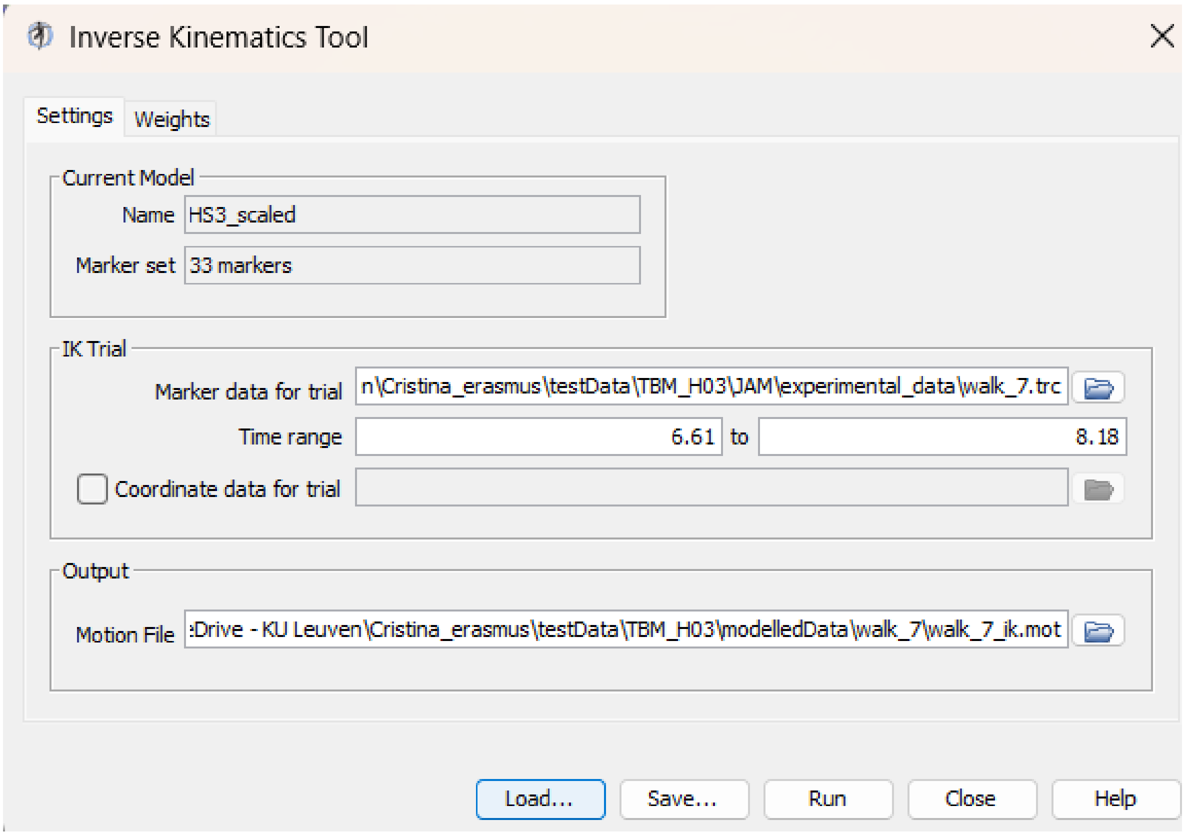


Figure 43: The creation of the model workflow [61].

4.3.2. *Inverse kinematics workflow*

Once obtaining the final model, it was necessary to evaluate the inverse kinematics. First, inverse kinematics was obtained from the Inverse Kinematics Tool in OpenSim [64]. In the settings section of this tool, the markers data file .trc captured by MoCap of each walking trial of the subject, has to uploaded. Automatically, the corresponding time range in seconds is going to appear in the window, during which the subject performed that specific trial. As output, the inverse kinematics tool will create a .mot file containing all the segment body positions in degrees for all the axes during the walking.



Current Model

Name HS3_scaled

Marker set 33 markers

IK Trial

Marker data for trial n\Cristina_erasmus\testData\TBM_H03\JAM\experimental_data\walk_7.trc

Time range 6.61 to 8.18

□ Coordinate data for trial

Output

Motion File :Drive - KU Leuven\Cristina_erasmus\testData\TBM_H03\modelledData\walk_7\walk_7_ik.mot

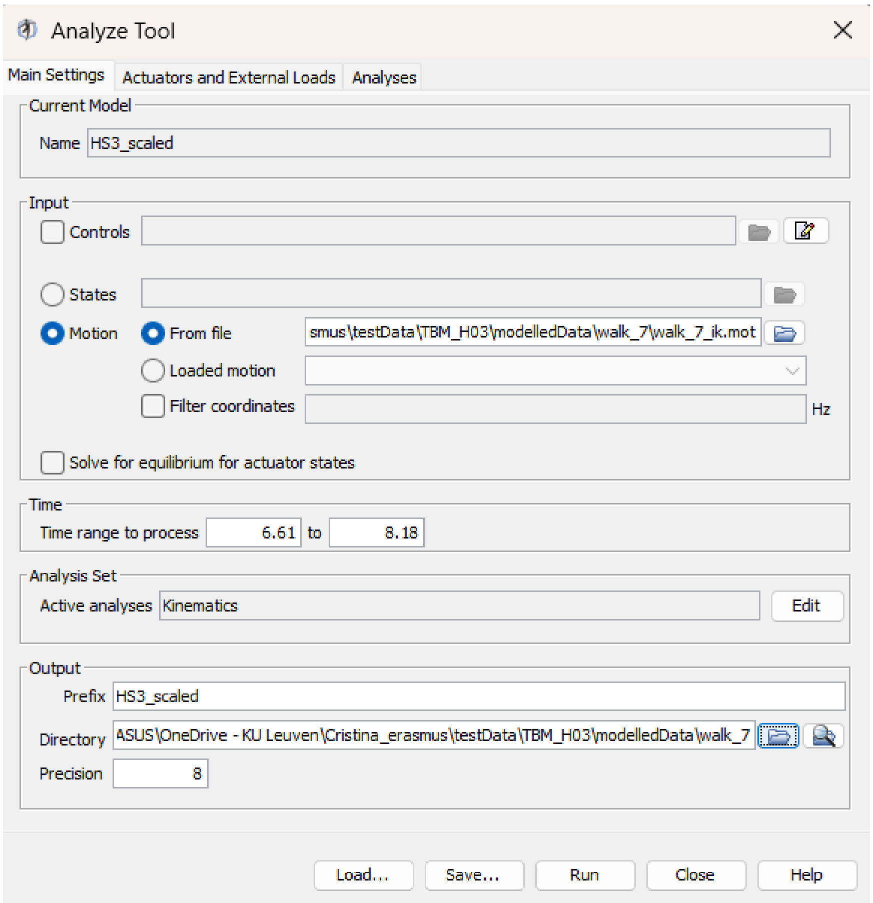
Figure 44: Example of using Inverse Kinematics OpenSim tool in this work [64].

OpenSim IK tool goes through each time step (frame) of motion and computes generalized coordinate values which positions the model in a pose that "best matches" experimental markers and coordinate values for that time step. The experimental coordinate values are the joint angles obtained directly from a motion capture system. Mathematically, the "best match" is expressed as a weighted least squares problem, whose solution aims to minimize both marker and coordinate errors [65]. But this motion does not ensure the correct placement of the foot (i.e., the contact of the five spheres) on the ground during the stance phase, that is from Heel-strike to Toe-off. This happens because of the laboratory calibration, since the model, and its system, does not automatically recognize where the floor is. Generally, to resolve this problem, it is used by the Vicon software to indicate where the ground is, directly in the MoCap system.

In this study, there is no knowledge of the exact geometry of the foot and the floor. For this reason, in the kinematics trial measured from MoCap (file .trc) and then from the cited tool, there is a mismatch between the foot and the floor. To reduce as much as possible this mismatch, and so adjust the kinematics to ensure the contact of the spheres with the ground, the Moco approach described in more detail in the study [61] was followed.

OpenSim Moco [66] is a software toolkit for the optimization of the motion and control of MSK models using the direct collocation that handles diverse problems (motion tracking, motion prediction, parameter optimization, etc.) for MSK simulations and also handles kinematic constraints, which enable modeling of kinematic loops and complex anatomy. The direct collocation optimal control method allows to estimate the muscle activity and predict new motions without the need for experimental data. In the direct collocation method, the states and controls 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. The dynamics requires the time derivative of the state splines to match the derivative from the system differential equations at specified time points. This method then produces a nonlinear program in which the states are introduced as variables and the system dynamics are enforced as constraints.

In this approach, at the beginning, are defined states and controls that Moco needed in the MSK model. The states are the degrees of freedom (DOFs) of the model's joints and both angular and velocity values; the controls are the muscles present on the model in terms of activations and fiber lengths, and the value of the reserve actuators. These data are present in the 'StatesReporter_states.sto' file, obtained thanks to the OpenSim Analyze Tool as it can see in the figure below.


Analyze Tool

Main Settings | Actuators and External Loads | Analyses

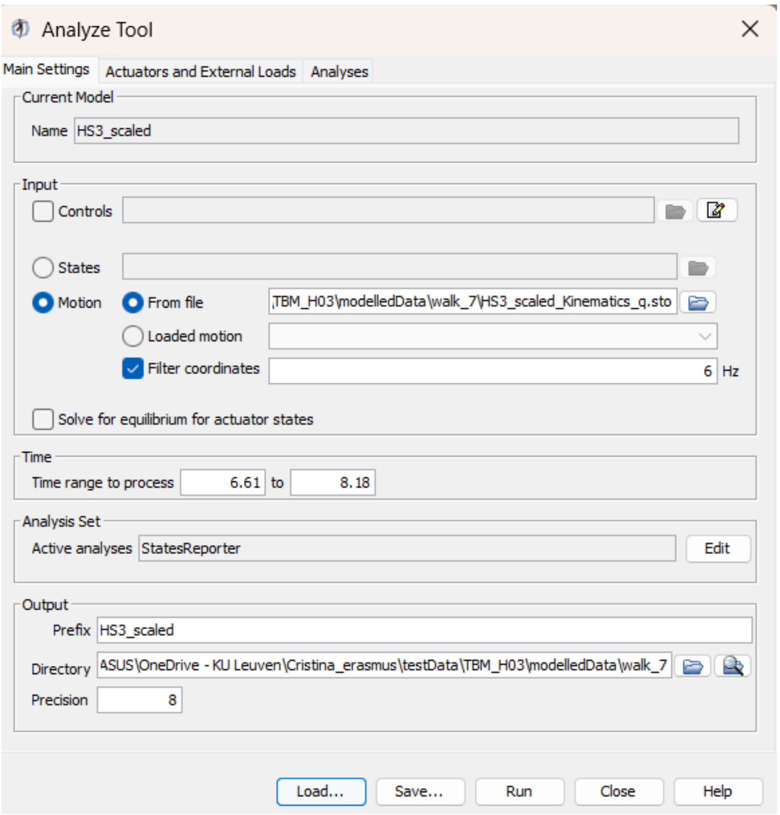
Current Model
Name HS3_scaled

Input
□ Controls
○ States
● Motion ● From file smus\testData\TBM_H03\modelledData\walk_7\walk_7_ik.mot
○ Loaded motion
□ Filter coordinates Hz
□ Solve for equilibrium for actuator states

Time
Time range to process 6.61 to 8.18

Analysis Set
Active analyses Kinematics

Output
Prefix HS3_scaled
Directory ASUS\OneDrive - KU Leuven\Cristina_erasmus\testData\TBM_H03\modelledData\walk_7
Precision 8



Current Model

Name HS3_scaled

Input

☐ Controls

○ States

● Motion ● From file TBM_H03\modelledData\walk_7\HS3_scaled_Kinematics_q.sto

○ Loaded motion

☑ Filter coordinates 6 Hz

☐ Solve for equilibrium for actuator states

Time

Time range to process 6.61 to 8.18

Analysis Set

Active analyses StatesReporter

Output

Prefix HS3_scaled

Directory ASUS\OneDrive - KU Leuven\Cristina_erasmus\testData\TBM_H03\modelledData\walk_7

Precision 8

Figures 45-46: An example of using the OpenSim Analyze Tool in this study.

Then, it was necessary to use a custom MATLAB code containing the function from OpenSim Moco Tool, called MocoTrack [66]. The inputs of this function are the scaled musculoskeletal model and the States Reporter file (.sto) created containing the motion to improve. To allow modifications to the model and to IK file, the MATLAB 'ModelProcessor' and 'TableProcessor' commands were used, and they were performed by their respective operators 'ModelOperators' and 'TableOperators'. The modifications that were applied are:

- 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 these modifications to the inputs, the code provides the application of two cost functions implemented by Moco. The cost functions are so defined as:

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

For both functions, specific weights were set in order to obtain the desired movement tracking and the estimated muscle activations as close as possible to the measured ones. The next step of the approach followed was to set the settings required by MocoProblem built into the MocoTrack package, to solve the problem of this study. Then, it was necessary to set bounds, i.e., the ranges of values within states produced as output by Moco must be, so as to impose an additional constraint to limit deviations from the data provided as input. The final step was invoking the MocoSolver solver also present in the MocoTrack package, with which was possible to launch the MocoTrack analysis, obtaining as output the Moco_solution_MoCap.sto file. This file contains the tracked motion (that is, the values of joint angles and their respective velocities), and the values of muscle activations and length of muscle fibers and it is crucial for the optimization problem that is the main focus of this study. Further information about MocoTrack packages are in [66] article.

So, the use of OpenSim Moco was important to ensure that during the stance phase, the contact spheres are in contact with the ground, because it determined a change in joint angles permitting a better estimation of ground reaction forces extracted from the spheres. This has allowed a better definition of the optimization problem, that is the core of this thesis, which is described in the next paragraph.

Moreover, it was also calculated the measured IK from MoCap data, in order to compare these values with the ones obtained with OpenSim Moco workflow.

4.3.3. Optimization problem workflow

The aim of the optimization problem, of this thesis is based on, is to find a position of the five spheres on the right foot in order to extract GRFs from the model as close as possible to the GRF measured from force plates during MoCap measurements, that is the state of the art at the time of writing this thesis. The next paragraphs are going to describe all the procedures required to get the best solution obtained in this study.

Extraction of Ground Reaction Forces

During walking, the forces that the ground exerts on the right foot are collected by the five spheres placed on it. In order to get these forces during the movement, it was created a custom MATLAB function called 'extractGRFfromSpheres.m' [Appendix] that accepts as inputs a motion file, that in this case is the movement tracked with Moco, and the scaled and fixed model. With this function, it is possible to obtain the real force extraction from each sphere, considered as a ForceSet class, where the sphere is the object called 'SmoothSphereHalfSpaceForce', and then make the vector summation of all of them. The resulting force is thus decomposed in the three coordinates (Anteroposterior, Vertical and Mediolateral) and then saved in a matrix in function of the motion time.

Evaluation of COP and GRM

Starting from the force matrix created, it was possible to calculate the center of pressure (COP) and the ground reaction moment (GRM) using the zero-moment-point (ZMP) method [67]. This method is usually applied in robotics for the determination of a humanoid robot (or similar objects) stability. Stability is defined as the point on the ground where the sum of all forces and the sum of all moments around this point is equal to zero. The ZMP approach is thus important for maintaining stability and balance, since the point indicates the location where the center of mass is, to prevent and avoid falls. For this reason, the ZMP method was applied for the evaluation of the COP, so the point of application of the GRFs, considered as the point where the resultant tangential moments of the active forces are equal to zero (ZMP).

The position of ZMP and so the COP coordinates, can be calculated using these formulas derived from the method described above:

1. $x_{COP} = \underline{M}_z/\underline{GRF}_y$
2. $z_{COP} = -\ \underline{M}_x/\underline{GRF}_y$
3. $y_{COP} = 0$
4. $\underline{GRM}_y = \underline{M}_y - z_{COP} \cdot\ \underline{GRF}_x + x_{COP} \cdot\ \underline{GRF}_z$
5. $\underline{GRM}_x = \underline{GRM}_z = 0$

Where (x_{COP}, y_{COP}, z_{COP}) are the COP coordinates, $\underline{GRM} = (\underline{GRM}_x, \underline{GRM}_y, \underline{GRM}_z)$ are the ground reaction moments applied to COP and $\underline{M} = (\underline{M}_x, \underline{M}_y, \underline{M}_z)$ are the ground reaction moments extracted from the spheres.

COP coordinates and the GRMs around it were then added to the matrix of 'extractGRFfromSpheres.m' function. The output of this function thus created a new .mot file called 'walk_GRFMOCO.mot' that contains all the values expressed in the matrix.

Definition of the optimization problem

Before the formulation of the optimization problem, it was necessary to create a custom function on MATLAB, that calculates a new vector of ground reaction forces estimated from contact spheres in function of their position. This function accepts as inputs: the scaled model, a new vector of spheres position, the starting and the ending position of the time range of interest, and the components (that could be considered singularly or together) [Appendix]. The function takes the new position of the spheres and sets this position to the input model, creating a new model with the location changed. At the end, thanks to the function 'extractGRFfromSpheres.m' cited in the previous paragraph, this custom function creates the corresponding ground reaction forces from the 'walk_GRFMOCO.mot' file.

Finally, it could be possible to use the 'fminsearchbnd.m' function [68], from the MATLAB optimization tool (in particular, solver-based nonlinear optimization) for the finding of the optimal solution. The function 'fminsearchbnd.m' uses the Nelder-Mead simplex algorithm with bound constraints; this algorithm uses a simplex of n + 1 point for n-dimensional vectors x. The algorithm first makes a simplex around the initial guess x_0 by adding 5% of each component $x_0(i)$ to x_0, and using these n vectors as elements of the simplex in addition to x_0. Then, the algorithm modifies the simplex repeatedly according to the following procedure:

1. Let $x(i)$ denote the list of points in the current simplex, $i = 1, ...,n + 1$.
2. Order the points in the simplex from lowest function value $f(x(1))$ to highest $f(x(n + 1))$. At each step in the iteration, the algorithm discards the current worst point $x(n + 1)$, and accepts another point into the simplex. [Or, in the case of step 7 below, it changes all n points with values above $f(x(1))$].
3. Generate the *reflected* point

   $$r = 2m - x(n + 1),$$

   where

   $$m = \Sigma x(i)/n, i = 1...n,$$

   and calculate $f(r)$.

4. If $f(x(1)) \leq f(r) < f(x(n))$, accept r and terminate this iteration, giving as result ‘**Reflect’.**

5. If $f(r) < f(x(1))$, calculate the expansion point s

   $$s = m + 2(m - x(n + 1)),$$

   and calculate $f(s)$.

   a. If $f(s) < f(r)$, accept s and terminate the iteration, giving as result ‘**Expand’.**

   b. Otherwise, accept r and terminate the iteration, giving as result ‘**Reflect’.**

6. If $f(r) \geq f(x(n))$, perform a *contraction* between m and either $x(n + 1)$ or r, depending on which has the lower objective function value.

   a. If $f(r) < f(x(n + 1))$ (that is, r is better than $x(n + 1)$), calculate

      $$c = m + (r - m)/2$$

      and calculate $f(c)$. If $f(c) < f(r)$, accept c and terminate the iteration, giving as result ‘**Contract outside’.**

      Otherwise, continue with Step 7 (Shrink).

   b. If $f(r) \geq f(x(n + 1))$, calculate

      $$cc = m + (x(n + 1) - m)/2$$

      and calculate $f(cc)$. If $f(cc) < f(x(n + 1))$, accept cc and terminate the iteration, giving as result ‘**Contract inside’.**

      Otherwise, continue with Step 7 (Shrink).

7. Calculate the n points

   $$v(i) = x(1) + (x(i) - x(1))/2$$

   and calculate $f(v(i))$, $i = 2, ..., n + 1$.

   The simplex at the next iteration is $x(1), v(2), ..., v(n + 1)$, giving as result ‘**Shrink’.**

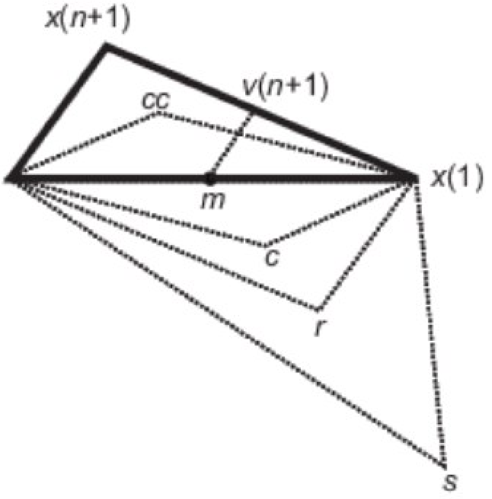


Figure 47: Points that fminsearchbnd might calculate in the procedure, along with each possible new simplex. The original simplex has a bold outline. The iterations proceed until they meet a stopping criterion [68].

In this study, the input parameters of 'fminsearchbnd.m' function thus are:

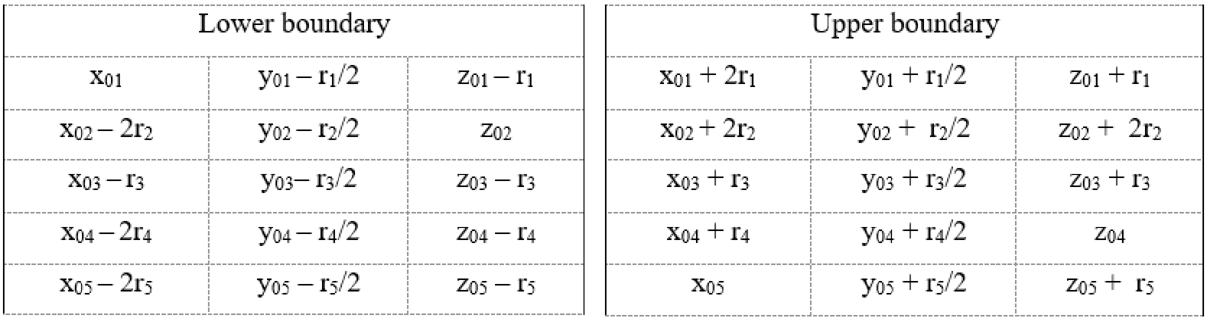
- The starting position of the spheres p_0, as the initial guess (values expressed in [Tab.3]);
- The range within the spheres could move through the foot, defined with a lower and an upper boundary;
- The Optimization options, specified as a structure such as 'optimset' (a function provided by MATLAB Optimization tool ad hoc for this type of functions) returns. The options chosen for this optimization problem are:
    - Display: This is the level of display. It was set to 'iter' in order to display output at each iteration;
    - MaxFunEvals: Maximum number of function evaluations f(p) allowed (it has to be a positive integer). It was set to 1500;
    - MaxIter: Maximum number of iterations allowed (it has to be a positive integer). It was set to 1500;
    - TolFun: Termination tolerance on the function value, a positive scalar. It was set to 10^{-30};
    - TolX: Termination tolerance on the position vector p, a positive scalar. It was set to 10^{-30};

- The function to minimize that is dependent from the position p, f(p) that will be called the optimized function.

	Location (x, y, z) [m]				
Subject Sphere	H01	H02	H03	H04	H06
1	(0.009, 0.006, -0.004)	(0.009, 0.006, -0.004)	(0.009, 0.006, -0.004)	(0.008, 0.005, -0.004)	(0.008, 0.005, -0.004)
2	(0.05, 0.01, 0.02)	(0.05, 0.01, 0.02)	(0.05, 0.01, 0.02)	(0.05, 0.009, 0.02)	(0.05, 0.01, 0.02)
3	(0.15, 0.002, 0.02)	(0.14, 0.002, 0.02)	(0.14, 0.002, 0.02)	(0.13, 0.002, 0.02)	(0.13, 0.002, 0.02)
4	(0.15, 0.002, -0.01)	(0.14, 0.002, -0.009)	(0.14, 0.002, -0.009)	(0.13, 0.002, -0.008)	(0.13, 0.002, -0.009)
5	(0.18, 0.0004, -0.003)	(0.17, 0.0004, -0.003)	(0.17, 0.0004, -0.003)	(0.16, 0.0004, -0.003)	(0.16, 0.0004, -0.003)

Table 3: Initial position p_0 of the five spheres for each subject. These positions are from scaled positions based on the foot sizing.

The range within the spheres could move, is defined by their initial position and their radius, in order to get a physiological optimal position; defined $p_{0i} = (x_{0i}, y_{0i}, z_{0i})$ the starting position and r_i the spheres radius, with i = 1, …, 5, the lower and upper boundaries are:



Lower boundary		
x_{01}	$y_{01} - r_1/2$	$z_{01} - r_1$
$x_{02} - 2r_2$	$y_{02} - r_2/2$	z_{02}
$x_{03} - r_3$	$y_{03} - r_3/2$	$z_{03} - r_3$
$x_{04} - 2r_4$	$y_{04} - r_4/2$	$z_{04} - r_4$
$x_{05} - 2r_5$	$y_{05} - r_5/2$	$z_{05} - r_5$

Upper boundary		
$x_{01} + 2r_1$	$y_{01} + r_1/2$	$z_{01} + r_1$
$x_{02} + 2r_2$	$y_{02} + r_2/2$	$z_{02} + 2r_2$
$x_{03} + r_3$	$y_{03} + r_3/2$	$z_{03} + r_3$
$x_{04} + r_4$	$y_{04} + r_4/2$	z_{04}
x_{05}	$y_{05} + r_5/2$	$z_{05} + r_5$

These boundaries have been chosen based on their initial position p_{0i} on the right foot with respect to its size, scaled with respect to each subject's demographics; for this reason, there are different boundaries between the spheres and the space directions.

The optimization function that is used for this problem analysis is the root mean square error (RMSE) of the difference between the ground reaction forces measured in the laboratory using force plates, considered as reference and the ones estimated from MoCap data with Moco kinematics modifications. The formula is expressed below:

$$\text{Optimal function} = \text{RMSE(p)} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|GRFmeasured - GRFestimated(p)|^2}$$

where GRFestimated is the result of the 'estim.m' function, that is dependent from the position. At each iteration, the temporary vector position created with the algorithm follow with which 'fminsearchbnd' is based, set the position of the model creating a new momentary one, and thus a transitory vector of GRFs; then, these forces are subtracted from the GRFmeasured and later is calculated the root mean square of this difference. It follows the same process until the number of iterations is exceeded or when it reaches the tolerance.

Once the stopping criterion is met, 'fminsearchbnd' gives as results, the spheres position that made minimum the root mean square error, and the value of this error. With this optimal position it was defined in the final model and extracted the optimal ground reaction forces in the same way as before, creating the last file called 'GRF_MocoOptim.mot'.

At the end both measured and optimal estimated ground reaction forces were filtered with the low-pass Butterworth filter at 20Hz and then with the filtered forces it was calculated the correlation coefficient.

Correlation is a statistical measure that expresses the linear relationship between two variables [69]. The correlation is unable to verify the presence or the effect of other variables other than the two examined. In particular, it says nothing about cause and effect. The correlation is described by a value that does not have a specific unit, called the correlation coefficient, ranging from -1 to +1 and denoted by r. Statistical significance is indicated by a p-value (p-value is a measure of probability used in hypothesis testing). Therefore, correlations are typically written using two basic numbers: r and p.

- The closer r gets to zero, the weaker the linear correlation is.
- A positive r-value indicates a positive correlation, where the values of the two variables tend to increase in parallel.

- A negative r-value indicates a negative correlation, where the value of one variable tends to increase as the other decreases.
- The p-value is the proof that, based on what we observed on the sample, we can meaningfully conclude that the correlation coefficient of the population is different from zero.
- The lack of a unit of measurement implies that the correlation is measured according to its own scale.

Another useful piece of data is N, or the number of observations. As with most statistical tests, knowing the sample size helps to judge its validity and understand how well it represents the relevant population.

Once getting the IK with OpenSim Moco and then the ground reaction forces, it was important to give attention to the calculation of joint moments with these values, obtainable with Inverse Dynamics evaluation, in order to compare them with the ones measured with MoCap systems. This would be another parameter to validate the proposed entire workflow as an alternative to the actual state of the art.

4.3.4. Inverse dynamics workflow

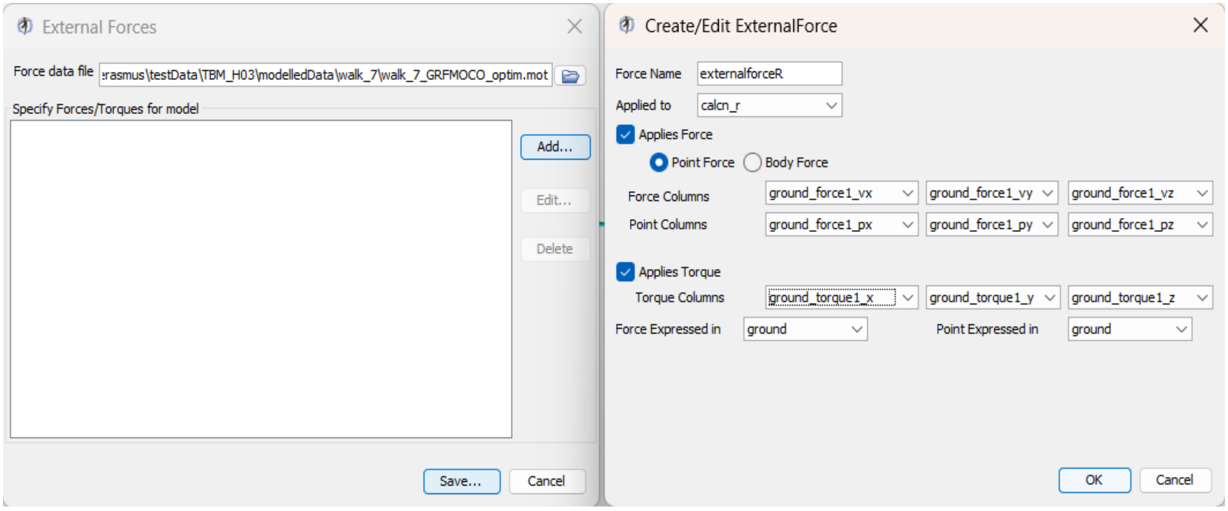
The Inverse Dynamics (ID) OpenSim Tool [70], determines the generalized forces (e.g., net forces and torques) at each joint responsible for a given movement. Given the kinematics (e.g., states or motion) describing the movement of a model and perhaps a portion of the kinetics (e.g., external loads) applied to the model, the ID Tool uses these data to perform an inverse dynamic analysis. Classical mechanics mathematically expresses the mass-dependent relationship between force and acceleration, **F** = m**a**, with equations of motion. The Inverse Dynamics Tool solves these equations, in the inverse dynamics sense, to yield the net forces and moments at each joint which produce the movement. The classical equations of motion may be written in the following form [71]:

$$\underbrace{\mathbf{M(q)\ddot{q}+C(q,\dot{q})+G(q)}}_{\text{knowns}} = \underbrace{\tau}_{\text{unknowns}}$$

Where $q, \dot{q}, \ddot{q} \in R^N$ are the vectors of generalized positions, velocities, and accelerations, respectively; M(q) ∊ R^{NxN} is the system mass matrix; C($q, \dot{q}$) ∊ R^N is the vector of Coriolis and centrifugal forces; G(q) ∊ R^N is the vector of gravitational forces; and $\tau \in R^N$ is the vector of generalized forces (N is the number of degrees of freedom).

The motion of the model is completely defined by the generalized positions, velocities, and accelerations. Consequently, all of the terms on the left-hand side of the equations of motion are known. The remaining term on the right-hand side of the equations of motion is unknown. The inverse dynamics tool uses the known motion of the model to solve the equations of motion for the unknown generalized forces [71].

In fact, in order to obtain inverse dynamics for this work, it was necessary to insert the Moco kinematics, previously created, in the main settings of this tool. Then, to add the kinetics applied to the model of this study, in the External Loads section of the same tool, the 'GRF_MocoOptim.mot' file was uploaded, specifying the ground reaction forces and moment for this scaling model by creating an external force. This external force is applied with respect to the right calcaneus (calcn_r body name in OpenSim).



External Forces

Force data file: :rasmus\testData\TBM_H03\modelledData\walk_7\walk_7_GRFMOCO_optim.mot

Specify Forces/Torques for model

Create/Edit ExternalForce

Force Name: externalforceR

Applied to: calcn_r

☑ Applies Force

● Point Force ○ Body Force

Force Columns: ground_force1_vx, ground_force1_vy, ground_force1_vz

Point Columns: ground_force1_px, ground_force1_py, ground_force1_pz

☑ Applies Torque

Torque Columns: ground_torque1_x, ground_torque1_y, ground_torque1_z

Force Expressed in: ground

Point Expressed in: ground

Figure 48: An example of using the External Forces interface.

The output of this tool is a file containing all the joint moments during the walking trials.

The ID workflow has been followed for the estimated case, (i.e., Moco IK and extracted GRF), for the measured case (i.e., MoCap IK and GRF), but also for a 'mix' case, that is Moco IK and measured GRF. This was done for the comparison between the three cases in order to confirm if this workflow could be a valid alternative to the gold standard for clinical and sport rehabilitation.

5. RESULTS

In this chapter are going to be illustrated all the results previously introduced in the Methods chapter. For each subject in this study, three trials were considered for a total of fifteen tasks evaluated. Moreover, one of the three tasks was randomly chosen as the optimized trial obtaining a certain spheres position for each subject MSK model (five trials in total); this configuration was used to extract GRFs and then compute ID in the remaining two trials (ten trials in total).

The results, to allow a better comparison, will be shown with the calculation of average and standard deviation across all the subjects' walking trials as functions of the percentage of stance phase. Before this calculation, all the ground reaction forces and joints moments were time normalized as one hundred and one data points at one percent intervals of the gait cycle.

In the obtained graphs, the blue curves will represent the reference data (force plates measurements) while the red ones will refer to the results of this study's approach. The green curves will represent the results before the optimization, well-illustrated in [61] study, with the exception of the inverse dynamics results, where the green curves will represent the measured forces with Moco IK (more details will be explained in the specific paragraph). The x-axis corresponds to the stance phase, and it will be represented as a percentage; the y-axis corresponds to the value under analysis, and it will be represented as:

- Degrees (°) for joints' angle variation;
- Body Weight (BW) for ground reaction forces;
- Meters (m) for center of pressure's position;
- Centimeters (cm) for spheres' position variation;
- Nm/kg for joint moments.

All the calculations, normalizations and creation of the graphs were done thanks to the MATLAB functions [72]. It is going to discuss the results for each value singularly in the following paragraphs.

5.1. OpenSim Moco inverse kinematics results

The OpenSim Moco approach was used in order to get an adhesion of the spheres on the ground, since the model and its system does not automatically know where the floor is, and thus having a physiological walking and an actual contact between foot and ground, then the possibility to evaluate the GRFs. Better contact implies more precise evaluation of ground reaction forces. This approach generates a variation of joint angle values, as can be seen in the table [Tab.4].

Joint	**mean ± std of RMSE of joints' angle variation [°]**
Hip Flx/Ext	2.13 ± 1.62
Hip Add/Abd	1.31 ± 1.36
Hip IR/ER	1.07 ± 0.77
Knee Flx/Ext	10.67 ± 5.64
Knee Add/Abd	4.53 ± 2.74
Knee IR/ER	4.09 ± 2.87
Ankle DF/PF	7.81 ± 4.79
Subtalar	6.33 ± 2.95

Table 4: Average value (± standard deviation) of RSME joints' angle variation for all the fifteen trials' inverse kinematics.

Average root mean square error of the variation between kinematics measured with MoCap and those calculated with OpenSim Moco function, MocoTrack, shows that the values are within the range of 1-7° with the exception of the knee flexion/extension, that is almost 11°.

5.2. Optimal ground reaction forces results

The estimated GRFs extracted from the values recorded by the five contact spheres placed on the right foot (in red) are compared in [Fig.49] with the forces recorded by the force platforms in the MoCap system (in blue). Both are represented as the three components (Anterior-posterior, Vertical and Medio-Lateral GRF) singularly, with respect to the global reference system.

These graphs were obtained using the 'plot_distribution.m' MATLAB function that represents the data as a curve for the mean, and as the area around it, for the standard deviation. In order to validate the optimization problem, two plots of GRFs comparison were generated: One for the optimized trials, one for the remaining ones [Fig.50].

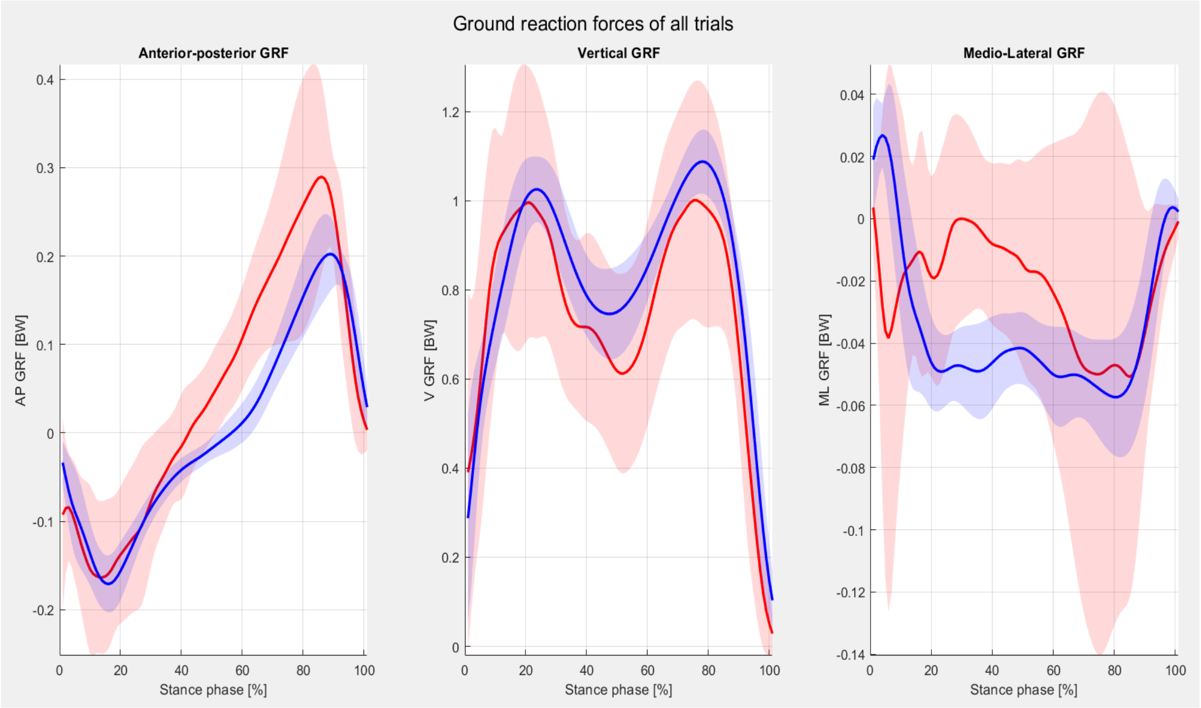


Figure 49: GRFs comparison: in red the GRFs estimated while in blue the GRFs measured for all the fifteen trials.

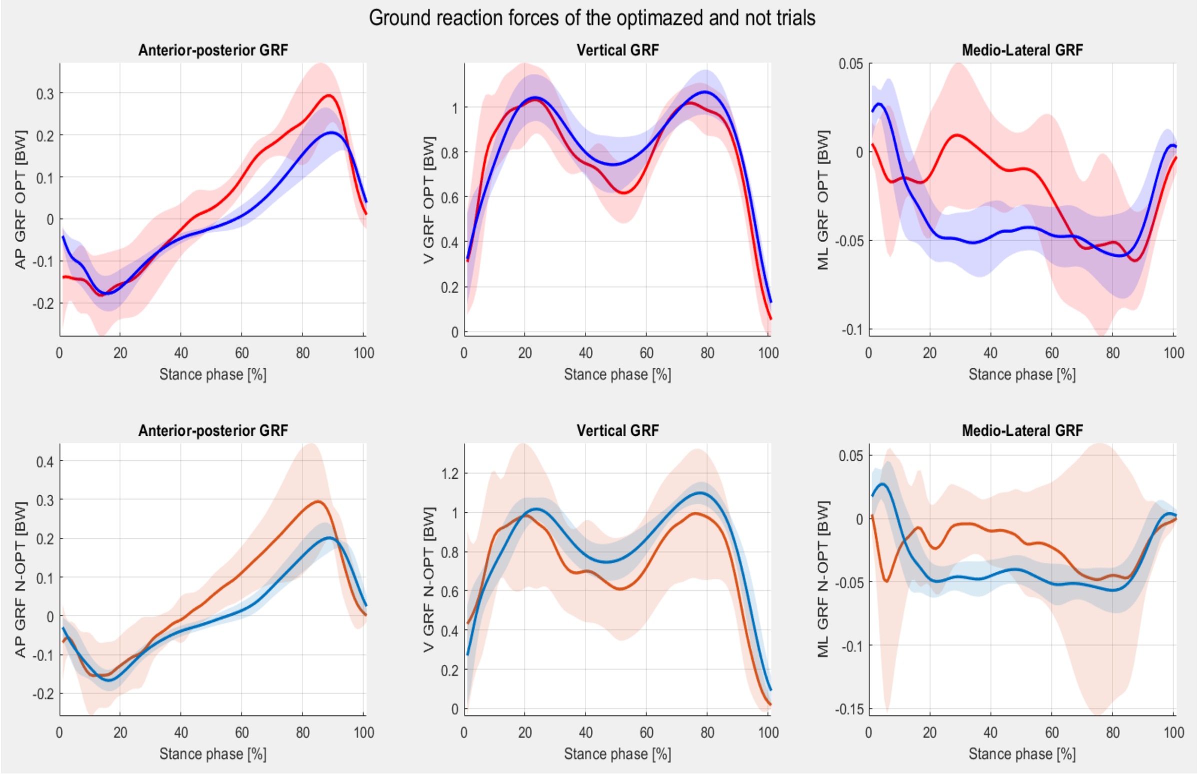


Figure 50: GRFs comparison: in red the GRFs estimated while in blue the GRFs measured for the five optimal trials in the first row, and for the ten remaining trials in the second one.

For both cases (all trials together and the division in optimized and not optimized trials), was calculated the average and standard deviation of the root mean square error (RMSE) and also of Pearson coefficient. All the results are summarized in the tables below [Tabs.5-6]:

GRF component	**RMSE [BW]**	**Pearson coefficient**
Anterior-Posterior	0.09 ± 0.04	0.82 ± 0.06
Vertical	0.23 ± 0.12	0.6 ± 0.29
Medio-Lateral	0.06 ± 0.03	0.1 ± 0.12

Table 5: Average value (± standard deviation) of GRFs' Pearson coefficient (R^2) and root mean square error (RMSE) for each component for all the trials.

GRF component	**RMSE [BW]**		**Pearson coefficient**	
	Optimal	**Not optimal**	**Optimal**	**Not optimal**
Anterior-Posterior	0.08 ± 0.02	0.09 ± 0.05	0.86 ± 0.05	0.79 ± 0.04
Vertical	0.13 ± 0.05	0.28 ± 0.11	0.72 ± 0.27	0.54 ± 0.29
Medio-Lateral	0.04 ± 0.01	0.06 ± 0.03	0.07 ± 0.07	0.12 ± 0.14

Table 6: Average value (± standard deviation) of GRFs' Pearson coefficient (R^2) and root mean square error (RMSE) for the optimal and unseen trials.

These values reflect the graphs' behavior represented in the figures [Figg.47-48], especially for the estimated medio-lateral GRF component, where the standard deviation of Root mean square error, is proportionally higher than the others two with a value of mean (± standard deviation) of 0.06 ± 0.03 BW (instead for the vertical component is 0.23 ± 0.12 BW, while for the anterior/posterior component is 0.09 ± 0.04 BW). In addition, the Pearson coefficient is worse for the medio/lateral component, with an average (± standard deviation) of 0.1 ± 0.12, that justifies the weak correlation between the estimated GRF (red) and the measured one (blue). Different behavior is for both anterior/posterior and vertical components, where the average (± standard deviation) is partially strong for the first, with a value of 0.6 ± 0.29 and a value of 0.82 ± 0.06 for the second, that evidence the strong correlation.

Regarding the optimized trials, the average (± standard deviation) of root mean square error, is lower than the consideration of all trials together case, for all GRFs components with values of 0.08 ± 0.02 BW, 0.13 ± 0.05 BW, and 0.04 ± 0.01 BW for A/P, V, and M/L component respectively; the average (± standard deviation) of Pearson coefficient is higher than the condition in which there are considered all the fifteen trials for A/P and V values of 0.86 ± 0.05 and 0.72 ± 0.27 respectively, with the exception of the M/L component, that is lower compared with the same component for all trials case, with a value of 0.07 ± 0.07 that increases the weak correlation, supported also by the presence of only one trial per subject.

Regarding the not optimized trials, there are similar values of mean (± standard deviation) of RMSE for A/P and M/L component with the case of all trials, while for the vertical component is higher compared to that case, with a value of 0.28 ± 0.11 BW. This behavior happens because the vertical component has more weight in the optimization problem (since it has the highest magnitude), and this causes a smaller error in such cases, while in the current case of not optimized trials, the error is bigger. On the contrary of the RMSE, the average (± standard deviation) of Pearson coefficient is lower than the case of all trials for A/P and V component with values of 0.79 ± 0.04 and 0.54 ± 0.29 respectively, with the exception of M/L component that is of 0.12 ± 0.14. This last value could be justified by the fact that there are two trials per subject, and this could mean more correlation than the optimized case.

Here, in the following figure [Fig.51] it is represented how the optimization changes the behavior of the GRFs components curves with respect to the initial condition of the study [61] (only for all trials). The green curves show the initial condition, while the red ones show the condition after the application of the optimization problem. It can be seen that only the vertical component is closer to the corresponding measured GRF (blue curve). This is justified by the higher weight on the optimization problem.

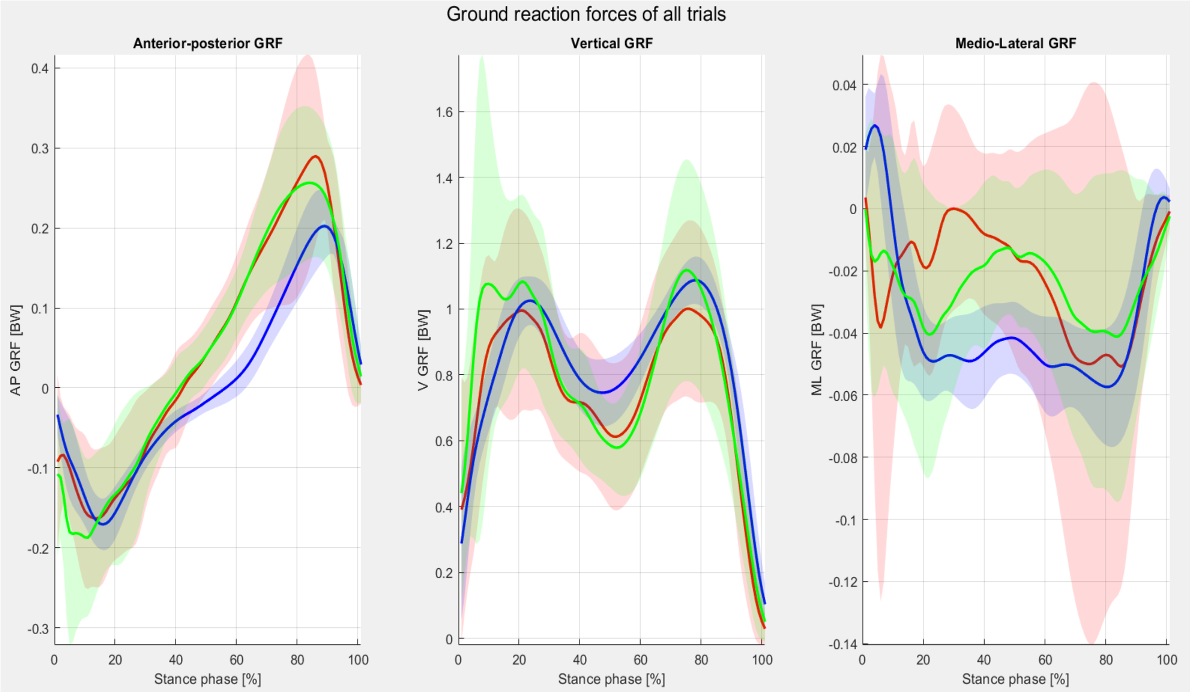


Figure 51: GRFs comparison: in red the GRFs estimated after the optimization, in blue the GRFs measured and in green the GRFs estimated before the optimization for all the fifteen trials.

It was interesting to observe the behavior of a random subject's optimized trial, and so its spheres' position, on another random subject. This has been done in order to demonstrate if this optimization problem approach could be considered subject specific or not. This step is fundamental when a physician has to decide to give a personalized treatment or could be adapted with ascertained previous clinical cases. The graphs obtained from this analysis are represented below [Fig.52]:

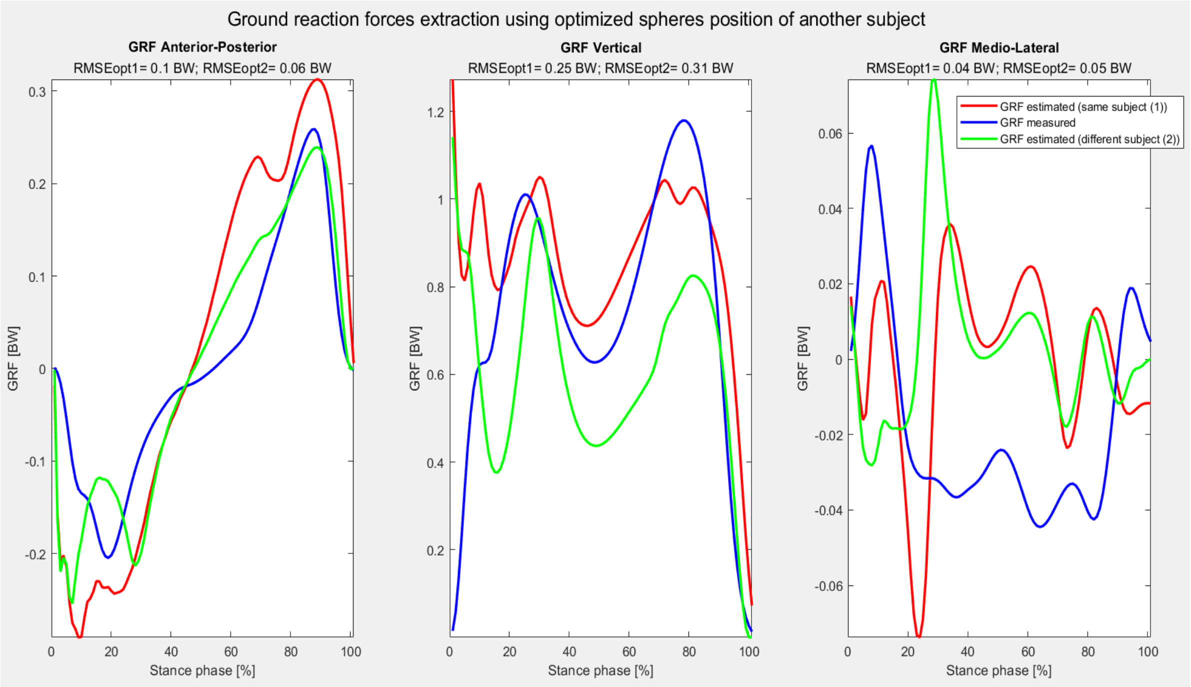


Figure 52: GRFs comparison: in red GRFs estimated using the optimization of a random subject on its random trial; in blue GRFs measured on the same trial; in green GRFs estimated using the optimization of another random subject on the same trial.

It can be seen that the optimized GRFs behavior of the same subject (red curves) is better than the GRFs using the optimization of a different subject (represented by the green curves), with the exception of the anterior-posterior component, but this behavior is anyway far from the measured GRFs (blue curves), in fact the value of root mean square error is 0.06 BW (against 0.1 BW of the first case). The values of RMSE of the other two components are 0.25 BW and 0.04 BW for vertical and medio-lateral components respectively (against the values of 0.31 BW, and 0.05 BW respectively for the first case).

5.3. Variation of spheres position results

In the paragraph of the optimization problem workflow of the Methods chapter was explained that, in order to obtain the minimum RMSE of the difference between the measured and estimated GRFs, the five spheres have been moved within the right foot according to the previously defined boundaries, to extract the forces that satisfy this optimization relationship. The last iteration of the MATLAB optimization framework ‘fminsearchbnd.m’, has generated as output, the position vectors of all the spheres, that are shown in the following table for each subject [Tab.7].

Spheres' location (x, y, z) [m]					
Subject **Sphere**	H01	H02	H03	H04	H06
1	(0.013, 0.003, 0.01)	(0.009, 0.005, -0.018)	(0.02, 0.008, -0.04)	(0.013, 0.019, -0.005)	(0.064, 0.001, 0.002)
2	(0.06, 0.01, 0.02)	(0.068, 0.026, 0.018)	(0.076, 0.009, 0.018)	(0.042, 0.017, 0.016)	(0.112, 0.023, 0.028)
3	(0.142, 0.004, 0.026)	(0.16, 0.013, 0.021)	(0.133, 0.008, 0.016)	(0.119, 0.013, 0.013)	(0.152, 0.011, -0.005)
4	(0.168 0.006, -0.001)	(0.14, 0.005, -0.009)	(0.161, 0.012, -0.011)	(0.117, -0.002, -0.009)	(0.153, 0.006, -0.014)
5	(0.18, 0.0008, -0.019)	(0.172, -0.003, 0.009)	(0.171, -0.008, -0.007)	(0.156, 0.003, -0.019)	(0.159, 0.008, -0.009)

Table 7: Final position of the five spheres for each subject after the application of the optimization problem.

Thanks to OpenSim, it was possible to see how the five spheres actually moved in the right foot. In fact, the following figures show the displacement in the lateral and plantar view [Fig.53].

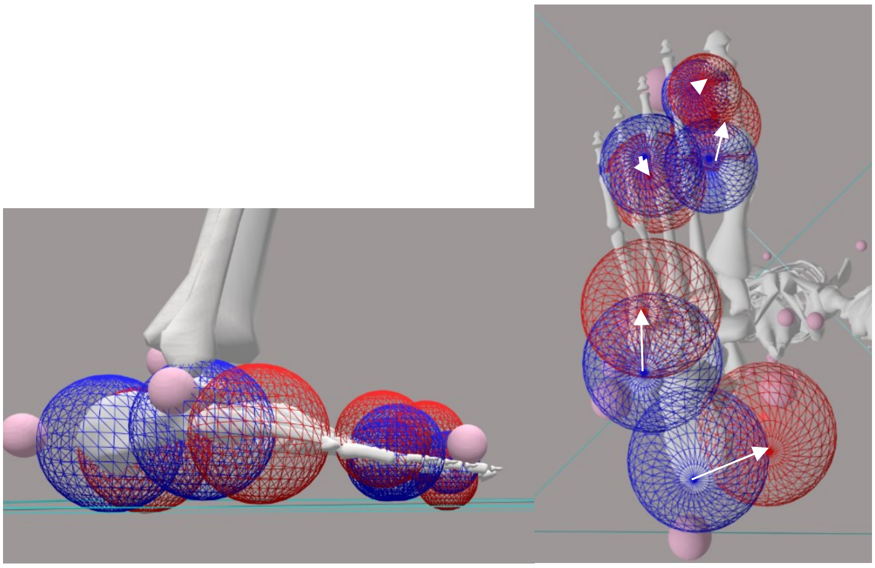


Figure 53: Spheres' position variation: in red, new spheres' location; in blue, start spheres' location. Lateral view on the left, plantar view on the right.

At the end to have a quantitative value of this displacement, with a custom MATLAB code, the difference between the ending ("optimal") and the starting position of the five spheres for each subject in the three directions, thus in the space, was calculated.

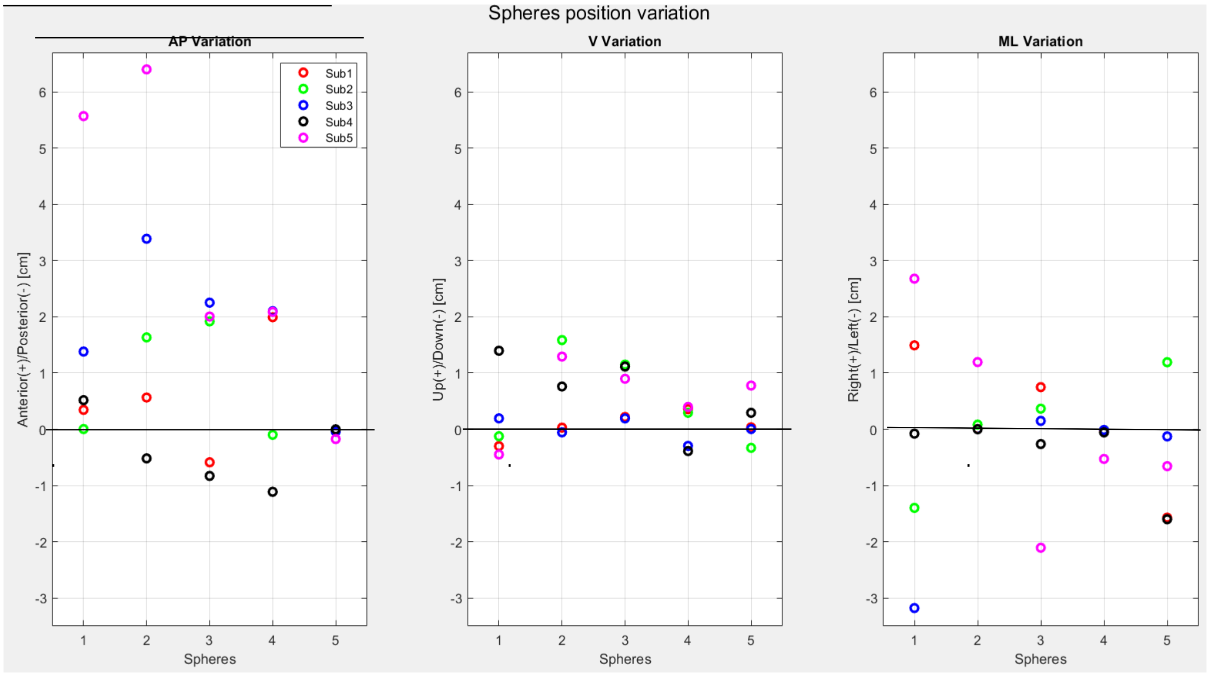


Figure 54: Spheres' position variation in all directions (the circle represents the difference between Final position [Tab.7] and Initial position [Tab.3]) for each subject after the application of the optimization problem of a random trial.

According to the custom boundaries settled, it can be seen that the spheres have a higher variation on the Anterior/Posterior axis (x-axis) with the exception of the fifth sphere of all the subject, while there is the lowest variation on vertical axis (y-axis).

5.4. Center of pressure results

As mentioned in the Methods chapter, for the extraction of GRFs components was created a custom function (extractGRFfromSpheres.m; further details are contained in the Appendix), in which are also present the center of pressure components and the ground reaction moment, obtained with ZMP method [67]. Then, after the application of the optimization problem, a new file in which contained the previously illustrated GRFs components (red curves of [Figg.49-50] is generated, but also the center of pressure associated with them. Also in this case, for the explanation of the center of pressure behavior, it was useful to consider the results of all the trials together and the results of the optimized and not optimized trials [Figg.55-56].

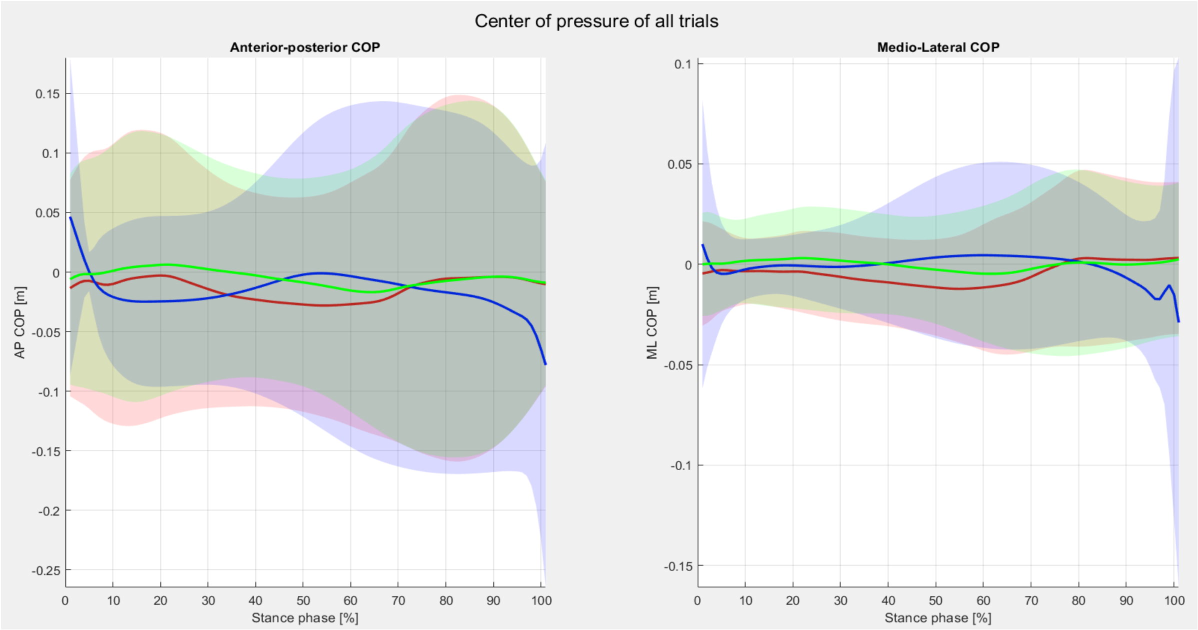


Figure 55: Center of pressure (COP) comparison: in red the COP coordinates estimated after the optimization while in blue the COP coordinates measured and in green the COP coordinates estimated before the optimization for all the fifteen trials.

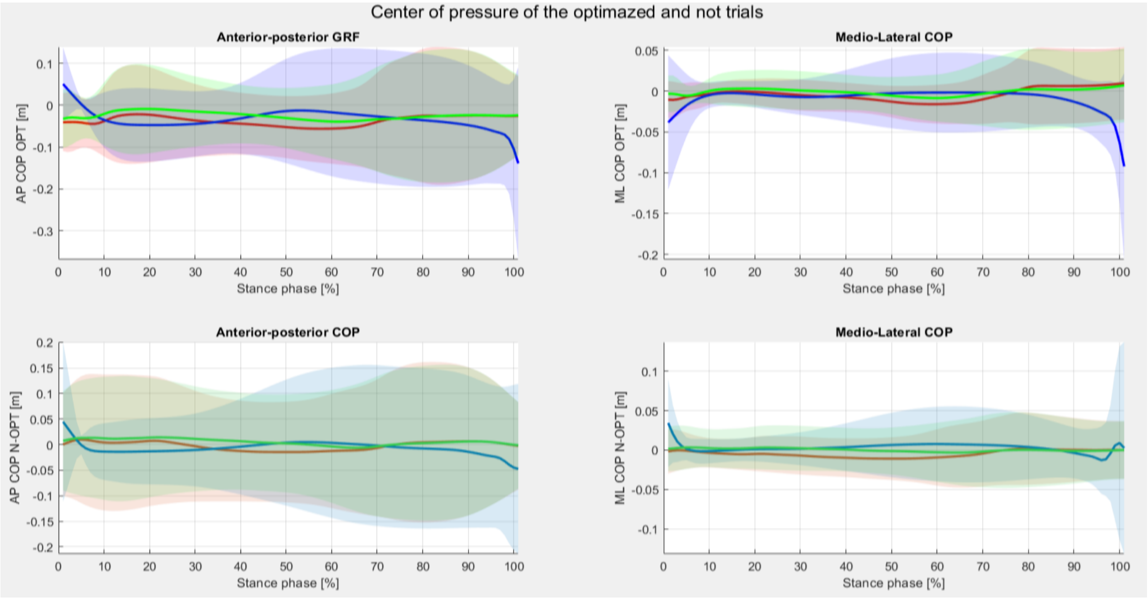


Figure 56: Center of pressure (COP) comparison: in red the COP coordinates estimated after the optimization while in blue the COP coordinates measured and in green the COP coordinates estimated before the optimization for the optimized and unseen trials.

In order to get quantitative values of the behavior previously illustrated, there were calculated the average (± standard deviation) of the root mean square error and the Pearson coefficient, that are showed in the following tables for both cases (optimized and not optimized trials) [Tabs. 8-9].

COP component	RMSE [m]	Pearson coefficient
Anterior-Posterior	0.08 ± 0.04	0.2 ± 0.19
Medio-Lateral	0.03 ± 0.01	0.3 ± 0.28

Table 8: Average value (± standard deviation) of COPs' Pearson coefficient (R^2) and root mean square error (RMSE) for all the trials.

COP component	RMSE [m]		Pearson coefficient	
	Optimal	Not Optimal	Optimal	Not optimal
Anterior-Posterior	0.08 ± 0.04	0.07 ± 0.04	0.26 ± 0.23	0.18 ± 0.17
Medio-Lateral	0.03 ± 0.02	0.03 ± 0.01	0.24 ± 0.26	0.33 ± 0.3

Table 9: Average value (± standard deviation) of COPs' Pearson coefficient (R^2) and root mean square error (RMSE) for the optimal and unseen trials.

The values illustrated are in line with the curves shown in [Figg.55-56]. In fact, COP coordinates have a low RMSE (0.08 ± 0.04 m and 0.03 ± 0.01 m for A/P and M/L coordinates respectively), but the small values of Pearson coefficient (0.2 ± 0.19 and 0.3 ± 0.28 for A/P and M/L coordinates respectively) justify the weak correlation between the curves, in both cases (all trials, optimized and not trials [Tab.9]).

5.5. Joint moments from inverse dynamics

Finally, [Fig.57] shows joint moments for the hip (first row), knee (second row), and ankle (third row) joints during overground walking for all the tasks, obtained from the OpenSim Inverse Dynamics Tool, as explained in Methods chapter, using Moco inverse kinematics approach on measured GRFs with force plates, (green curve) and on estimated GRFs extracted from the five spheres, with the point of application obtained via ZMP calculations (red curve) and using MoCap approach that used the kinematics and GRFs measured by the stereophotogrammetric system (blue curve).

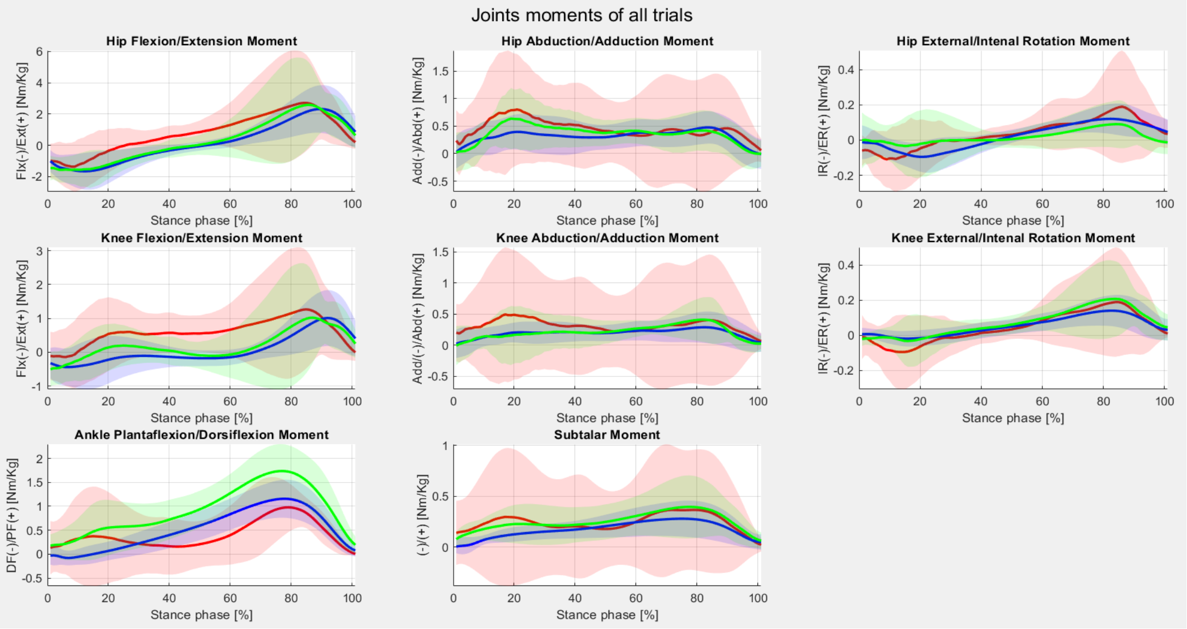


Figure 57: ID comparison: in red the moments obtained from estimated GRF with Moco kinematics; in green the moments obtained from measured GRF with Moco kinematics; in blue the moments obtained from measured GRF with MoCap kinematics; for all the fifteen trials.

For a better understanding of these graphs the average and standard deviation of the root mean square error and Pearson coefficient were calculated, and they are expressed in the following table [Tab.10].

Joint Moments	***Inverse Dynamics obtained with Moco kinematics***			
	Estimated GRFs		***Measured GRFs***	
	RMSE [Nm/kg]	**Pearson Coefficient**	**RMSE [Nm/kg]**	**Pearson Coefficient**
Hip Flx/Ext	1.2 ± 1.09	0.52 ± 0.36	0.47 ± 1.02	0.93 ± 0.1
Hip Add/Abd	0.6 ± 0.62	0.17 ± 0.14	0.17 ± 0.25	0.8 ± 0.32
Hip IR/ER	0.14 ± 0.1	0.46 ± 0.26	0.09 ± 0.07	0.83 ± 0.28
Knee Flx/Ext	0.91 ± 0.68	0.53 ± 0.18	0.41 ± 0.59	0.88 ± 0.25
Knee Add/Abd	0.49 ± 0.66	0.19 ± 0.22	0.14 ± 0.25	0.8 ± 0.24
Knee IR/ER	0.12 ± 0.11	0.53 ± 0.37	0.08 ± 0.08	0.83 ± 0.23
Ankle DF/PF	0.65 ± 0.39	0.31 ± 0.22	0.58 ± 0.35	0.88 ± 0.21
Subtalar	0.26 ± 0.41	0.29 ± 0.25	0.18 ± 0.09	0.58 ± 0.27

Table 10: Average value (± standard deviation) of inverse dynamics' Pearson coefficient (R^2) and root mean square error (RMSE) with respect to measured ID for all the fifteen trials.

These values reflect the graphs' behavior represented in the figure [Fig.57], especially for both estimated Adduction/Abduction hip and knee moments and Subtalar moment, where the standard deviation of Root mean square error, is proportionally higher than the others two with a value of mean (± standard deviation) of 0.6 ± 0.62 Nm/kg, 0.49 ± 0.66 Nm/kg, and 0.26 ± 0.41 Nm/kg respectively for the Estimated GRFs case. Moreover, for the Adduction/Abduction hip and knee moments, the Pearson coefficient is worse than the one of the other moments, with an average (± standard deviation) of 0.17 ± 0.14, and 0.19 ± 0.22, that justifies the weak correlation between the estimated joint moments (red) and the measured one (blue). Different behavior is for both Flexion/Extension and Internal/External rotation moments for either hip and knee, where the average (± standard deviation) is partially strong, with values of 0.52 ± 0.36, 0.46 ± 0.26, 0.53 ± 0.18, and 0.53 ± 0.37 respectively, that evidence the good correlation. It can be seen that there is the absence of strong correlation between the cited curves, for Estimated GRFs case.

Instead for the Measured GRFs case, the values are better compared to the previous case: the RMSE is very low with the exception (especially regarding the standard deviation) of hip and knee flexion/extension and ankle dorsiflexion/plantarflexion where the values of average (± standard deviation) are 0.47 ± 1.02 Nm/kg, 0.41 ± 0.59 Nm/kg, and 0.58 ± 0.35 Nm/kg respectively, but the same joint moment have the best Pearson coefficient where the values of average (± standard deviation) are 0.93 ± 0.1, 0.88 ± 0.25, and 0.88 ± 0.21, indicating a strong correlation with the full MoCap joint moments. All the values are indicated in [Tab.10].

As done before with the GRFs and COP results, also for the joint moments it was done the difference between all the trials and the optimized and not trials. Here, there are shown the graphs [Figg.58-59] and then the table [Tab.11] contained the average and standard deviation of root mean square error (RMSE) and Pearson coefficient (R^2).

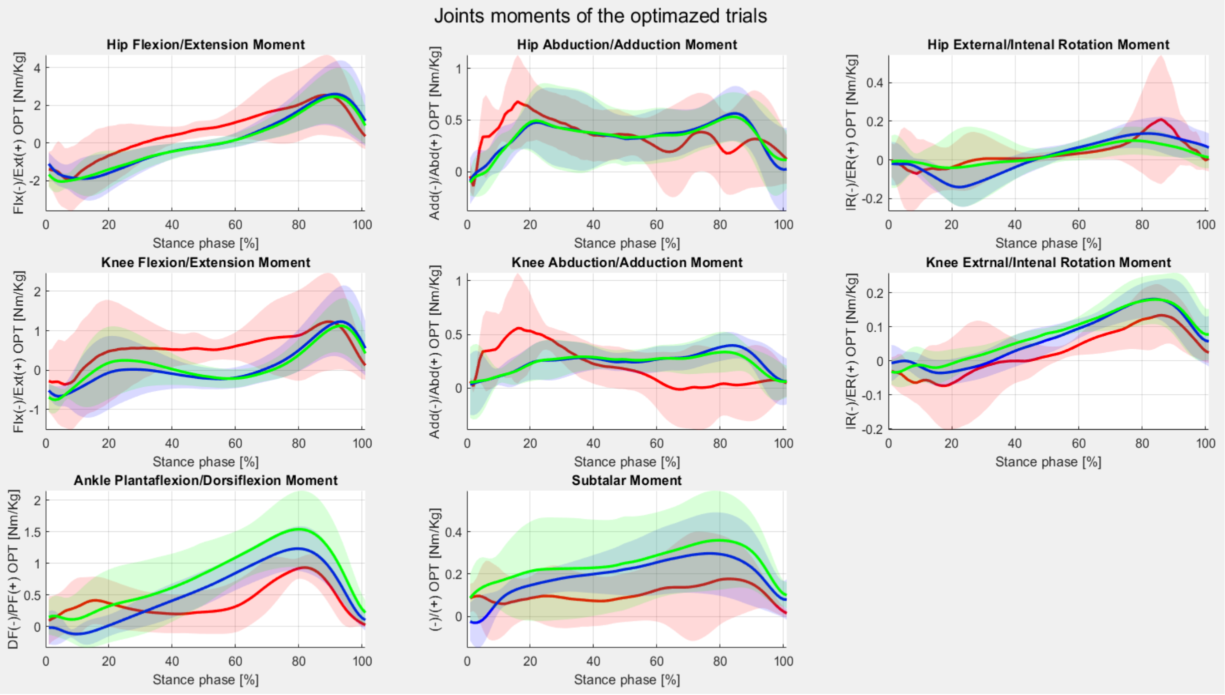


Figure 58: ID comparison: in red the joints moments obtained from estimated GRFs with Moco kinematics; in green the moments obtained from measured GRFs with Moco kinematics; in blue the moments obtained from measured GRFs with MoCap kinematics; for the five optimal trials.

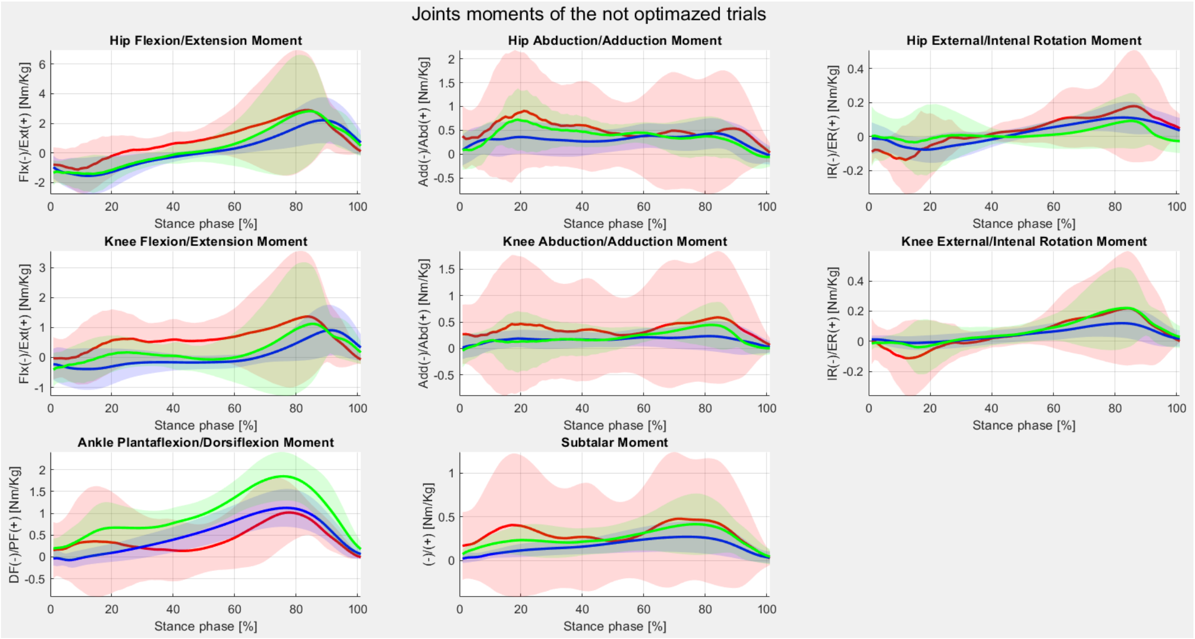


Figure 59: ID comparison: in red the joints moments obtained from estimated GRFs with Moco kinematics; in green the moments obtained from measured GRFs with Moco kinematics; in blue the moments obtained from measured GRFs with MoCap kinematics; for the ten unseen trials.

Joint Moments	***Inverse Dynamics obtained with Moco kinematics – Estimated GRFs***			
	RMSE [Nm/kg]		**Pearson coefficient**	
	Optimal	**Not optimal**	**Optimal**	**Not optimal**
Hip Flx/Ext	1.05 ± 0.22	1.28 ± 1.35	0.52 ± 0.44	0.52 ± 0.34
Hip Add/Abd	0.4 ± 0.14	0.69 ± 0.74	0.22 ± 0.18	0.15 ± 0.12
Hip IR/ER	0.16 ± 0.06	0.13 ± 0.11	0.5 ± 0.23	0.44 ± 0.28
Knee Flx/Ext	0.74 ± 0.22	0.99 ± 0.82	0.58 ± 0.15	0.5 ± 0.19
Knee Add/Abd	0.36 ± 0.1	0.56 ± 0.81	0.17 ± 0.23	0.2 ± 0.38
Knee IR/ER	0.08 ± 0.03	0.14 ± 0.13	0.65 ± 0.38	0.47 ± 0.19
Ankle DF/PF	0.51 ± 0.13	0.72 ± 0.46	0.33 ± 0.3	0.3 ± 1.36
Subtalar	0.15 ± 0.04	0.32 ± 0.5	0.25 ± 0.23	0.3 ± 0.27

Table 11: Inverse Dynamics obtained with Moco kinematics – Estimated GRFs: Average value (± standard deviation) of inverse dynamics' Pearson coefficient (R^2) and root mean square error (RMSE) with respect to measured ID for the optimal and unseen trials.

Joint Moments	***Inverse Dynamics obtained with Moco kinematics – Measured GRFs***			
	RMSE [Nm/kg]		**Pearson coefficient**	
	Optimal	**Not optimal**	**Optimal**	**Not optimal**
Hip Flx/Ext	0.23 ± 0.11	0.63 ± 1.25	0.96 ± 0.05	0.91 ± 0.11
Hip Add/Abd	0.07 ± 0.02	0.22 ± 0.3	0.92 ± 0.08	0.74 ± 0.38
Hip IR/ER	0.07 ± 0.09	0.1 ± 0.06	0.94 ± 0.06	0.77 ± 0.33
Knee Flx/Ext	0.19 ± 0.08	0.52 ± 0.7	0.97 ± 0.01	0.83 ± 0.3
Knee Add/Abd	0.05 ± 0.03	0.19 ± 0.3	0.82 ± 0.28	0.79 ± 0.24
Knee IR/ER	0.04 ± 0.02	0.09 ± 0.1	0.87 ± 0.07	0.8 ± 0.28
Ankle DF/PF	0.45 ± 0.37	0.64 ± 0.34	0.97 ± 0.03	0.84 ± 0.25
Subtalar	0.19 ± 0.07	0.17 ± 0.11	0.6 ± 0.32	0.57 ± 0.25

Table 12: Inverse Dynamics obtained with Moco kinematics – Measured GRFs: Average value (± standard deviation) of inverse dynamics' Pearson coefficient (R^2) and root mean square error (RMSE) with respect to measured ID for the optimal and unseen trials.

Regarding the five optimized trials results, which curves are illustrated in [Fig.56], the average (± standard deviation) of root mean square error, is lower for all joints moments, with the exception of Hip Internal/external rotation moment, which has a value of 0.16 ± 0.06 Nm/kg (against the previous result 0.14 ± 0.1 Nm/kg in [Tab.10]); the average (± standard deviation) of Pearson coefficient is higher than the condition in which there are considered all the fifteen trials, with the exception of Subtalar moment, with a value of 0.25 ± 0.23 (against the previous result 0.29 ± 0.25 in [Tab.10]).

Regarding the ten not optimized trials, as expected, there are higher values of mean (± standard deviation) of RMSE for all joints moments, with the exception of Hip Internal/external rotation moment, which has a value of 0.13 ± 0.11 Nm/kg (against the previous result 0.14 ± 0.1 Nm/kg in [Tab.10]);. the average (± standard deviation) of Pearson coefficient is slightly lower than the condition in which there are considered all the fifteen trials, with the exception of Subtalar moment, with a value of 0.3 ± 0.27 (against the previous result 0.29 ± 0.25 in [Tab.10]).

In [Tab.12] are reported the average (± standard deviation) of RMSE and Pearson coefficient for the 'mix' case (Measured GRFs and Moco IK). The behavior is the same as the one explained for [Tab.11] with the exception of hip internal/external rotation and subtalar moment regarding the average (± standard deviation) of RMSE, and of Knee adduction/abduction and subtalar joints regarding the average (± standard deviation) of Pearson coefficient, in which is the opposite: for hip IR/ER, the optimal case RMSE is lower than the unseen case (0.07 ± 0.09 Nm/kg optimal against 0.1 ± 0.06 Nm/kg unseen trial), vice versa for subtalar joint's RMSE (0.19 ± 0.07 Nm/kg optimal against 0.17 ± 0.11 Nm/kg unseen trial); for knee add/abd joint the optimal case Pearson coefficient is better than the unseen case (0.82 ± 0.28 optimal against 0.79 ± 0.24 unseen trial), vice versa for subtalar joint's Pearson coefficient (0.6 ± 0.32 optimal against 0.57 ± 0.25 unseen trial). All the comparisons just illustrated between optimal and unseen trials values in [Tab.11] and [Tab.12] have been done knowing that the RMSE and Pearson coefficient parameters for measured GRFs (and Moco IK) are better than estimated GRFs (and Moco IK) as previously written for [Tab.10].

6. DISCUSSION

The results illustrated in the previous chapter allow us to do some considerations about the developed optimization problem on the estimated ground reaction forces extracted from five contact geometries in a custom musculoskeletal model, during over ground walking gait.

The aim of this study was to develop an approach that substituted the gold standard, that is the use of force plates, in order to extract the ground reaction forces (and moments), and avoid the use of these instruments for the measurement of these forces. One of the reasons why there is the want to limit the use of the force plates, is that, despite the accuracy of the measures, their cost is extremely high.

In general, there are already such systems for the estimation of GRFs without the use of force plates [54], as mentioned in the third chapter of this thesis, that are the pressure insoles, that remain expensive but avoid another problem of the usage of force plates, that is the MoCap system instrument. In fact, besides this study representing an interesting alternative to pressure insoles, it is the beginning of a project that has, as a goal, the elimination of the entire use of motion capture systems environment. There are already research projects that have the aim of avoiding the use of such systems that allow the study of movements in a controlled environment but limiting the applications of motion analysis. One of them is the already validated OpenCap system [54], that allows the registration of human motion kinematics outside the motion laboratory (for example in an athletics field) thanks to common instruments (i.e., iPhone and/or iPad) and then upload the video on the software for the motion analysis. Another approach is the use of the inertial wearable sensors IMUs and OpenSim Moco tool coupled with models that have contact geometry for which the results are explained in the study [61]. The last one has been the approach followed in this thesis to reach the goal previously illustrated.

Regarding the results of this study, to allow a better understanding of them, they are going to be explained in the following specific paragraphs.

6.1. Discussion of Ground reaction forces results

As for the results obtained using the optimization framework ‘fminsearchbnd’ in MATLAB to extract the best ground reaction forces as possible, with the custom model and Moco kinematics as inputs, it can be stated that the joint angles data were consistent, as it can be seen in the Result chapter [Tab.4]. Since 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 and larger errors are not acceptable, joints angles, such as Dorsi/Plantar flexion and Inversion/Eversion of the ankle, and especially, the Flexion/Extension of the knee, have to be improved in MocoTrack approach, changing the weights in the cost functions, MocoTrackingGoal and/or MocoControlGoal, or changing the boundaries in MocoProblem section.

The aim of this thesis is to validate an optimization method in order to generate the best estimation of ground reaction forces based on a biomechanical model that has specific contact geometries suitable forces acting on them. The best estimation of ground forces are obtained if these forces are as close as possible to the ones extracted from force plates, that is the state of the art for this purpose. There are several studies in literature (for example the one showed in [73]), applied on different clinical and sportive research, with which the ground reactions forces extracted from the contact spheres could be compared, beside the ones recorded in the Movement and Posture Analysis Laboratory Leuven (MALL), at the Department of Movement Science of the KU Leuven, but in this chapter the GRFs obtained will be compared with these last explained forces, which results are illustrated in the Result chapter.

From a visual comparison, shown in [Fig.51], it is possible to establish how the approach presented provides improvements with respect to the initial GRFs (i.e., the ones before the optimization), but also some problems. The improvement is confirmed by the small root mean square error for the vertical component (0.23 ± 0.12 BW), that is followed by a good correlation (0.6 ± 0.29), instead limitations are confirmed by the weak correlation (0.1 ± 0.12) for the medio-lateral component and its behavior is strongly dependent on the trial, confirmed by the large standard deviation and by the worsening compared to the initial case; the anterior-posterior component has a similar behavior respect to the initial case, and it presents a strong correlation (0.82 ± 0.06). These considerations are in common with the optimized and not trials cases individually, knowing that in the first case, the results are slightly higher than the ones commented.

In order to justify these inappropriate values that bring such RMSE, the reason could be firstly attributed to the geometric configuration of the contact spheres and their mechanical properties during the development of the musculoskeletal problem used in this study and explained in the Methods chapter, that do not recreate a perfect gait motion. Secondly, the parameters considered in the MocoTrack algorithm of OpenSim Moco library bring some limitations because they do not ensure the perfect adhesion of the foot with the ground, even if the algorithm tries to get as close as possible to it. Then, these values could be justified by the several parameters (for example, the physical ones, illustrated in the methods chapter) of the contact spheres that were not managed by this optimization method, i.e., the optimization function chosen; by the choice of the boundaries, that should be more restrictive around the foot and respect its physiology, or by the several simplifications made on the musculoskeletal model. But the main problem of these results is the number of contact geometries utilized in the model: this causes a very high approximation that might have brought these results. Despite this modification may bring the optimization problem to be more difficult to manage, considering also the parameters mentioned before, thus, to be computationally slower, it could be a decisive way to improve the results.

Moreover, as it can be seen in [Fig.52], the optimization is strongly subject dependent. This happens because there is a high inter-variability between subjects and this reflects on the subject's foot size, height, weight, way to walk, how to put the foot on the ground, and the eventual pathology (even if it is not this study's case). On the contrary, the optimization is slightly trial dependent. For this reason, it was chosen to do the optimization on a random trial instead of considering the one that would have provided the smallest root mean square error.

Regarding all the considerations made 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. Suggestions of improvement are shown in the Conclusion chapter.

6.2. Discussion of Center of pressure results

The optimization method used in this study, thus the variation of the contact spheres' position variation, also strongly influences the estimation of the anterior/posterior and medio/lateral center of pressure (COP) components and the ground reaction moment. As explained in the Methods chapter, for their calculations, the zero-moment-point (ZMP) method [67] was employed, contained in the 'extractGRFfromSpheres.m' custom function code in MATLAB.

Since the optimization function is applied only on ground reaction forces, there is no evident improvement in the calculation of these values despite the obtained forces being used as input of the ZMP method. This method is a validate approach in order to obtain these points of action of these ground reaction forces, and the ground reaction moment, but it is evident that there are discrepancies, reported in [Tab.8-9] of the Result chapter, where high RMSE values and small Pearson coefficients (which led to a very weak correlation), can be seen for all the components.

According to D. Gupta et al., the ZMP method [67] elicits different simulation kinetics, and so COP coordinates, but besides this, the discrepancies are especially given by the not perfectly good approximations of the GRFs obtained from the optimization framework. Improvements of this evaluation have to be done firstly on the GRF problem optimization approach.

6.3. Discussion of Inverse Dynamics results

These considerations and results about ground reaction forces and center of pressure, reflect on the estimations of joint moments, obtained from the application of inverse dynamics mathematical equation reported in the methods chapter. Graphically, [Fig.57] clearly reports how the results from the estimated GRFs are highly variable: in particular, regarding the moments of hip and knee flexion/extension (hip Flx/Ext, knee Flx/Ext) and internal/external rotation (hip IR/ER, knee IR/ER), it is possible to find anomalous peaks at the beginning with strongly overestimated values at the end. This overestimation can be attributed to the inaccurate estimations of GRFs, GRMs, and COP components, which, when added together, produce such errors. Besides these inaccuracies, the evaluation of IK with the MocoTrack approach has also brought such ID (red curves in [Fig.57]). In fact, the green curves shown in the same figure, obtained with measured GRFs demonstrate how this IK influences the evaluation of ID. These curves showing a similar behavior compared to the full MoCap ID (blue curves in the same figure), justified with the high Pearson coefficient reported in [Tab.10], with the exception of Ankle plantarflexion/dorsiflexion, which overcomes the physiological positive range value. As confirmation of the graphs shown in [Fig.57], the Pearson coefficient and root mean square error (RMSE) were calculated and reported in [Tab.10] for full estimated ID (red curves), and mix ID (green curves) cases with respect to the full MoCap ID (blue curves).

Moreover, [Fig.57] shows an unexpected trend for hip and knee abduction/adduction moments (hip Add/Abd and knee Add/Abd) and subtalar moment as result of the influence of the unseen trials provided by [Fig.59], with the consequence of the high standard deviation, already mentioned in the Result chapter. This particular trend could be attributed to the fact that there are larger errors in both GRFs and COP cases, which are added together in the inverse dynamics' calculation. This in fact happens with less intensity in the optimized trials case, reported in [Fig.58]. As expected, [Tab.11] highlights the differences illustrated in [Figg.58-59], reporting mean (± std) RMSE and Pearson coefficient values of these joint moments, highlighted in the Result chapter. The same comments could be made also on green curves due to OpenSim Moco IK influence in ID calculations.

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 and the higher variability of hip and knee abduction/adduction moments (hip Add/Abd and knee Add/Abd) and subtalar moment present in [Fig.57], consequently increasing the correlation between reference data and data obtained by minimizing the errors present for the full estimated ID.

7. CONCLUSION

This thesis was developed with the aim of proposing a new optimization problem approach in order to estimate the best ground reaction forces, i.e., the closest to those measured with force plates. The estimation of these forces is based on a biomechanical model and data from an IMU-based approach [61]. 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].

One clinical application of this study could be that of finding a rehabilitation therapy solution for osteoarthritis, in particular knee osteoarthritis, for elder and/or lonely people that cannot reach the laboratory in which is present the MoCap system (since most of the time, they are in other cities given the low attendance due to high costs or small environments) or for anyone who cannot afford high expenses.

Osteoarthritis (OA) is the most frequent musculoskeletal disease and the most common degenerative joint disorder that affects one or several diarthrodial joints, including small joints (such as those in the hand) and large joints (such as the knee and hip joints) [74]. Clinically, the condition is characterized by the syndrome of joint pain, transient morning stiffness, crepitus on joint, and dysfunction caused by substantial joint degeneration, that lead to instability and physical disability, thus impairing quality of life. OA can be classified as primary (or idiopathic) and secondary (based on the attribution to recognized causative factors, such as trauma, surgery on the joint structures, abnormal joints at birth, etc.) [75]. Osteoarthritis is not a curable disease at present, as the mechanism by which it arises and progresses remains incompletely understood. Therefore, the major goals of OA treatment are pain control with minimal adverse effects, maintenance or improvement of joint mobility and function, and improved health related quality of life. Treatment should be tailored to each individual. Because no single therapy is adequate, the major clinical guidelines for disease management generally agree that therapy should involve a combination of nonpharmacologic and pharmacologic therapies. In fact, the therapeutic spectrum ranges from general measures to physiotherapy, orthopedic aids and orthoses, pharmacotherapy, and finally surgery and rehabilitation [75].

The results presented in the dedicated chapter (Results) clearly show how the approach developed in this project is promising: through a variation of a single parameter of spherical contact geometries with specific geometrical and physical characteristics ([Tab.2]), it was possible to get ground reaction forces closer to those measured in MoCap environment. The MATLAB optimization function [68], previously described in the Methods chapter, has allowed us to estimate these new GRFs, varying the spheres' position, considered as an input, within physiological ranges. In the result chapter, in addition to the main objective of this thesis, i.e., the comparison of estimated GRFs with the measure GRFs ([Fig.49], [Fig.50], [Tab.5], and [Tab.6]), were presented compared results of center of pressure coordinates ([Fig.55], [Fig.56], [Tab.8], and [Tab.9]) followed by the joints moments ([Fig.57], [Fig.58], [Fig.59], [Tab.10], and [Tab.11]).

Conclusions based on the results obtained and the discussion of these presented in the dedicated chapter (Discussion) can be drawn: the workflow developed represents a new method for the optimization of ground reaction forces. As mentioned in the previously cited chapter, further improvements to contact geometry should be made, despite the results obtained being promising. Many factors must be considered: for example, evaluating a different number of spheres, adding the variation of new parameters (such as radius or stiffness), or modifying the tolerances of 'fminsearchbnd' MATLAB function. In particular, the results estimated from the spheres still have some discrepancies from the values measured from the force platforms, despite the optimization; these discrepancies are directly responsible for errors in the subsequent calculation of center of pressure coordinates and joint moments. This shows how further improvements in this direction suitable for enhancing the reported estimated values are needed, since the gold standard devices (force plates) still appear to be the most reliable.

The proposed study constitutes a starting point for future developments and improvements that aim to validate this optimization method and that would bring such values of ground reaction forces to be reliable and clinically applicable. Beyond these, a 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, lead to significative changes in terms of proximity to measured force values by these estimated forces extracted from these geometries; 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 spheres that produces the most reliable optimization 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) making it less simplifying, with the ultimate goal of employing such a model in the study of movements of patients with diseases that impair normal walking, such as the already mentioned osteoarthritis, and neurological diseases.

Appendix: Code used

1. *Optimization_problem.m*

The script reported was used to get the best as possible ground reaction forces components (here the code is applied to the last subject, HS6, but the method is the same for the others with the exception of GRF measured range of interested, indicated with startL and endL, and GRF estimated, both before and after optimization, indicated with startP and endP).

```
%% GRF optimization
import org.opensim.modeling.*;
sub = 6;
walk = 0;
startL = 579;
endL = 1141;
startP = 11;
endP = 68;
BW = 68*9.81; %body masses of the subjects = [87.6 77.5 76.5 55.6 68];

lab = ['walk_',num2str(walk),'_grf.mot'];
modelOsim= ['C:\Users\ASUS\OneDrive - KU
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\model\HS',num2str(sub),'_G
RFextr_scaled.osim'];
motionFile = ['Moco_solution_MoCap_',num2str(walk)];

% Creation of GRF file
extractGRFfromSpheres_v2(lab,motionFile,modelOsim,sub,walk,false);
model = Model(modelOsim);
model.initSystem();

%% Ground reaction forces and moments MEASURED
[D_lab,C_lab,T_lab]= readMOTSTOTRCfiles_v1(['C:\Users\ASUS\OneDrive - KU
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\experimentalData\walk_',nu
m2str(walk),'\'], ['walk_',num2str(walk),'_grf.mot']);

if sub == 1 || (sub == 2 && walk == 7)
    grf_lab = TimeNormalize_v1([T_lab.ground_force_vx(startL:10:endL),
T_lab.ground_force_vy(startL:10:endL),
T_lab.ground_force_vz(startL:10:endL)],101);
    grm_lab = TimeNormalize_v1(T_lab.ground_torque_y(startL:10:endL),101);
    cop_x_lab = TimeNormalize_v1(T_lab.ground_force_px(startL:10:endL),101);
    cop_z_lab = TimeNormalize_v1(T_lab.ground_force_pz(startL:10:endL),101);
elseif sub == 4
    if walk == 5
        grf_lab = TimeNormalize_v1([T_lab.ground_force1_vx(startL:10:endL),
T_lab.ground_force1_vy(startL:10:endL),
T_lab.ground_force1_vz(startL:10:endL)],101);
        grm_lab = TimeNormalize_v1(T_lab.ground_torque1_y(startL:10:endL),101);
        cop_x_lab = TimeNormalize_v1(T_lab.ground_force1_px(startL:10:endL),101);
        cop_z_lab = TimeNormalize_v1(T_lab.ground_force1_pz(startL:10:endL),101);
    else
        D_lab(:,20) = -D_lab(:,20);
        D_lab(:,22) = -D_lab(:,22);
        generateMotFile_v3(D_lab,C_lab,[['C:\Users\ASUS\OneDrive - KU
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\experimentalData\walk_',nu
m2str(walk),'\'],['walk_',num2str(walk),'_grfAP.mot']])
```

```
        [D_lab1,C_lab1,T_lab1]= readMOTSTOTRCfiles_v1(['C:\Users\ASUS\OneDrive - 
KU 
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\experimentalData\walk_',nu
m2str(walk),'\'], ['walk_',num2str(walk),'_grfAP.mot']);
        grf_lab = TimeNormalize_v1([T_lab1.ground_force3_vx(startL:10:endL), 
T_lab1.ground_force3_vy(startL:10:endL), 
T_lab1.ground_force3_vz(startL:10:endL)],101);
        grm_lab = TimeNormalize_v1(T_lab1.ground_torque3_y(startL:10:endL),101);
        cop_x_lab = TimeNormalize_v1(T_lab.ground_force3_px(startL:10:endL),101);
        cop_z_lab = TimeNormalize_v1(T_lab.ground_force3_pz(startL:10:endL),101);
    end
elseif sub == 2
    if walk == 9 || walk == 12
        grf_lab = TimeNormalize_v1([T_lab.ground_force1_vx(startL:10:endL), 
T_lab.ground_force1_vy(startL:10:endL), -
T_lab.ground_force1_vz(startL:10:endL)],101);
        grm_lab = TimeNormalize_v1(T_lab.ground_torque1_y(startL:10:endL),101);
        cop_x_lab = TimeNormalize_v1(T_lab.ground_force1_px(startL:10:endL),101);
        cop_z_lab = TimeNormalize_v1(T_lab.ground_force1_pz(startL:10:endL),101);
    else
        grf_lab = TimeNormalize_v1([T_lab.ground_force_vx(startL:10:endL), 
T_lab.ground_force_vy(startL:10:endL), 
T_lab.ground_force_vz(startL:10:endL)],101);
        grm_lab = TimeNormalize_v1(T_lab.ground_torque_y(startL:10:endL),101);
        cop_x_lab = TimeNormalize_v1(T_lab.ground_force_px(startL:10:endL),101);
        cop_z_lab = TimeNormalize_v1(T_lab.ground_force_pz(startL:10:endL),101);
    end
else % sub == 3 || sub == 6
    D_lab(:,2) = -D_lab(:,2);
    D_lab(:,4) = -D_lab(:,4);
    generateMotFile_v3(D_lab,C_lab,[['C:\Users\ASUS\OneDrive - KU 
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\experimentalData\walk_',nu
m2str(walk),'\'],['walk_',num2str(walk),'_grfAP.mot']])
    [D_lab1,C_lab1,T_lab1]= readMOTSTOTRCfiles_v1(['C:\Users\ASUS\OneDrive - KU 
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\experimentalData\walk_',nu
m2str(walk),'\'], ['walk_',num2str(walk),'_grfAP.mot']);
    grf_lab = TimeNormalize_v1([T_lab1.ground_force1_vx(startL:10:endL), 
T_lab1.ground_force1_vy(startL:10:endL), 
T_lab1.ground_force1_vz(startL:10:endL)],101);
    grm_lab = TimeNormalize_v1(T_lab1.ground_torque1_y(startL:10:endL),101);
    cop_x_lab = TimeNormalize_v1(T_lab.ground_force1_px(startL:10:endL),101);
    cop_z_lab = TimeNormalize_v1(T_lab.ground_force1_pz(startL:10:endL),101);
end

%% INITIAL Ground reaction forces ESTIMATED
[D_MoCo,C_MoCo,T_MoCo]= readMOTSTOTRCfiles_v1(['C:\Users\ASUS\OneDrive - KU 
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\modelledData\walk_',num2st
r(walk),'\'], ['walk_',num2str(walk),'_GRFMOCO.mot']);

if sub == 1 || (sub == 2 && walk == 7)
    grf_MoCo_start = TimeNormalize_v1([T_MoCo.ground_force_vx(startP:endP), 
T_MoCo.ground_force_vy(startP:endP), T_MoCo.ground_force_vz(startP:endP)],101);
    grm_MoCo_start = T_MoCo.ground_torque_y(startP:endP);
    cop_x_MoCo_start = TimeNormalize_v1(T_MoCo.ground_force_px(startP:endP),101);
    cop_z_MoCo_start = TimeNormalize_v1(T_MoCo.ground_force_pz(startP:endP),101);
    if walk == 11
        grf_MoCo_start = 
TimeNormalize_v1([T_MoCo.('1_ground_force_vx')(startP:endP), 
T_MoCo.('1_ground_force_vy')(startP:endP), 
T_MoCo.('1_ground_force_vz')(startP:endP)],101);
        grm_MoCo_start = T_MoCo.('1_ground_torque_y')(startP:endP);
        cop_x_MoCo_start =
TimeNormalize_v1(T_MoCo.('1_ground_force_px')(startP:endP),101);
        cop_z_MoCo_start =
TimeNormalize_v1(T_MoCo.('1_ground_force_pz')(startP:endP),101);
    end
elseif sub == 4 && walk == 5
    grf_MoCo_start = TimeNormalize_v1([T_MoCo.ground_force1_vx(startP:endP),
T_MoCo.ground_force1_vy(startP:endP), T_MoCo.ground_force1_vz(startP:endP)],101);
    grm_MoCo_start = T_MoCo.ground_torque1_y(startP:endP);
    cop_x_MoCo_start = TimeNormalize_v1(T_MoCo.ground_force1_px(startP:endP),101);
    cop_z_MoCo_start = TimeNormalize_v1(T_MoCo.ground_force1_pz(startP:endP),101);
else
    grf_MoCo_start = TimeNormalize_v1([T_MoCo.ground_force1_vx(startP:endP),
T_MoCo.ground_force1_vy(startP:endP), T_MoCo.ground_force1_vz(startP:endP)],101);
    grm_MoCo_start = T_MoCo.ground_torque1_y(startP:endP);
    cop_x_MoCo_start = TimeNormalize_v1(T_MoCo.ground_force1_px(startP:endP),101);
    cop_z_MoCo_start = TimeNormalize_v1(T_MoCo.ground_force1_pz(startP:endP),101);
end

%% Spheres Position
% Initial position
sph = model.get_ContactGeometrySet();
S1 = org.opensim.modeling.ContactSphere.safeDownCast(sph.get('s1_r'));
S2 = org.opensim.modeling.ContactSphere.safeDownCast(sph.get('s2_r'));
S3 = org.opensim.modeling.ContactSphere.safeDownCast(sph.get('s3_r'));
S4 = org.opensim.modeling.ContactSphere.safeDownCast(sph.get('s4_r'));
S5 = org.opensim.modeling.ContactSphere.safeDownCast(sph.get('s5_r'));
p1 = S1.get_location();
p2 = S2.get_location();
p3 = S3.get_location();
p4 = S4.get_location();
p5 = S5.get_location();

x = [p1.get(0),p2.get(0),p3.get(0),p4.get(0),p5.get(0)];
y = [p1.get(1),p2.get(1),p3.get(1),p4.get(1),p5.get(1)];
z = [p1.get(2),p2.get(2),p3.get(2),p4.get(2),p5.get(2)];
p0 = [x;y;z]';

%% Radius
r1 = S1.getRadius();
r2 = S2.getRadius();
r3 = S3.getRadius();
r4 = S4.getRadius();
r5 = S5.getRadius();

%% FMINSEARCHBND FOR THE OPTIMIZATION
grf_lab = reshape(grf_lab,[1,size(grf_lab,1)*3]);
opt_fun = @(p) rms(grf_lab - estim_v3(model,p,sub,walk,startP,endP,6,false));
options = optimset('Display','iter','MaxFunEvals', 1500, 'MaxIter', 1500,
'TolFun', 1e-30, 'TolX', 1e-30);

LB = [p1.get(0), p1.get(1) - (r1/2), p1.get(2) - (r1);
      p2.get(0) - (2*r2), p2.get(1) - (r2/2), p2.get(2);
      p3.get(0) - (r3), p3.get(1) - (r3/2), p3.get(2) - (r3);
      p4.get(0) - (2*r4), p4.get(1) - (r4/2), p4.get(2) - (r4);
      p5.get(0) - (2*r5), p5.get(1) - (r5/2), p5.get(2) - (r5)];
UB = [p1.get(0) + (2*r1), p1.get(1) + (r1/2), p1.get(2) + (r1);
      p2.get(0) + (2*r2), p2.get(1) + (r2/2), p2.get(2) + (2*r2);
      p3.get(0) + (r3), p3.get(1) + (r3/2), p3.get(2) + (r3);
      p4.get(0) + (r4), p4.get(1) + (r4/2), p4.get(2);
      p5.get(0), p5.get(1) + (r5/2), p5.get(2) + (r5)];
```

```
[p_opt,fval] = fminsearchbnd(opt_fun, p0, LB, UB, options);
optimizationError = fval;

vec1_new = Vec3(p_opt(1,1),p_opt(1,2),p_opt(1,3)); %new quantity
vec2_new = Vec3(p_opt(2,1),p_opt(2,2),p_opt(2,3));
vec3_new = Vec3(p_opt(3,1),p_opt(3,2),p_opt(3,3));
vec4_new = Vec3(p_opt(4,1),p_opt(4,2),p_opt(4,3));
vec5_new = Vec3(p_opt(5,1),p_opt(5,2),p_opt(5,3));
S1.set_location(vec1_new);
S2.set_location(vec2_new);
S3.set_location(vec3_new);
S4.set_location(vec4_new);
S5.set_location(vec5_new);

%% Creation of the new OPTIMAL model
path = ['C:\Users\ASUS\OneDrive - KU
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\model\'];
osim = ['HS',num2str(sub),'_optimization_walk',num2str(walk),'.osim'];
model.initSystem();
model.print([path, osim]);
modelOsim_new = ['C:\Users\ASUS\OneDrive - KU
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\model\',osim];
model_new = Model(modelOsim_new);
model_new.initSystem();

% Creation of GRF file
extractGRFfromSpheres_v2(lab,motionFile,modelOsim_new,sub,walk,true);

[D_MoCo_new,C_MoCo_new,T_MoCo_new]= readMOTSTOTRCfiles_v1(['C:\Users\ASUS\OneDrive
- KU
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\modelledData\walk_',num2st
r(walk),'\'], ['walk_',num2str(walk),'_GRFMOCO_optim.mot']);

%% OPTIMIZED grfm MEASURED (RESPECT TO p_opt)
if sub == 1 || (sub == 2 && walk == 7)
    grf_MoCo_opt = [T_MoCo_new.ground_force_vx(startP:endP),
T_MoCo_new.ground_force_vy(startP:endP), T_MoCo_new.ground_force_vz(startP:endP)];
    grm_MoCo_opt = T_MoCo_new.ground_torque_y(startP:endP);
    cop_x_MoCo_opt =
TimeNormalize_v1(T_MoCo_new.ground_force_px(startP:endP),101);
    cop_z_MoCo_opt =
TimeNormalize_v1(T_MoCo_new.ground_force_pz(startP:endP),101);
    if walk == 11
        grf_MoCo_opt =
TimeNormalize_v1([T_MoCo_new.('1_ground_force_vx')(startP:endP),
T_MoCo_new.('1_ground_force_vy')(startP:endP),
T_MoCo_new.('1_ground_force_vz')(startP:endP)],101);
        grm_MoCo_opt = T_MoCo_new.('1_ground_torque_y')(startP:endP);
        cop_x_MoCo_opt =
TimeNormalize_v1(T_MoCo_new.('1_ground_force_px')(startP:endP),101);
        cop_z_MoCo_opt =
TimeNormalize_v1(T_MoCo_new.('1_ground_force_pz')(startP:endP),101);
    end
elseif sub == 4 && walk == 5
    grf_MoCo_opt = TimeNormalize_v1([T_MoCo_new.ground_force1_vx(startP:endP),
T_MoCo_new.ground_force1_vy(startP:endP),
T_MoCo_new.ground_force1_vz(startP:endP)],101);
    grm_MoCo_opt = T_MoCo_new.ground_torque1_y(startP:endP);
    cop_x_MoCo_opt =
TimeNormalize_v1(T_MoCo_new.ground_force1_px(startP:endP),101);
```

```
    cop_z_MoCo_opt = 
TimeNormalize_v1(T_MoCo_new.ground_force1_pz(startP:endP),101);
else
    grf_MoCo_opt = [T_MoCo_new.ground_force1_vx(startP:endP), 
T_MoCo_new.ground_force1_vy(startP:endP), 
T_MoCo_new.ground_force1_vz(startP:endP)];
    grm_MoCo_opt = T_MoCo_new.ground_torque1_y(startP:endP);
    cop_x_MoCo_opt = 
TimeNormalize_v1(T_MoCo_new.ground_force1_px(startP:endP),101);
    cop_z_MoCo_opt = 
TimeNormalize_v1(T_MoCo_new.ground_force1_pz(startP:endP),101);
end
%% FILTERING
fc = 20; % cut off frequency [Hz]
fs = 101; % sample frequency [Hz]
[magnitude,freq] = butter(2,fc/(fs/2)); %2nd order Butterworth low-pass filter 
(20Hz)

if sub == 1 || (sub == 2 && walk == 7)
    grf_lab = TimeNormalize_v1([T_lab.ground_force_vx(startL:10:endL), 
T_lab.ground_force_vy(startL:10:endL), 
T_lab.ground_force_vz(startL:10:endL)],101);
    grm_lab = TimeNormalize_v1(T_lab.ground_torque_y(startL:10:endL),101);
    cop_x_lab = TimeNormalize_v1(T_lab.ground_force_px(startL:10:endL),101);
    cop_z_lab = TimeNormalize_v1(T_lab.ground_force_pz(startL:10:endL),101);
elseif sub == 4
    if walk == 5
        grf_lab = TimeNormalize_v1([T_lab.ground_force1_vx(startL:10:endL), 
T_lab.ground_force1_vy(startL:10:endL), 
T_lab.ground_force1_vz(startL:10:endL)],101);
        grm_lab = TimeNormalize_v1(T_lab.ground_torque1_y(startL:10:endL),101);
        cop_x_lab = TimeNormalize_v1(T_lab.ground_force1_px(startL:10:endL),101);
        cop_z_lab = TimeNormalize_v1(T_lab.ground_force1_pz(startL:10:endL),101);
    else
        grf_lab = TimeNormalize_v1([T_lab1.ground_force3_vx(startL:10:endL), 
T_lab1.ground_force3_vy(startL:10:endL), 
T_lab1.ground_force3_vz(startL:10:endL)],101);
        grm_lab = TimeNormalize_v1(T_lab1.ground_torque3_y(startL:10:endL),101);
        cop_x_lab = TimeNormalize_v1(T_lab.ground_force3_px(startL:10:endL),101);
        cop_z_lab = TimeNormalize_v1(T_lab.ground_force3_pz(startL:10:endL),101);
    end
elseif sub == 2
    if walk == 9 || walk == 12
        grf_lab = TimeNormalize_v1([T_lab.ground_force1_vx(startL:10:endL), 
T_lab.ground_force1_vy(startL:10:endL), 
T_lab.ground_force1_vz(startL:10:endL)],101);
        grm_lab = TimeNormalize_v1(T_lab.ground_torque1_y(startL:10:endL),101);
        cop_x_lab = TimeNormalize_v1(T_lab.ground_force1_px(startL:10:endL),101);
        cop_z_lab = TimeNormalize_v1(T_lab.ground_force1_pz(startL:10:endL),101);
    else
        grf_lab = TimeNormalize_v1([T_lab.ground_force_vx(startL:10:endL), 
T_lab.ground_force_vy(startL:10:endL), 
T_lab.ground_force_vz(startL:10:endL)],101);
        grm_lab = TimeNormalize_v1(T_lab.ground_torque_y(startL:10:endL),101);
        cop_x_lab = TimeNormalize_v1(T_lab.ground_force_px(startL:10:endL),101);
        cop_z_lab = TimeNormalize_v1(T_lab.ground_force_pz(startL:10:endL),101);
    end
else % sub == 3 || sub == 6
    grf_lab = TimeNormalize_v1([T_lab1.ground_force1_vx(startL:10:endL), 
T_lab1.ground_force1_vy(startL:10:endL), 
T_lab1.ground_force1_vz(startL:10:endL)],101);
```

```
    grm_lab = TimeNormalize_v1(T_lab1.ground_torque1_y(startL:10:endL),101);
    cop_x_lab = TimeNormalize_v1(T_lab.ground_force1_px(startL:10:endL),101);
    cop_z_lab = TimeNormalize_v1(T_lab.ground_force1_pz(startL:10:endL),101);
end

grf_lab = filtfilt(magnitude,freq,grf_lab);
grf_MoCo_opt = filtfilt(magnitude,freq,grf_MoCo_opt);
grf_MoCo_opt = TimeNormalize_v1(grf_MoCo_opt,101);
grf_MoCo_opt_time =
TimeNormalize_v1(grf_MoCo_opt,length(T_MoCo.time(startP:endP)));
grm_MoCo_opt = filtfilt(magnitude,freq,grm_MoCo_opt);
```

2. *Evaluation of estimated GRFs (estim_v3 function)*

This function was used to get the extraction of the estimated GRFs at each iteration of the optimization problem.

```
function GRF_Torq_new =
estim_v3(model,p,sub,walk,startingpoint,endingpoint,component)
% p should be in the form p = [x,y,z] where x = [x.sphere1, x.sphere2, ...,
x.sphere5]' and y, z are the same.
% sub is a scalar representing the subject that you want to analyze
% walk is a scalar representing the walking data that you want to analyze
% component is a scalar, and it is equal to:
% 0 --> Fx
% 1 --> Fy
% 2 --> Fz
% 3 --> COPx
% 4 --> COPz
% 5 --> My
% 6 --> Fx, Fy, Fz
% 7 --> COPx, COPz, My
% 8 --> 6 + 7

import org.opensim.modeling.*;

sph = model.get_ContactGeometrySet();
S1 = sph.get('s1_r');
S2 = sph.get('s2_r');
S3 = sph.get('s3_r');
S4 = sph.get('s4_r');
S5 = sph.get('s5_r');

vec1 = Vec3(p(1,1),p(1,2),p(1,3));
vec2 = Vec3(p(2,1),p(2,2),p(2,3));
vec3 = Vec3(p(3,1),p(3,2),p(3,3));
vec4 = Vec3(p(4,1),p(4,2),p(4,3));
vec5 = Vec3(p(5,1),p(5,2),p(5,3));
S1.set_location(vec1);
S2.set_location(vec2);
S3.set_location(vec3);
S4.set_location(vec4);
S5.set_location(vec5);

%% Creation of the new model
path = ['C:\Users\ASUS\OneDrive - KU
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\model\'];
osim = ['HS',num2str(sub),'_scaled_NewLocSph_walk',num2str(walk),'.osim'];
```

```
model.initSystem();
model.print([path, osim]);
modelOsim_new = ['C:\Users\ASUS\OneDrive - KU
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\model\',osim];

lab = ['walk_',num2str(walk),'_grf.mot'];
motionFile = ['Moco_solution_MoCap_',num2str(walk)];
extractGRFfromSpheres_v2(lab,motionFile,modelOsim_new,sub,walk,true);
[~,~,T_MoCo_new]= readMOTSTOTRCfiles_v1(['C:\Users\ASUS\OneDrive - KU
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\modelledData\walk_',num2st
r(walk),'\'], ['walk_',num2str(walk),'_GRFMOCO_optim.mot']);
if sub == 1 || (sub == 2 && walk == 7)
    if walk == 11
        GRF_Torq_new =
TimeNormalize_v1([T_MoCo_new.('1_ground_force_vx')(startingpoint:endingpoint),
T_MoCo_new.('1_ground_force_vy')(startingpoint:endingpoint),
T_MoCo_new.('1_ground_force_vz')(startingpoint:endingpoint),
T_MoCo_new.('1_ground_force_px')(startingpoint:endingpoint),
T_MoCo_new.('1_ground_force_pz')(startingpoint:endingpoint),
T_MoCo_new.('1_ground_torque_y')(startingpoint:endingpoint)],101);
    else
        GRF_Torq_new =
TimeNormalize_v1([T_MoCo_new.ground_force_vx(startingpoint:endingpoint),
T_MoCo_new.ground_force_vy(startingpoint:endingpoint),
T_MoCo_new.ground_force_vz(startingpoint:endingpoint),
T_MoCo_new.ground_force_px(startingpoint:endingpoint),
T_MoCo_new.ground_force_pz(startingpoint:endingpoint),
T_MoCo_new.ground_torque_y(startingpoint:endingpoint)],101);
    end
elseif sub == 4 && walk == 5
    GRF_Torq_new =
TimeNormalize_v1([T_MoCo_new.ground_force1_vx(startingpoint:endingpoint),
T_MoCo_new.ground_force1_vy(startingpoint:endingpoint),
T_MoCo_new.ground_force1_vz(startingpoint:endingpoint),
T_MoCo_new.ground_force1_px(startingpoint:endingpoint),
T_MoCo_new.ground_force1_pz(startingpoint:endingpoint),
T_MoCo_new.ground_torque1_y(startingpoint:endingpoint)],101);
else
    GRF_Torq_new =
TimeNormalize_v1([T_MoCo_new.ground_force1_vx(startingpoint:endingpoint),
T_MoCo_new.ground_force1_vy(startingpoint:endingpoint),
T_MoCo_new.ground_force1_vz(startingpoint:endingpoint),
T_MoCo_new.ground_force1_px(startingpoint:endingpoint),
T_MoCo_new.ground_force1_pz(startingpoint:endingpoint),
T_MoCo_new.ground_torque1_y(startingpoint:endingpoint)],101);
end

%% Creating GRF
switch component
    case 0
        GRF_Torq_new = GRF_Torq_new(:,1)';
    case 1
        GRF_Torq_new = GRF_Torq_new(:,2)';
    case 2
        GRF_Torq_new = GRF_Torq_new(:,3)';
    case 3
        GRF_Torq_new = GRF_Torq_new(:,4)';
    case 4
        GRF_Torq_new = GRF_Torq_new(:,5)';
    case 5
        GRF_Torq_new = GRF_Torq_new(:,6)';
    case 6
        GRF_Torq_new = GRF_Torq_new(:,1:3);
        GRF_Torq_new = reshape(GRF_Torq_new,[1,size(GRF_Torq_new,1)*3]);
    case 7
        GRF_Torq_new = GRF_Torq_new(:,4:6);
        GRF_Torq_new = reshape(GRF_Torq_new,[1,size(GRF_Torq_new,1)*3]);
    case 8
        GRF_Torq_new = reshape(GRF_Torq_new,[1,size(GRF_Torq_new,1)*6]);
end
end
```

3. *GRFM and COP extraction (extractGRFfromSpheres_v2 function)*

```
function extractGRFfromSpheres_v2(grfFile,motionFile,osimFile,sub,walk,new)
%INPUTS
%grfFile: from 'lab measurement' needed only for labels
%motionFile: IK
%osimFile: model path
%sub and trial (walk)
%new = Boolean; it indicates if it is of the new optimized model or not
%OUTPUT
%file cointaining the GRFs

ik_path = ['C:\Users\ASUS\OneDrive - KU
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\modelledData\walk_',num2st
r(walk),'\'];
grfpath = ['C:\Users\ASUS\OneDrive - KU
Leuven\Cristina_erasmus\testData\TBM_H0',num2str(sub),'\experimentalData\walk_',nu
m2str(walk),'\'];
[data, names, ~] = readMOTSTOTRCfiles_v1(ik_path, strcat(motionFile,'.sto'));
[data_grf, names_grf, ~] = readMOTSTOTRCfiles_v1(grfpath,grfFile);

% Add fiber length name in the sto file
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) = [];
statesFile = strcat([motionFile '_wFiber.sto']);
generateMotFile_v3(data_new, names_new, [ik_path, statesFile])
newStatesTraj=StatesTrajectory();
import org.opensim.modeling.*

model = Model(osimFile);
model.initSystem();

ikStatesTraj=newStatesTraj.createFromStatesStorage(model, [ik_path, statesFile]);
time = data_new(:,1);

SF1 = model.getForceSet().get('SmoothSphereHalfSpaceForce_s1_r');
SF2 = model.getForceSet().get('SmoothSphereHalfSpaceForce_s2_r');
SF3 = model.getForceSet().get('SmoothSphereHalfSpaceForce_s3_r');
SF4 = model.getForceSet().get('SmoothSphereHalfSpaceForce_s4_r');
SF5 = model.getForceSet().get('SmoothSphereHalfSpaceForce_s5_r');

for i = 0:ikStatesTraj.getSize()-1
    mState = ikStatesTraj.get(i);
    model.realizeVelocity(mState);
    clear f_sphere1
    clear f_sphere2
    clear f_sphere3
    clear f_sphere4
    clear f_sphere5

    f_sphere1= SF1.getRecordValues(mState);
    for j = 0:f_sphere1.getSize()-1
        Forces_sphere1(i+1,j+1) = f_sphere1.get(j);
    end
    f_sphere2= SF2.getRecordValues(mState);
    for j = 0:f_sphere2.getSize()-1
        Forces_sphere2(i+1,j+1) = f_sphere2.get(j);
    end
    f_sphere3= SF3.getRecordValues(mState);
    for j = 0:f_sphere3.getSize()-1
        Forces_sphere3(i+1,j+1) = f_sphere3.get(j);
    end
    f_sphere4= SF4.getRecordValues(mState);
    for j = 0:f_sphere4.getSize()-1
        Forces_sphere4(i+1,j+1) = f_sphere4.get(j);
    end
    f_sphere5= SF5.getRecordValues(mState);
    for j = 0:f_sphere5.getSize()-1
        Forces_sphere5(i+1,j+1) = f_sphere5.get(j);
    end
end

Forces_All=Forces_sphere1+Forces_sphere2+Forces_sphere3+Forces_sphere4+Forces_sphe
re5;
ForcesAll_Time=time;

oa_GRF_data=zeros(size(Forces_All,1),7);
oa_GRF_data(:,1)=ForcesAll_Time;
oa_GRF_data(:,2:7)=-1*Forces_All(:,7:12);

oa_GRF.nr=size(Forces_All,1);
oa_GRF.nc=size(data_grf,2);
oa_GRF.labels=names_grf;
oa_GRF.data=zeros(oa_GRF.nr,oa_GRF.nc);
oa_GRF.data(:,1)=oa_GRF_data(:,1);

oa_GRF.data(:,2:4)= oa_GRF_data(:,2:4);
if sub == 1 || sub == 2 || (sub == 4 && walk == 5)
    if walk == 11
        oa_GRF.data(:,2) = oa_GRF.data(:,2);
        oa_GRF.data(:,4) = -oa_GRF.data(:,4);
    else
        oa_GRF.data(:,2) = oa_GRF.data(:,2);
        oa_GRF.data(:,4) = oa_GRF.data(:,4);
    end
elseif sub == 3 || (sub == 4 && walk == 2)
    oa_GRF.data(:,2) = -oa_GRF.data(:,2);
    oa_GRF.data(:,4) = oa_GRF.data(:,4);
else
    oa_GRF.data(:,2) = -oa_GRF.data(:,2);
    oa_GRF.data(:,4) = -oa_GRF.data(:,4);
end

oa_GRF.data(:,5)=oa_GRF_data(:,7)./oa_GRF_data(:,3);
oa_GRF.data(:,7)=-1*(oa_GRF_data(:,5)./oa_GRF_data(:,3));
if sub == 1
    oa_GRF.data(:,15)=oa_GRF_data(:,6) + (oa_GRF.data(:,5).*oa_GRF_data(:,4)) -
(oa_GRF.data(:,7).*oa_GRF_data(:,2));
else
    oa_GRF.data(:,9)=oa_GRF_data(:,6) + (oa_GRF.data(:,5).*oa_GRF_data(:,4)) -
(oa_GRF.data(:,7).*oa_GRF_data(:,2));
end

for i=1:oa_GRF.nr
    for j=1:oa_GRF.nc
        if isnan(oa_GRF.data(i,j))
            oa_GRF.data(i,j)=0;
        end
    end
end
oa_GRF.inDeg='no';

if new == false
    generateMotFile_v3(oa_GRF.data, oa_GRF.labels, [ik_path, 'walk_' num2str(walk)
'_GRFMOCO.mot'])
else
    generateMotFile_v3(oa_GRF.data, oa_GRF.labels, [ik_path, 'walk_' num2str(walk)
'_GRFMOCO_optim.mot'])
end
end
```

Bibliography

[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).

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

[4] J. Tu, K. Inthavong, and G. Ahmadi. “Reconstruction of the Human Airways”. In: Computational Fluid and Particle Dynamics in the Human Respiratory System, pp 45-71

[5] https://geekymedics.com/anatomical-movements-of-the-human-body/

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

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

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

[9] DOI:10.1109/ICOEI51242.2021.9452951

[10] MD. Akhtaruzzaman, A. A. Shafie, and MD. R. Khan “Gait analysis: System, Technologies, and importance”. In: Journal of Mechanics in Medicine and Biology, Vol.16, No. 07

[11] 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.

[12] N. Goldfarb, A. Lewis, A. Tacescu, and G.S. Fisher, “Open source Vicon Toolkit for motion capture and Gait Analysis”. In Computer Methods and Programs in Biomedicine, Volume 212 (2021)

[13] 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.

[14] T. Ebara, R. Azuma, N. Shoji, T. Matsukawa, Y. Yamada, T. Akiyama, T. Kurihara, and S. Yamada. “Reliability of Smartphone-Based Gait Measurements for Quantification of

Physical Activity/inactivity Levels.” In: Journal of Occupational Health 59.6 (2017), pp. 506–512.

[15] N. Petrone. “Lectures Notes of Sport engineering and rehabilitation devices course”. In: University of Padua (2022).

[16] G. Pascucci, “L'elettromiografia: principio di funzionamento e applicazioni in ambito sportivo.”, University of Bologna (2016).

[17] I. Rodríguez, L. Gila-Useros, A. Malanda, “Motor Unit Action Potential Duration: Measurement and Significance”. In: Advances in Clinical Neurophysiology (2012).

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

[19] P. Sturm, “Pinhole Camera Model”, In: Computer Vision (2016), pp. 610-613.

[20] C. Valenza, “Applicazione della modellazione muscolo-scheletrica nella patologia del piede diabetico: confronto tra tre diversi modelli muscoloscheletrici 3D del piede”, University of Padua (2021).

[21] R.B. Davis, S. Ounpuu, D.J. Tyburski, J.R. Gage, “A gait analysis data collection and reduction technique,” Hum. Mov. Sci., vol. 10, pp. 575-587, (1991).

[22] A. Capozzo, F. Catani, U. Della Croce, A. Leardini, “Position, and orientation in space of bones during movement: anatomical frame definition and determination”. In: Clinical Biomechanics (1995); 10(4): 171-8

[23] A. Leardini, Z. Sawacha, G. Paolini, S. Ingrosso, R. Nativo, M. G. Benedetti. “A new anatomically based protocol for gait analysis in children.” (2007). In: Gait and posture, Vol. 26, pp. 560-571.

[24] G. Barberato, “Confronto tra stereofotogrammetria ed analisi video nella ricostruzione della cinematica articolare durante il cammino”, University of Padua (2022)

[25] F. Yazdani, M. Razeghi et al., “The influence of foot hyperpronation on pelvic biomechanics during stance phase of the gait: A biomechanical simulation study”. In Sage Journals, Vol 232, n°7.

[26] A. Ancillao. Modern Functional Evaluation Methods for Muscle Strength and Gait Analysis. (2018).

[27] N. de Oliveira, L.L. Menegaldo et al., “Inverse Kinematics of Push-Up Exercise Using Joint Coordinate System”. In: Journal of Medical Imaging and Health Informatics, Vol 4(1.) (2014), pp. 83-91.

[28] D. DeMers, and K. Kreutz-Delgado, “Inverse Kinematics of Dextrous Manipulators”. In: Neural Systems for Robotics, Chapter 4, 1997, pp. 75-116.

[29] A. Lunia, et al., “Modeling, Motion Planning, and Control of Manipulators and Mobile Robots”, Chapter 3, 2021.

[30] Y. Jung, M. Jung, K. Lee, S. Koo, “Ground reaction force estimation using an insole-type pressure mat and joint kinematics during walking”. In: Journal of Biomechanics, Vol.47(11), (2014), pp. 2693-2699.

[31] B. Meadows, R. Bowers, and E. Owen, “Biomechanics of the Hip, Knee, and Ankle”. In: Atlas of Orthoses and Assistive Devices (5th Edition), 2019, pp. 207-215.

[32] C. Dickin, K.A. Faust, H. Wang, and J. Frame, “The acute effects of whole-body vibration on gait parameters in adults with cerebral palsy”. In: Journal of Musculoskeletal & Neuronal Interactions, Vol 13(1), (2013), pp. 19-26.

[33] D.A. Winter, The biomechanics and motor control of human gait: normal, elderly, and pathological. University of Waterloo press, Ontario (1991).

[34] H.C. Seon, H.S. Lee, M.S. Koo, and S.W. Park, “Correlation between Gait Speed and Velocity of Center of Pressure Progression during Stance Phase in the Older Adults with Cognitive Decline: A Pilot Study”. In: Journal of the Korean Society of Physical Medicine, Vol 15(4), (2020), pp. 67-74.

[35] H. Faber, A.J. van Soest, and D.A. Kistemarker, “Inverse dynamics of mechanical multibody systems: An improved algorithm that ensures consistency between kinematics and external forces”. In: Plos One, Vol 13(9), (2018).

[36] http://www.clinicalgaitanalysis.com/teach-in/inverse-dynamics.html

[37] J.P. Holden, J.A. Orsini, K.L. Siegel, T.M. Kepple, L.H. Gerber, S.J. Stanhope, "Surface movement errors in shank kinematics and knee kinetics during gait". In: Gait & Posture 5 (3) (1997) pp. 217-227.

[38] P. Flores, "Contact Mechanics for Multibody Dynamics". In: The Fifth Joint International Conference on Multibody System Dynamics, (2018).

[39] https://en.wikipedia.org/wiki/Contact_mechanics

[40] T. de Melo Malaquias, "A combined multibody and plantar pressure approach to estimate and predict foot kinematics applied to 3D-printed insoles", 2019.

[41] Hertzian contact theory - About Tribology (tribonet.org)

[42] D. Younesian, A. Hosseinkhani, H. Askari, and E. Esmailzadeh, "Elastic and viscoelastic foundations: a review on linear and nonlinear vibration modeling and applications", In: Nonlinear Dynamics, Vol 97, (2019), pp. 853-895.

[43] K. H. Hunt, and F. R. E. Crossley, "Coefficient of Restitution Interpreted as Damping in Vibroimpact", In: Journal of Applied Mechanics, Vol 42(2), 1975, pp. 440-445.

[44] M.R. Rodrigues da Silva, F. Marques, M. Tavares da Silva, P. Flores, "A compendium of contact force models inspired by Hunt and Crossley's cornerstone work". In: Mechanism and Machine Theory, Vol 167, 2022.

[45] N.H. Kim, "Finite Element Analysis for Contact Problems". In: Introduction to Nonlinear Finite Element Analysis (2015), pp. 367-426.

[46] A. Gefen, M. Megido-Ravid, Y. Itzchak, and M. Arcan, "Biomechanical analysis of the three-dimensional foot structure during gait: a basic tool for clinical applications". In: Journal of Biomechanical Engineering 122, December 2000 (2000), pp. 630–639.

[47] A. Gefen, "Plantar soft tissue loading under the medial metatarsals in the standing diabetic foot". In: Medical Engineering and Physics 25, 6 (2003), pp. 491–499.

[48] https://medium.com/@oscarsc/four-ways-to-create-a-mesh-for-a-sphere-d7956b825db4

[49] https://www.sciencedirect.com/topics/engineering/rigid-body-model

[50] M. de Oliveira Nunes, L. Menegaldo et al., “Inverse Kinematics of Push-Up Exercise Using Joint Coordinate System”. In: Journal of Medical Imaging and Health Informatics, Vol 4(1.) (2014), pp. 83-91.

[51] https://www.anybodytech.com/software/anybodymodelingsystem/#featurelist

[52] Y. Kim, J. Kim, K. M. Lee, S. Koo, “The increase of joint contact forces in foot joints with simulated subtalar fusion in healthy subjects”, In: Gait & Posture, (2019), pp.27-32.

[53] Welcome to OpenSim - OpenSim Documentation - Global Site (stanford.edu)

[54] K. Tanghe, M. Afschrift, I. Jonkers, F. De Groote, et al., “A probabilistic method to estimate gait kinetics in the absence of ground reaction force measurements”. In: Journal of Biomechanics, Vol 96, 2019.

[55] S. D. Uhlrich, A. Falisse, Ł. Kidziński, J. Muccini, et al., “OpenCap: 3D human movement dynamics from smartphone videos”. In: bioRxiv (2022).

[56] https://www.ibm.com/topics/machine-learning

[57] S. E. Oh, A. Choi, and J. H. Mun, “Prediction of ground reaction forces during gait based on kinematics and a neural network model”. In: Journal of Biomechanics, Vol 46(14), 2013, pp. 2372-2380.

[58] C. Chatterjee, L. Wen, K. Vinsen, “Using deep learning to localize gravitational wave sources”, 2019.

[59] Site Home - Vicon Documentation

[60] A. Falisse, M. Afschrift, F. De Groote, “Modeling toes contributes to realistic stance knee mechanics in three-dimensional predictive simulations of walking”. In: Plos One Vol 17(1), 2022.

[61] L. Bottini, “Ground reaction forces estimation using IMU-based kinematics and OpenSim Moco”. University of Padua, 2023.

[62] https://simtk-confluence.stanford.edu:8443/display/OpenSim/How+to+Use+the+Scale+Tool

[63] K. J. Middleton, P. M. Mills, B. C. Elliott, and J. A. Alderson, “The association between lower limb biomechanics and ball release speed in cricket fast bowlers: a comparison of high-performance and amateur competitors”. In: Sports Biomechanics (2016), pp. 357-369.

[64]https://simtk-confluence.stanford.edu:8443/display/OpenSim/How+to+Use+the+IK+Tool

[65] https://simtk-confluence.stanford.edu:8443/display/OpenSim/How+Inverse+Kinematics

[66] C. L. Dembia, N. A. Bianco, A. Falisse, J. L. Hicks, and S. L. Delp, "OpenSim Moco: Musculoskeletal optimal control". In: Plus computational biology, 2020.

[67] D. Gupta, C. J. Donnelly , and J. A. Reinbolt, "Physics-Based Guidelines for Accepting Reasonable Dynamic Simulations of Movement". In: IEEE Transactions on biomedical engineering, Vol 69(3), 2022, pp. 1194-1201.

[68]https://www.mathworks.com/help/releases/R2021b/optim/ug/unconstrained-nonlinear-optimization-algorithms.html

[69] https://www.jmp.com/it_it/statistics-knowledge-portal/what-is-correlation.html

[70] How to Use the Inverse Dynamics Tool - OpenSim Documentation - Global Site (stanford.edu)

[71] How Inverse Dynamics Works - OpenSim Documentation - Global Site (stanford.edu)

[72] https://mathworks.com/products/matlab.html

[73] 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, 2018.

[74] A. Barg et al., "Ankle osteoarthritis: etiology, diagnostics, and classification". In: Foot and ankle clinics (2013), pp. 411-426.

[75] J. Martel-Pelletier, A. J. Barr, F.M. Cucittini et al., "Osteoarthritis". In: Nature reviews: Disease primers, 2016.

[76] 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.

List of Figures

List of Tables