Treball de Fi de Grau

Enginyeria en Tecnologies Industrials

Optimal Control Prediction of Dynamically Consistent Walking Motions

MEMÒRIA

Autor: Roger Pallarès López
Directors: Josep Maria Font Llagunes, Míriam Febrer Nafría

Convocatòria: Juny 2017

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

ABSTRACT

The main objective of this bachelor thesis in Industrial Technology Engineering is to predict dynamically consistent walking motions from kinematic and dynamic measurements obtained at the UPC Biomechanics Laboratory. A healthy gait cycle is captured and foot-ground contact forces are measured. Then, in order to acquire the new motions, optimal control techniques are applied.

The human body is modeled with a multibody system formed by rigid bodies. Concretely, a two-dimensional simplified skeletal model focused on the lower extremity is used in this work. It is formed by a total of 12 rigid bodies (trunk, pelvis and leg segments) and it has 10 degrees of freedom. The inverse dynamic analysis is performed using OpenSim, a free software tool developed by Stanford University that is widely used by the scientific community.

The optimal control algorithm to obtain dynamically consistent walking motions from experimental data is implemented in MATLAB. Moreover, the software used to solve the optimal control problem is GPOPS-II, a general-purpose MATLAB-based software for solving multiple-phase optimal control problems, developed by the University of Florida. Parameters of GPOPS-II are changed to study the influence on the solution. Then, different formulations are analyzed to assess convergence and similarity between the new motion and the captured one.

During this report, all the processes involved in the analysis and the related theory are detailed, as well as the methodology used. Theoretical background is presented and complemented with examples of other works. The skeletal model used is described in detail. The process to export and obtain the experimental kinematics and dynamics using OpenSim is explained step by step. Optimal control theory and GPOPS-II working environment, which are employed as the tool to predict new motions, are also explained. And finally, results are presented and discussed.

This project is considered an initial study of optimal control techniques to predict human motion. Thereby, it allows to understand these techniques and gain knowledge about how they can be used in order to be applied, in the future, in more complex models.

CONTENTS

ETSEIB

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation

Over recent decades, multibody system dynamics techniques have been used as a tool for analyzing and assessing the human motion. Biomechanics is the area of research that applies these techniques, and it can be defined as *"the study of the structure and function of biological systems by means of the methods of mechanics"* [23]. It embraces a wide range of applications, such as the investigation of animals motion, the analysis of sports performance, or even the study of the mechano-biological behavior of tissues and cells. Notwithstanding, this project is focused on the analysis of human motion and the rehabilitation engineering. Concretely, on the rehabilitation of incomplete spinal cord injured (SCI) patients.

Incomplete spinal cord injured subjects suffer from a several reduction of movement and sensation below the injury level. Gait restoration among SCI patients is considered to be a high priority [12]. Thus, the use of active orthoses and exoskeletons can highly improve the sensation and control of the affected limbs. These devices help to exploit the remaining capacities of the patient and foster the active participation of the subject in its rehabilitation process.

Related to this topic, a lower limb robotic orthosis for incomplete SCI patients has been developed in the Biomechanical Engineering Group (BIOMEC) of the Department of Mechanical Engineering at the Barcelona School of Industrial Engineering (ETSEIB) and in the Biomedical Engineering Research Centre (CREB), at Universitat Politècnica de Catalunya (UPC). This orthosis has been developed in the frame of two national projects (DPI2009-13438-C03-03 and DPI2012-38331-C03-02), in collaboration with researchers from University of La Coruña (UDC) and University of Extremadura (UEX).

The present work, entitled "Optimal Control Prediction of Dynamically Consistent Walking Motions", corresponds to the Bachelor Thesis of Industrial Technology Engineering. Moreover, this thesis is framed within a current national project entitled "Low-cost motor-FES hybrid orthosis for the gait of spinal cord injured subjects and simulation methods to support the design and adaptation" (DPI2015-65959-C3-2-R), coordinated by the same groups UPC-UDC-UEX. One of the goals of this national project is to develope a computational tool, which allows to predict gait motions, for the purpose of customizing and optimizing the design of the orthosis in order to ease the adaptation process for the patient. The study developed in this project is aligned with this objective.

1.2. Project objectives

The general objective of this project is to predict a 2D dynamically consistent healthy gait motion applying optimal control techniques and using experimental measurements. This general goal is subdivided into the following specific objectives:

- Capturing a healthy walking motion of a subject and measuring the foot-ground contact forces.
- Performing the inverse kinematic and dynamic analyses of a 2D skeletal model using the OpenSim software.
- Learning optimal control theory and formulations that are used for human motion prediction.
- Getting acquainted with the optimal control software GPOPS-II.
- Studying the effect of different parameter changes on the solution and analyzing different optimal control formulations.
- Implementing a final formulation to predict a 2D dynamically consistent walking motion based on experimental data.

1.3. Project scope

This project is an initial study of prediction using GPOPS-II to obtain a dynamically consistent and physiologically feasible walking motion. Concretely, a whole 2D gait cycle of a healthy subject is taken in the Laboratory of Biomechanics at ETSEIB. From this gait cycle, the joint angles and torques are calculated with OpenSim and are used as input information for GPOPS-II. Lastly, new joint angles and joint torques that correspond to a dynamically consistent motion are found through GPOPS-II.

Because of the timing and the circumstances that involve this study, some limitations must be kept in mind: First, owing to the difficulty for GPOPS-II to converge and find an optimal solution for a human motion problem, a 2D model is studied as a first stage. Secondly, the experimental data obtained from the motion capture is always tracked in the prediction, and a foot-ground contact model is not used. Finally, muscle forces are not found because of its complexity, conversely, joint torques that are the result of these muscle forces spanning a specific joint are determined.

To sum up, the scope of this project is the prediction of a 2D healthy gait cycle by analyzing different optimal control formulations. It is therefore an initial study of a future work, in which the 3D walking motion of an SCI subject, wearing orthoses, will be predicted to customize the orthosis design to that subject.

2. THEORETICAL BACKGROUND

Theoretical background related to the project (biomechanics of human motion, dynamics analysis and human motion prediction) is presented and explained. Furthermore, all the information is complemented with examples of other works.

2.1. Biomechanics of human motion

2.1.1. Reference planes

Three reference planes are used to describe the motion of the human body. They are called anatomical planes and separate the body in different sections (see Figure 2.1). The sagittal or lateral plane is a vertical plane that divides the body into a left and a right part. The frontal or coronal plane divides the body into a front section and a back section (anterior and posterior sections). Finally, the transverse or axial plane is a horizontal plane that splits the body into an upper portion and a lower portion [25].

The sagittal plane is probably the most important one, where much of the movement takes place. As an example, in [35] different control techniques are applied to the forward dynamic simulation using a biomechanical 2D model in the sagittal plane. However, when an impaired or pathological gait is analyzed, another plane could also give relevant information. For example, the coronal plane in the case of bilateral hip pain analysis [39], or the transverse plane when studying lower limb joint kinematics in patients after anterior cruciate ligament reconstruction [8].

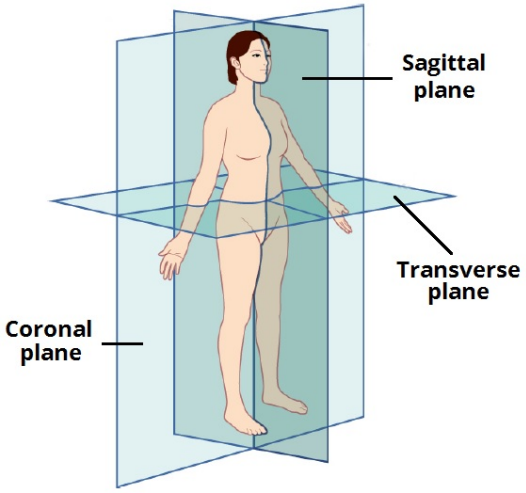


Figure 2.1 Reference planes of the human body. Extracted from [25].

2.1.2. Gait cycle

The human gait is a bipedal locomotion method in which there are alternate movements of different limbs of the body. It is characterized by a sequence of single and double supports, that is, one foot is in contact with the ground while the other advances to another support contact (single support), or both feet are in contact with the ground while the weight is transferred from one limb to the other (double support). A gait cycle is defined as a sequence of these events between two equivalent instants during locomotion. For one limb, a gait cycle can be divided in two phases: the stance phase in which the foot is on the ground, and the swing phase in which the same foot is no longer in contact with the ground and the leg is swinging. In a healthy gait cycle the stance phase approximately corresponds to the 60% of the cycle, whereas the swing phase represents the remaining 40% of the cycle [39]. Figure 2.2 represents one gait cycle for the right leg.

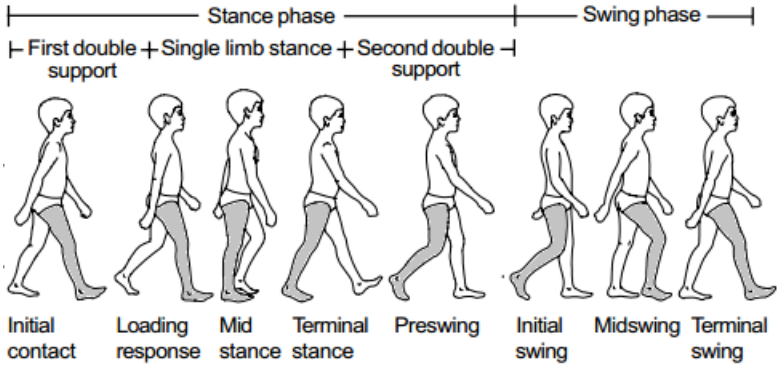


Figure 2.2 Divisions and subdivisions of one gait cycle for the right leg. Adapted from [39].

Considering that healthy people initiate ground contact with their heel, it is common to define a gait cycle for a limb from heel strike to heel strike (HE - HE). Conversely, in other cases, the gait cycle is comprised from toe off to toe off (TO - TO), *i.e.*, when the foot looses contact with the ground.

The gait cycle can also be determined by the term stride. One stride can be measured as the length between the heels from one heel strike to the next heel strike of the same limb. On the other hand, the step is the distance between the heel strike of one limb and the next heel strike of the other limb [39]. Two steps make one stride. The representation of these two terms can be seen in Figure 2.3.

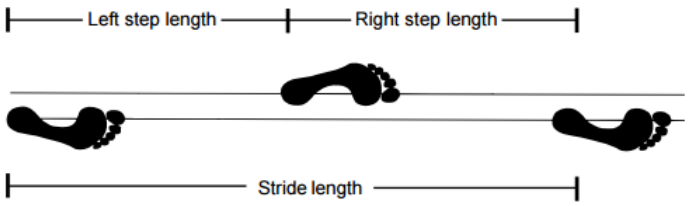


Figure 2.3 Representation of step and stride. Adapted from [39].

2.1.3. Multibody system modelling

The kinematic and dynamic study of human motion is based on rigid body dynamics. The skeleton is defined as a multibody system, formed by rigid bodies that represent the bones. Each rigid body must include the necessary information for the analysis: the physical characteristics of each segment (mass, length, tensor of inertia and center of mass), called body segment parameters (BSP). The model includes joints that define and restrict the relative movement between the bodies. These joints are usually considered as ideal joints. Two different group of joints must be defined in the model: Internal joints, habitually modeled as rotations of one or more degrees of freedom; and external joints, for example the foot-ground contact in the case of human walking. If there is no fixed body to the ground, it is very common to consider that one body of the system, usually the pelvis, is linked to the reference frame with a six degrees of freedom joint (three for translation and three for rotation) [6, 1].

Skeletal models do not usually have the same number of bodies as the human body. It is common to group a set of bones with reduced relative movement as a single body. Furthermore, depending on the study, some parts of the model can be simplified. For example, the action of throwing the ball in basketball in a seated position is studied in [7], using an accurate model for the arms. However, since the subject is seated, there is no need to model the lower limbs in such case. In another study, where the human gait is analyzed, the head, arms and trunk are modeled as a single body called HAT, and the legs are more precisely modeled [4]. Also, the model can be two or three dimensional. Figure 2.4 shows an example of a 3D model in which head, arms and trunk are modeled in a single HAT body.

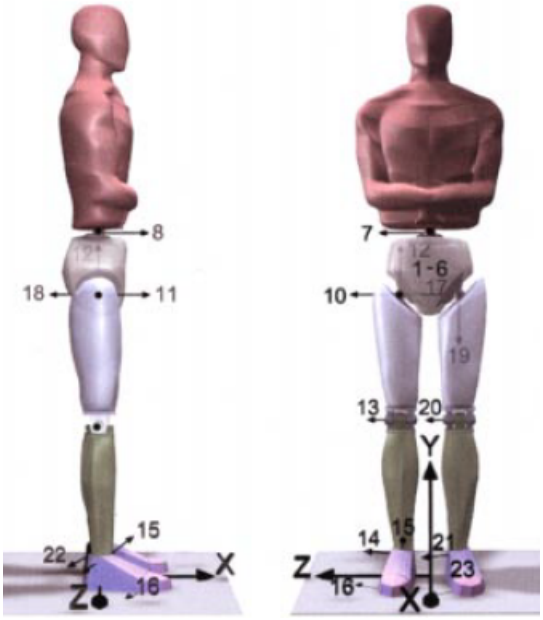


Figure 2.4 3D legs and HAT model with a total of 23 generalized coordinates [4]. Sagittal plane view on the left and frontal plane view on the right.

The configuration of the different bodies of the model is made through the definition of a group of generalized coordinates. Mathematically, the joints can be modeled by constraint equations that introduce kinematic relations between these coordinates. The allowed joint rotations and the six degrees of freedom of one body with respect to the ground are usually chosen as independent coordinates. On the other hand, the positions of the center of mass of the rigid bodies are normally chosen as dependent variables [33].

With reference to forces and moments, actuators are directly associated to all the generalized coordinates, thereby, each degree of freedom is controlled by an actuator. Torque values from actuators that control relative rotations between two bodies are called joint torques. Regarding to the body directly linked to the ground (reference frame), six actuators (three lineal and three angular) are defined. They are commonly known as residual forces and moments, or residual wrench. Lastly, external forces and torques from the ground are introduced to the model by defining the body and point where they are acting [6].

In order to find the motion equations of the model, analytical methods and vector methods can be used. The Newton-Euler equations method is a vector method in which the motion of each rigid body (time derivatives of linear and angular momenta) is related to the sum of external forces and torques acting to this body. An example of this formulation can be found in [1]. On the other hand, the Lagrange equations method is an analytical method in which the equations of motion are systematically obtained from the kinetic and potential energies of the multibody system and the involved generalized forces. This second method is used to find the equations of motion in [35].

2.2. Dynamics analysis

As it has been explained in Section 2.1.3, the study of human motion is carried out through a dynamic analysis of a multibody system. The dynamic analysis allows to obtain the desired information of the studied motion, such as muscle forces [21], rehabilitation outcomes [30], knowledge of a specific motion [7], etc. According to the goal of the study, two different types of dynamic analyses can be performed: inverse dynamic analysis (IDA) or forward dynamic analysis (FDA).

2.2.1. Inverse dynamic analysis

In inverse dynamic analysis, motion is known and is introduced as input information in order to compute forces and torques. It is the most common technique to analyze human motion: net joint torques that have to be applied at joints to perform the studied motion are obtained. Since motion variables are known, forces and torques are calculated by replacing coordinates, velocities and accelerations to the equations of motion. Thus, this method is algebraic and no integration is needed.

Experimental motion is usually acquired from marker trajectories. Therefore, a preprocess called inverse kinematics (IK) has to be done in order to get the coordinates and their time derivatives (this preprocess will be detailed in Section 5.2). Moreover, additional experimental data is usually introduced as input information: ground reaction forces (GRF) and body segment parameters (BSP) of the model (see Figure 2.5). An example of this formulation can be found in [6] where a healthy gait is analyzed and joint torques are calculated from experimental motion. Another example is the study of the action of throwing the ball in basketball [7].

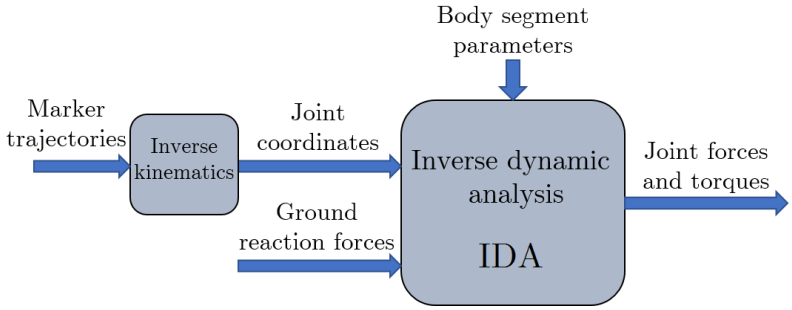


Figure 2.5 Diagram of an inverse dynamic analysis.

2.2.2. Forward dynamic analysis

In forward dynamic analysis, forces and torques are known and are the input information to compute the resulting motion (see Figure 2.6). It is used to determine how a mechanical system will evolve when a group of forces and torques are applied. Unlike IDA, in order to find the evolution of joint coordinates, the differential equations of motion have to be integrated with respect to time.

Although torques that produce a specific known motion (for instance, acquired from an IDA) were used as input information for a FDA, the computed motion would not be the same as the reference one. This is caused by numerical integration errors and the unstable character of human walking, in which the accumulated error leads to a unstable solution of motion and the model usually falls down. In order to solve this problem, control methods are used. For example, Pätkau [35] studied two different control methods to stabilize numerical integration and find a correct stable motion: a proportional-derivative (PD) control, and the computed torque control (CTC). Results showed that the second approach led to a lower error.

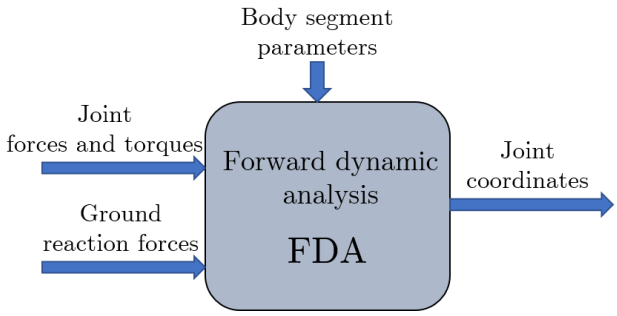


Figure 2.6 Diagram of a forward dynamic analysis.

2.3. Human motion prediction

Over recent years, a growing interest in motion prediction has appeared. Some applications of motion prediction are: the dynamic simulation of a specific motion task and the study of cause-effect relations, the design optimization of assistive devices, or the anticipation of surgery results [26, 29]. In order to predict human motion, the most frequently used approaches are optimization based methods where a certain cost function is minimized. For instance, minimizing weighted muscle activation, squared muscle excitations or squared joint torques [2, 9, 37]. Considering the methods mentioned in Section 2.2, three different approaches for motion prediction can be distinguished: inverse dynamics-based methods, forward dynamics-based methods and predictive dynamics methods.

Inverse dynamics based methods are those prediction methods in which the design variables are joint coordinates and their time derivatives. These motion variables are discretized, and for each iteration, the cost function is evaluated and an IDA is solved in order to get the joint torques. Xiang *et al.* [41] predicted a walking motion for different backpack weights using inverse dynamics prediction.

On the other hand, in forward dynamics-based methods the predicted motion is obtained using joint torques, muscle forces or other muscle parameters as design variables that are discretized. At each iteration the cost function is evaluated and, through a FDA, the motion is acquired. Anderson *et al.* [4] used this approach to predict a full gait cycle using a 3D neuromusculoskeletal model.

Finally, in predictive dynamics methods both joint coordinates and joint torques are design variables. The set of variables are discretized and equations of motion are converted into a group of algebraic constraint equations. At each iteration, the cost function is evaluated and the design variables are forced to satisfy the constraint equations. In [2] different cost functions were studied for a 2D gait prediction. Another example that uses this approach can be found in [26], where the autors predicted a pedaling motion using a 2D lower-limb musculoskeletal model in which an optimal control problem was solved to find muscle excitations.

3. SKELETAL MODEL

This chapter presents the skeletal model employed in the thesis. Moreover, the configuration of markers used in the motion capture is explained and illustrated.

3.1. Model used in the project

The model used in this project is the *Gait10dof18musc*. This model is provided by OpenSim and was created by Ajay Seth, Darryl Thelen, Frank C. Anderson and Scott L. Delp [18]. It is a 2D simplified model focused on the lower extremity.

This model was mainly employed in two different tasks: obtaining the experimental coordinates and forces/torques from the motion capture, through inverse kinematic and dynamic analyses; and performing an inverse dynamic analysis at each iteration of the optimal control algorithm in GPOPS-II (see Section 6.3.1 for a further explanation). It should be noted that for the first task, the model was slightly changed by adding three more degrees of freedom at the pelvis. This is due to the fact that the motion capture is carried out in a 3D space and, therefore, in order to properly scale the model, model markers must be relocated as similar as possible to experimental markers (this process is explained in detail in Section 5.1). Thus, this edition allowed the model to be positioned and oriented in the 3D space and, thereby, correctly performing the scaling process and the marker relocation. However, only experimental coordinates and forces/torques from the original 2D multibody system have been considered.

In following sections, the model provided by OpenSim is detailed. Bodies, joints and generalized coordinates are described and presented.

3.1.1. Bodies

HAT (head, arms and trunk), pelvis and leg segments define a model composed by 12 rigid bodies and the ground. Body segment parameters needed for each body are automatically determined by OpenSim. The different bodies and their corresponding name in Opensim are listed in Table 3.1 and shown in Figures 3.1 and 3.2.

Body name	Name in OpenSim
HAT (head, arms and trunk)	*torso*
Pelvis	*pelvis*
Right femur	*femur_r*
Left femur	*femur_l*
Right tibia	*tibia_r*
Left tibia	*tibia_l*
Right talus	*talus_r*
Left talus	*talus_l*
Right calcaneus + metatarsal bones	*calcn_r*
Left calcaneus + metatarsal bones	*calcn_l*
Right phalange bones	*toes_r*
Left phalange bones	*toes_l*
Ground	*ground*

Table 3.1 Body of the model. In italics the corresponding name in OpenSim.

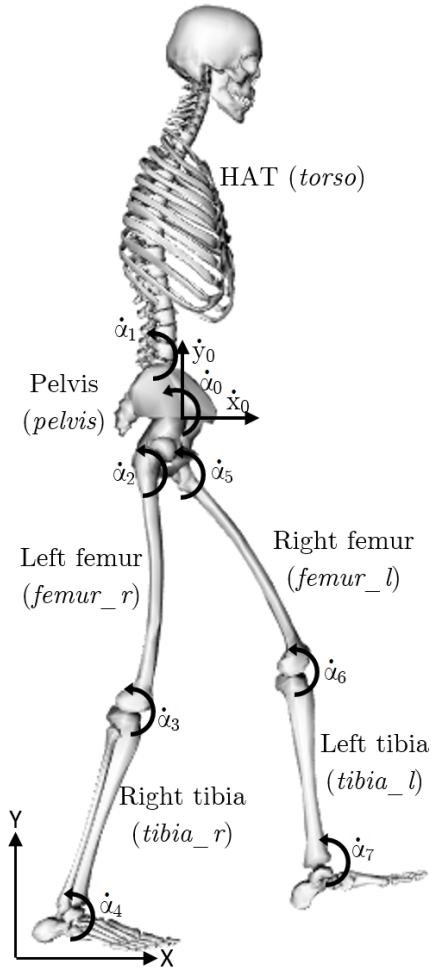


Figure 3.1 Bodies and generalized coordinates of the model. Between brackets the corresponding name in OpenSim. Note the positive direction of the generalized velocities.

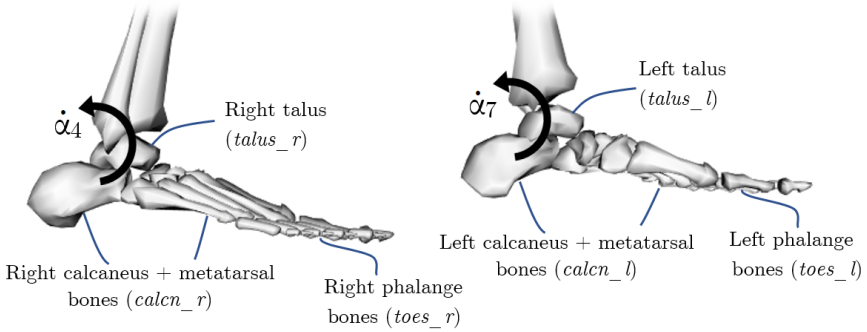


Figure 3.2 Bodies and generalized coordinates of the model, detail of feet. Between brackets the corresponding name in OpenSim. Note the positive direction of the generalized velocities.

3.1.2. Joints

The multibody system has a total of 10 degrees of freedom. Three degrees of freedom for the pelvis with respect to the ground, and the other seven correspond to relative movements between the different segments of the model which are linked with revolute joints. Note that during the project, the number of degrees of freedom in equations is represented by the letter n. Table 3.2 shows the existing joints between bodies, as well as the number of degrees of freedom allowed for each joint. "Parent" and "Son" names are used to express the relative motion between bodies: son bodies move with respect to parent bodies.

Moreover, as seen in Table 3.2, subtalar and metatarsal joints are locked, there are not degrees of freedom between their parent and son bodies. This configuration makes the foot to be a whole rigid body with an only degree of freedom with respect to the tibia.

Joint name	**DOF**	**Parent Body**	**Son Body**	**Name in Opensim**
Ground - Pelvis	3	Ground	Pelvis	*ground_pelvis*
Lumbar joint	1	Pelvis	HAT	*back*
Right hip	1	Pelvis	Right femur	*hip_r*
Right knee	1	Right femur	Right tibia	*knee_r*
Right ankle	1	Right tibia	Right talus	*ankle_r*
Right subtalar	0	Right talus	Right calcaneus	*subtalar_r*
Right metatarsal	0	Right calcaneus	Right phalange bones	*mtp_r*
Left hip	1	Pelvis	Left femur	*hip_l*
Left knee	1	Left femur	Left tibia	*knee_l*
Left ankle	1	Left tibia	Left talus	*ankle_l*
Left subtalar	0	Left talus	Left calcaneus	*subtalar_l*
Left metatarsal	0	Left calcaneus	Left phalange bones	*mtp_l*

Table 3.2 Model joints and degrees of freedom allowed for each of them. "Parent" and "Son" names are used to express the relative motion between bodies. In italics the corresponding name in OpenSim.

3.1.3. Generalized coordinates

There are 10 generalized coordinates in the model. Since the number of degrees of freedom is the same as the number of generalized coordinates, the system is holonomic. The generalized coordinates can be classified in two groups: the absolute coordinates, those that express the position and orientation of the pelvis with respect to the ground; and the relative coordinates, which express the orientation of a son body with respect to its parent body. Note that during the project, vectors of generalized coordinates, velocities and accelerations are expressed as $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, respectively.

In Table 3.3 the classification of generalized coordinates is made: the first row indicates the son body with its parent body between brackets. Then, in columns, the generalized coordinates that express the configuration of the son body with respect to its parent are written with the name used in OpenSim. Also, next to the generalized coordinate, the symbol used to illustrate them in Figures 3.1 and 3.2 is written between brackets.

Pelvis (Ground)	**HAT (Pelvis)**		**Femur (Pelvis)**	**Tibia (Femur)**	**Talus (Tibia)**
pelvis_tx (x0)	*lumbar_extension* (α_1)	Right	*hip_flexion_r* (α_2)	*knee_angle_r* (α_3)	*ankle_angle_r* (α_4)
pelvis_ty (y0)					
pelvis_tilt (α_0)		Left	*hip_flexion_l* (α_5)	*knee_angle_l* (α_6)	*ankle_angle_l* (α_7)

Table 3.3 Generalized coordinates of each son body with respect to its parent. Between brackets the symbol used to illustrate them in Figures 3.1 and 3.2.

3.2. Marker protocol

Once defining the model used in the project, it is necessary to design the marker protocol. In order to design the model, advanced knowledge of human physiology is needed and usually, predefined models are taken from specialists. On the contrary, the marker protocol depends more on the capture and different configurations can be made, as long as certain guidelines are followed.

First, it is important to understand which is the goal of the motion capture and the calculations involved in. One must identify the bodies that correspond to the model and focus the markers position on them. It has to be possible to mathematically determine the position and orientation of each body at any instant of time. In general, in order to define the configuration of a body in the 3D space, three markers are required. For instance, this is the case of the pelvis, the body that is linked to the ground with six degrees of freedom.

As soon as the position of the body with the general movement is known, it is necessary to determine the position of its son bodies. If the joint with the following body is of three rotational degrees of freedom, two markers are needed to position the second body in relation to the first, whereas none

of them is in a common point between the bodies. In case of being a joint of one or two rotational degrees of freedom, only one marker is needed in the son body to determine its position with respect to its parent, and again, assuring that the marker is not placed in a common point between the bodies.

Moreover, it is important to design the markers configuration by following two criteria. The first one is the configuration to be as simple as possible, as the higher the density of markers is in the capture, the more complicated is to capture the motion and then treating the corresponding data. On the other hand, increasing the number of markers guarantees not to lose information of the bodies position. If a marker is lost in a period of time, the position of the others can be enough to ensure the configuration of the body. Hence, an equilibrium of these two criteria has to be found.

Finally, it must be taken into account another detail: the position of the marker in the body. Positions for markers should be easy to identify, either because they are known anatomical points, or for being positions easily to find from other points. Also, it is very favorable to place markers as close as possible to bones, where the quantity of tissue between bone and skin is the least possible. Thereby, undesirable movements from soft tissue, or what is the same, relative displacements of markers with respect to the bone, called soft tissue artifacts (STA) [11], are prevented.

3.2.1. Marker protocol of the project

The configuration of markers used for the motion capture is based on the Plug-in Gait marker placement, by ©Vicon Motion Systems, for the lower limbs. However, some modifications were made from the original configuration: one extra marker was added on the toes, only one marker was placed on the pelvis back, the tibia marker was relocated below the knee, and finally, because of the anatomical imprecision of placement and high soft tissue artifacts, the thigh markers were deleted. For the trunk, a total of five markers were placed: one on the head, two on the shoulders, one more in the sternum, and lastly, one on the C7 spinal segment of the neck.

The 22 markers used for the capture are listed in Table 3.4 and illustrated in Figure 3.3. Since the *Gait10dof18musc* model has already a marker protocol, the name used for the markers has been chosen the same.

Body	Markers in the body
HAT (*torso*)	- Top.Head (1) - C7 (2) - R.Acromium (3) - L.Acromium (4) - Sternum (5)
Pelvis (*pelvis*)	- R.ASIS (6) - L.ASIS (7) - V.Sacral (8)
Right femur (*femur_r*)	- R.Thigh.Upper (9) - R.Knee.Lat (10)
Left femur (*femur_l*)	- L.Thigh.Upper (11) - L.Knee.Lat (12)
Right tibia (*tibia_r*)	- R.Shank.Front (13) - R.Ankle.Lat (14)
Left tibia (*tibia_l*)	- L.Shank.Front (15) - L.Ankle.Lat (16)
Right calcaneus + metatarsal bones (*calcn_r*)	- R.Heel (17) - R.Toe.Lat (18) - R.Toe.Med (19)
Left calcaneus + metatarsal bones (*calcn_r*)	- L.Heel (20) - L.Toe.Lat (21) - L.Toe.Med (22)

Table 3.4 List of markers used in the motion capture. Between brackets the number that corresponds to the marker in Figure 3.3.

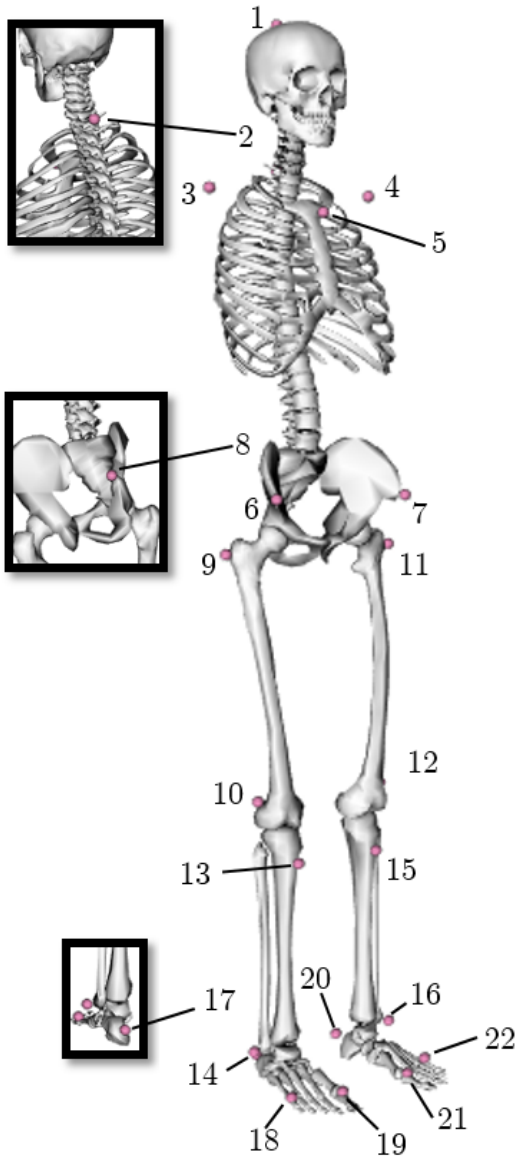


Figure 3.3 Distribution of markers in the body.

4. DATA COLLECTION

The experimental data necessary for the project was collected at the Biomechanics Laboratory of UPC, located at ETSEIB. Below all the equipment used to capture the motion is described. Then, the experimental procedure followed and the data processing are explained.

4.1. Laboratory equipment

The UPC Biomechanics Lab consists of a motion capture equipment OptiTrak^{TM} from NaturalPoint Inc, designed to capture the position of points in a 3D space. The system disposes of 16 cameras V100:R2 model, which incorporate infrared light (IR) LED's. The cameras can obtain the position of points by emitting the infrared light, which is reflected in markers fixed to the body being captured. The markers are little spheres covered with a very reflective fabric. The reflected light is discretely taken by the optical system of the cameras in a sampling rate of 100 Hz or fps (frames per second). It should be noted that each camera is only capable of measuring the position of each marker in the perpendicular plane to its optical axis. Figure 4.1 shows an example of a marker (4.1a), and the model of camera used in the capture (4.1b).

Combining the information of the 16 cameras, the system is able to determine the position of all the markers in the 3D space. Also, in order to process the information, collected data is transferred to a computer through the hubs, small USB connection boxes where a maximum of six cameras can be connected with USB 2.0 cables. In the computer, the software Motive allows to find the 3D trajectories of the markers.

(a) **(b)**

Figure 4.1 Reflective marker (4.1a) and model of camera (4.1b) used for the motion capture.

Cameras are positioned for the purpose of achieving more precision in a capture of human gait [6]. Eight of them are placed at three meters from the ground, the other eight are located at a meter and a half. This camera layout enables a good capture of human gait of four steps, *i.e.*, two whole walking cycles. In Figure 4.2 a view of the Biomechanics Lab with some of the cameras can be seen.

Figure 4.2 View of the Biomechanics Lab. Cameras can be seen in high and low position, as well as two force plates at the center of the room.

Finally, ground reaction forces are measured with two force plates AMTI Accugait. Each one of them measures the contact wrench (three forces and three moments) applied to the body in contact, and referred to the center of the plate. These plates are placed on the floor, at the center of the Lab (see Figure 4.2). An example of one force plate is shown in Figure 4.3.

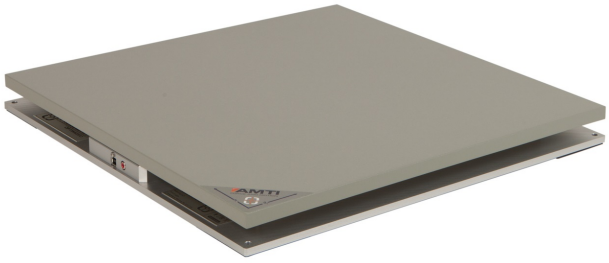


Figure 4.3 Force plate AMTI Accugait.

The information is directly transferred to a computer via RS-232 connection. As well as the cameras, the signal is sampled at 100 Hz. In order to synchronize the contact wrench measurement with the frames captured by the OptiTrak^{TM} system, an output signal is implemented by Matlab that indicates whether the cameras are recording or not.

4.2. Motion capture and data processing

Using the optical system and the force plates mentioned above, experimental data was taken. Two different captures were carried out:

- A static capture, in which the subject was quiet. This capture is needed to scale the model when OpenSim is used.
- A natural gait capture, being the subject recorded while walking in a normal way. This information is required to obtain the experimental coordinates used in GPOPS-II.

Note that in order to execute the capture, Matlab and Motive software are necessary. In the next sections, the experimental procedure followed to capture the motion and the subsequent data treatment are explained.

4.2.1. Experimental procedure

First of all, so as to capture the motion, reflective markers were placed onto the body of the subject. They must be placed following the protocol explained in Section 3.2, in the same anatomical points chosen in the model of OpenSim. To have the least relative movement between markers and bones, the subject worn sportive tights and a tight t-shirt, clothes that are attached to the body while the subject is walking. Markers were fixed employing adhesive tape, and for the marker placed onto the head, a cap was used. In Figure 4.4 it is shown the subject wearing the described clothes and the markers located at the anatomical positions of the protocol.

The motion was captured with Motive software, which tracks the position of markers by receiving the signals from the cameras. It must be noted that before starting the capture, a calibration file must be loaded in order to ensure that the optical system is calibrated and guarantees a good recording of the markers position. Also, it has to be checked that all the markers are visible by the cameras, and ensure that any object inside the room can interfere with the cameras, such as shiny objects or jewellery, which can reflect the infrared light emitted by the cameras.

Figure 4.4 Marker configuration placed on the subject.

Once the previous actions were done, the capture could start. A male subject of 65 kg of mass and 1,8 of height was recorded. First, the static pose was captured. The subject was quiet in a normal and relaxed pose while cameras took the position of markers for a short period of time (5 - 6 seconds). Then, the natural walking capture was carried out. The participant walked straight in his usual speed, crossing the force plates. In order to properly register foot-ground forces, only one feet must be supported by a plate at the same time. Thus, while walking, the subject stepped on with right foot on the first plate and with left foot on the second one (recording a HE-HE cycle for right leg). The walking motion capture was repeated five times, and the capture with the least number of errors was taken.

4.2.2. Data treatment

As soon as the capture is finished, data has to be treated and modified before it is analyzed with OpenSim software. Two different files must be processed separately: the motion capture files, which are the static and walking capture, and the foot-ground forces file.

Motion capture files

Before exporting files from Motive, walking capture must be edited. The program has the necessary tools to edit the marker trajectories if a strange behavior is detected. For example, in case a marker has been lost by the program during a period of frames, the gap is filled by the tool fill gaps. If the period is short, a spline curve can be used to fill the gap. On the contrary, if the period is rather

long, it is preferred to copy the information from another marker that has the same trajectory. Also, in case that the cameras loose a marker that after some frames it appears as a new one, both marker trajectories have to be merged with the editing tool merge markers. Finally, once all the capture is checked and arranged, all markers have to be named, using the same name as in OpenSim.

Then, files are exported as a cvs file extension, in which the X, Y and Z positions of markers over all the capture is saved. This extension cannot be read by OpenSim (it needs a .trc file extension), and a Matlab program is implemented to transform the information and create the .trc file. This program has to ensure the following tasks:

- Reading and taking all the information from the cvs file.
- Adjusting the axes, since they are different in Motive and OpenSim. The relation between axes is: $X_{OpenSim} = Z_{Motive}$, $Y_{OpenSim} = Y_{Motive}$ and $Z_{OpenSim} = -X_{Motive}$.
- Writing the required information to the .trc file. A header with the following settings information: name of the file, the camera sampling rate, number of markers, number of frames and units used. And then, in columns, the X, Y and Z positions of the different markers over all the capture.

Foot-ground forces file

Unlike motion files, this file is registered directly in Matlab and saved as a .mat file extension. It contains a matrix with the contact wrench for each plate over all the capture, in columns, and an extra column for the signal that synchronizes the motion capture with the force plates. As well as before, a specific file extension (.mot file extension) needs to be created in order to work with this data in OpenSim.

Due to the electrical noise that the registered signal contains, a low-pass filter has to be applied. As OpenSim does not have filtering options for foot-ground contact forces data, it has to be filtered in Matlab. Moreover, if a a whole gait cycle is wanted to be studied, as it has been said in the project scope (Section 1.3), a problem needs to be faced. When right heel strikes the first plate, left foot is still in contact with the ground, and therefore, a third force plate located before the first one is necessary to obtain all the contact wrench involved in a complete cycle. In order to solve this issue, considering that only two force plates are able in the Lab, periodicity in human gait is assumed. Thereby, left foot-ground contact forces registered by the second plate are copied in the frames in which the left foot is still touching the ground, as if forces were registered from a plate located before the first one. Taking all of this into account, the program implemented in Matlab needs to do the following tasks:

- Selecting the contact wrench data only when the cameras are active (only when the synchronization signal is open).

- Filtering the data with a first order Butterworth filter with a cut-off frequency of 8 Hz. Since forces does not have to be differentiated, a first order filter is enough. Furthermore, a cut-off frequency of 8 Hz is adequate to remove the electrical noise.
- Adjusting the axes, since they are different in the force plates and OpenSim. The relation between axes is: $X_{OpenSim} = X_{ForcePlates}$, $Y_{OpenSim} = Z_{ForcePlates}$ and $Z_{OpenSim} = Y_{ForcePlates}$.
- Coping the foot-ground contact forces to the frames in which left foot is still in contact with the ground, as explained before. Loss of contact between left foot and ground is determined to copy the correct frames from the second force plate to these frames. As an example, vertical force measured in force plates is shown in figure 4.5.
- Writing the required information to the .mot file. A header with the following settings information: name of the file, number of columns, number of rows and the time range. And then, in columns, the time, the point of application of forces for each force plate and finally, the contact wrench in each force plate over all the capture.

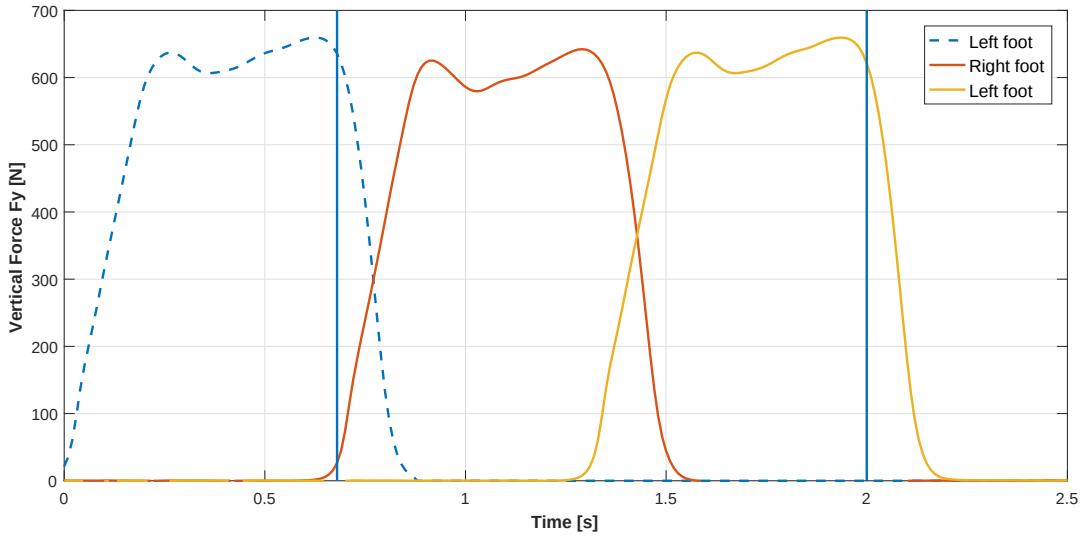


Figure 4.5 Example of vertical force measured in force plates. The dashed line represents the vertical force copied from the second plate. Vertical lines determine the gait cycle, and therefore, the information of foot-ground reaction forces that is used.

Once this new files are created, experimental data is ready to be analyzed with OpenSim software.

5. KINEMATIC & DYNAMIC ANALYSES

In order to perform the inverse dynamic analysis and the previous related tasks, the software OpenSim 3.3 [19] is used. The graphical user interface (GUI) with the loaded model looks as shown in Figure 5.1.

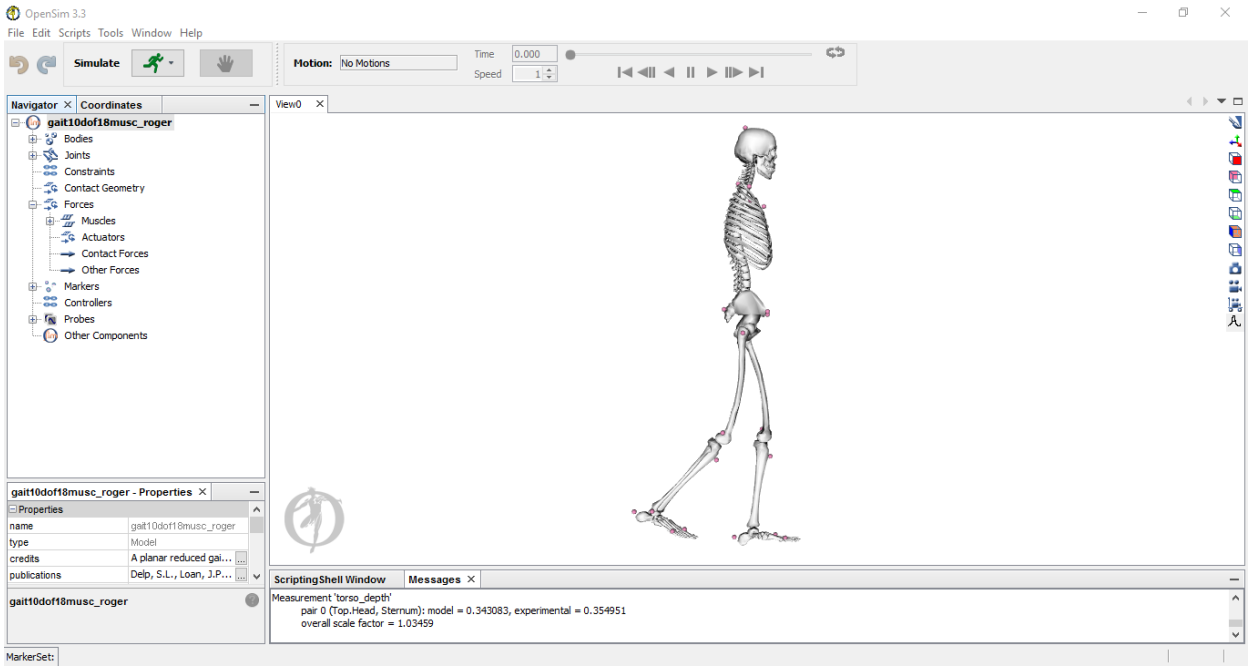


Figure 5.1 OpenSim GUI with the loaded model.

The *Navigator* window allows to select which elements are shown, such as bodies, joints or markers. In the *View* window, all the chosen elements are displayed. For instance, as can be seen in Figure 5.1, the whole skeletal model is displayed. Moreover, the *Tools* menu contains the main tools employed in the project. Following, a brief description of them is done:

- *Scale Model*: from a generic skeletal model, dimensions and inertias are transformed and adapted to the specific subject whose motion has been captured.
- *Inverse Kinematics*: it gets the value of generalized coordinates over time from the recorded position of experimental markers.
- *Inverse Dynamics*: it allows to obtain joint torques from a specific motion, described by its generalized coordinates.

5.1. Model scaling

Before starting the analysis, the skeletal model needs to be scaled in order to be as similar as possible to the real subject. The *Scale Model* tool allows to adapt the general model to the subject dimensions, acquiring body segment parameters automatically. Also, once the scaling process is done, this tool enables to relocate model markers in the most similar way as they were placed on the subject during the experimental trial.

In order to perform the scaling process, two files are needed: the static capture file (.trc) and the file that contains the generic model which is wanted to modify (.osim). The result is a new file containing the scaled model (.osim). Figure 5.2 shows the files involved in this process:

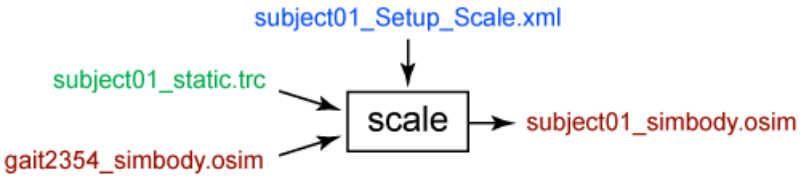


Figure 5.2 Scheme of files involved in the scaling process. Red files are model files, on the left the general model, on the right the scaled model. Green file is the static capture with the experimental data. Finally, blue file is a settings file, it is not needed if configurations are set up manually. Adapted from [16].

When editing *Scale Model* settings, the first step is to specify the mass of the model, which is the total mass of the real subject. So as to preserve the total mass when it is distributed segment by segment, the option *Preserve mass distribution during scale* must be activated. Next, scale factors, which are the relationship of distance between two markers in experimental conditions and in the model, have to be defined. Figure 5.3 and Eq. 5.1 show how this factors are represented mathematically:

$$scale\_factor_i = \frac{e_i}{m_i}, \quad i = 1, 2, \ldots, k, \tag{5.1}$$

being k the number of scale factors used.

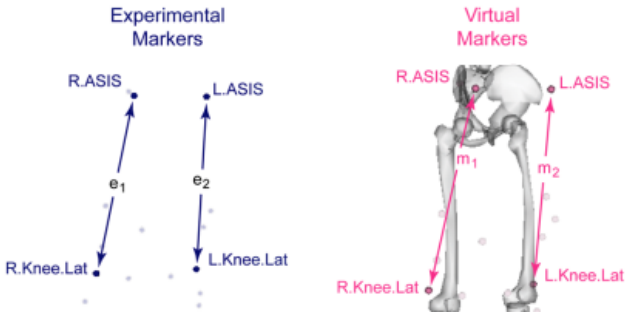
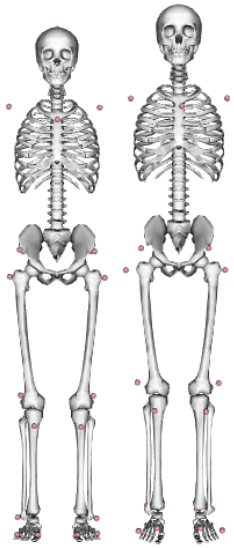


Figure 5.3 Distance of two markers. On the left, e_i represents the distance between two experimental markers. On the right, the distance of model markers is represented with m_i. Adapted from [17].

As it is shown in the previous figure, pairs of markers have to be chosen to define scale factors. As

soon as they are defined, they must be associated to the bodies of the model. Furthermore, bodies can be scaled with the same scale factor in all three directions, or choose different scale factors for each direction. Changes after scaling can be observed in Figure 5.4a. Pairs of markers and scale factors are shown in Figure 5.4b. Same scale factors have been used for right and left bones in order to have the same dimensions on both sides of the skeletal model.



Settings | Scale Factors | Static Pose Weights

Use measurements = = Uniform Edit Measurement Set

Use manual scales = = Uniform Reset to Measurement

Body Name	Measurement(s) Used			Applied Scale Factor(s)		
ground	Unassigned					1.0
pelvis	pelvis_depth	Unassigned	pelvis_width	0.996772	1.0	0.879997
femur_r	Unassigned	femur_lenght	Unassigned	1.0	0.936526	1.0
tibia_r	Unassigned	tibia_length	Unassigned	1.0	1.058578	1.0
talus_r	feet_length	feet_width	feet_width	0.948448	0.882099	0.882099
calcn_r	feet_length	feet_width	feet_width	0.948448	0.882099	0.882099
toes_r	feet_length	feet_width	feet_width	0.948448	0.882099	0.882099
femur_l	Unassigned	femur_lenght	Unassigned	1.0	0.936526	1.0
tibia_l	Unassigned	tibia_length	Unassigned	1.0	1.058578	1.0
talus_l	feet_length	feet_width	feet_width	0.948448	0.882099	0.882099
calcn_l	feet_length	feet_width	feet_width	0.948448	0.882099	0.882099
toes_l	feet_length	feet_width	feet_width	0.948448	0.882099	0.882099
torso	torso_depth	torso_lenght	torso_width	1.034592	1.034592	1.052801

Measurement Set

Measurements	Marker Pairs				
feet_length	L.Heel	L.Toe.Med	L.Heel	L.Toe.Lat	
feet_width	L.Toe.Med	L.Toe.Lat	R.Toe.Med	R.Toe.Lat	
tibia_length	L.Ankle.Lat	L.Shank.Front	R.Ankle.Lat	R.Shank.Front	
femur_lenght	L.Knee.Lat	L.Thigh.Upper	R.Knee.Lat	R.Thigh.Upper	
pelvis_width	R.ASIS	L.ASIS			
pelvis_depth	R.ASIS	V.Sacral	L.ASIS	V.Sacral	
torso_lenght	Sternum	Top.Head			
torso_width	R.Acromium	L.Acromium			
torso_depth	Top.Head	Sternum			

(a) **(b)**

Figure 5.4 Scaling process. (5.4a) Model before (on the left) and after (on the right) the scaling process and (5.4b) scaling factors of each body defined by marker pairs.

After scaling the model, model markers are relocated using the *Adjust model markers* option. Different weights can be chosen to adjust the different markers. High weighted markers are those with a precise anatomical position. On the contrary, low weights are chosen for markers with a more imprecise anatomical position. To check whether the scaling process and the relocation of markers have been successful, experimental markers of the static capture can be superimposed on the skeletal model. Figure 5.5 shows this association:

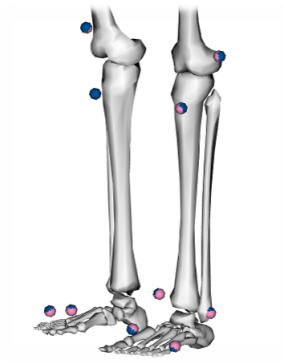


Figure 5.5 Lower limbs of the model where the association of experimental (blue) and model (pink) markers can be seen.

5.2. Inverse kinematics

Once the model is scaled and markers are relocated, the inverse kinematics process must be done to find the kinematics of the system bodies. The *Inverse Kinematics* tool obtains the history of the model's generalized coordinates by following the position of experimental markers over time.

In order to perform the inverse kinematic process, two files are necessary: the walking motion capture (.trc) and the file with the scaled model (.osim). The result is a *Motion* file (.mot) which includes the evolution of the generalized coordinates of the model, *i.e.*, their values at every instant of time. Figure 5.6 shows the files involved in the process.

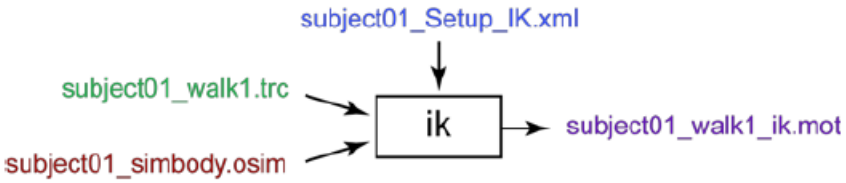


Figure 5.6 Scheme of files involved in the inverse kinematic process. Red file is the scaled model which is being analyzed. Green file is the walking motion capture. Blue file is a settings file, it is not needed if configurations are set up manually. Finally, purple file is the result motion file with the evolution of the generalized coordinates. Extracted from [14].

When editing *Inverse Kinematics* settings, the interval of time along which a solution has to be obtained must be indicated. Furthermore, weights for each marker can be chosen. Similar as the scaling process, these weights express the degree of accuracy associated to each marker, depending on how clear its position in the body is and how much soft tissue artifacts its position measurement contains. Mathematically, inverse kinematics is solved as an optimization problem, in which the distance between the position of model markers and experimental markers is minimized, for every instant of time. Hence, the objective function to minimize is:

$$\min_{q} \sum_{i=1}^{m} w_i \|\mathbf{x}_i^{exp} - \mathbf{x}_i(\mathbf{q})\|^2, \quad i = 1, 2, \ldots, m, \tag{5.2}$$

where m is the number of markers used in the analysis; w_i is the weight assigned to the i^{th} marker; $\mathbf{x}_i(\mathbf{q}) \in \mathbb{R}^3$ is the position vector of the i^{th} marker of the skeletal model, which depends on the system configuration described by the vector of generalized coordinates $\mathbf{q} \in \mathbb{R}^n$ (being n the number of degrees of freedom); finally, $\mathbf{x}_i^{exp} \in \mathbb{R}^3$ is the position vector of the i^{th} experimental marker.

As soon as the inverse kinematics process is done, the *Messages* window shows the report of the result, where the marker errors are evaluated. Approximately, for all markers, the maximum error should not exceed 2-4 cm and the RMS error should not be greater than 2 cm. Moreover, errors can also be appraised visually by associating the resulting file (.mot) to the motion capture file (.trc), so it shows the movement of the model and of the experimental markers simultaneously.

5.3. Inverse dynamics

Once the inverse kinematics process is completed, inverse dynamic analysis can be performed. As explained in Section 2.2.1, the inverse dynamic analysis is the process in which forces and torques are computed from a known motion. Net joint torques that produce the captured walking motion are obtained. Furthermore, residual forces and moments acting on the pelvis are found (since the skeletal model in this project is two dimensional, there are two residual forces, Fx and Fy; and one residual moment, Mz). Further information about actuators involved in a skeletal model was included in Section 2.1.3.

In reality, there is not any physical element that links the pelvis to the ground, therefore, residual forces and moments must be as close as possible to zero. When a subject is walking, it only receives external forces from the ground. Thus, high values of the residual wrench imply a dynamical inconsistency between the motion of the model and the measured foot-ground forces. However, it should be noted that is impossible to have null values of residual forces and moments. This is due to the existence of errors and their propagation. Different sources of error can be pointed out, such as: simplifications of the model with respect to the reality, the presence of soft tissue artifacts, errors in the measure systems and uncertainty in body segment parameters, among others.

In OpenSim, the inverse dynamic analysis is done by using the tool *Inverse Dynamics*. This tool needs three files: the motion file with the evolution of generalized coordinates (.mot), the foot-ground forces file (.mot), and finally, the file with the scaled model (.osim). The result is a *Storage* file (.sto) which contains the time histories of the net joint torques and the residual wrench. The following diagram (Figure 5.7) shows the files involved in this process:

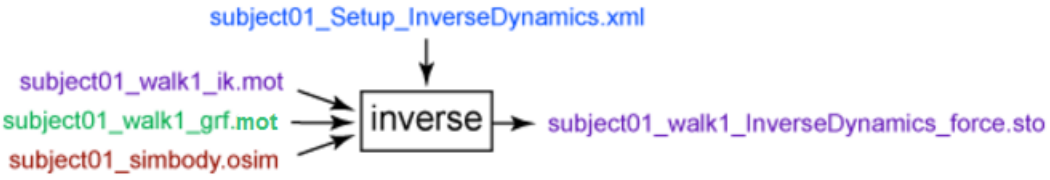


Figure 5.7 Scheme of files involved in the inverse dynamic process. Red file is the scaled model which is being analyzed. Green file is the foot-ground forces registration. Blue file is a settings file, it is not needed if configurations are set up manually. Left purple file is the result motion file with the evolution of the generalized coordinates. Finally, right purple file is the net joint torques and residual wrench storage. Adapted from [13].

When editing *Inverse Dynamics* settings, first, the interval of time along which a solution has to be obtained must be indicated. Then, because of noise that appears in the inverse kinematics process, the signals of the motion file must be filtered. If a low-pass filter is not used, when generalized coordinates are differentiated to obtain velocities and accelerations, the noise is highly increased, resulting in a distorted calculation of joint torque values. OpenSim has the option *Filter Coordinates*

to choose a cut-off frequency to filter the signals of generalized coordinates. It has been checked that a cut-off frequency of 6 Hz is enough to remove the noise, while maintaining the desired signal. Finally, regarding external forces, force plates have to be defined. For each plate, forces, moments and point of application must be chosen from the foot-ground forces file. Moreover, it needs to be determined the body in which the contact wrench is applied to. In this study, forces and moments are applied to right calcaneus (*calcn_r* in OpenSim) for the first plate, and to left calcaneus (*calcn_l* in OpenSim) for the second plate. An example of this configuration can be seen in the following Figure 5.8:

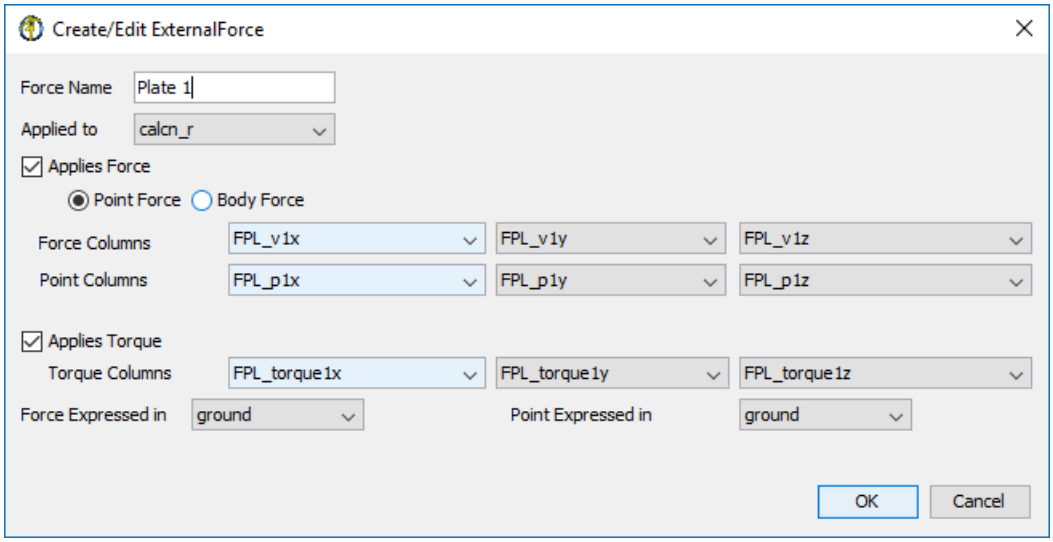


Figure 5.8 Configuration of ground reaction forces for the first plate "Plate 1".

The inverse dynamic analysis of a complex multibody system, such as skeletal models, is ideal to be treated by means of analytical mechanics. As it has been presented in Section 3.1.3, the skeletal model is holonomic, since it has the same number of generalized coordinates and degrees of freedom. Furthermore, its motion is described with the least number of generalized coordinates, which are independent, and its first time derivatives (generalized velocities). These two characteristics make possible to find the equations of motion of the multibody system through *ordinary Lagrange equations* [3]:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} = F_i^*, \quad i = 1, 2, \ldots, n, \tag{5.3}$$

where n is the number of degrees of freedom; q_i and $\dot{q}_i$ are respectively the i^{th} generalized coordinate and velocity defined in the model; T is the kinetic energy of the system and U the potential energy of the system associated with the gravitational field; finally, F_i^* is the generalized force associated with non-conservative forces of the i^{th} generalized coordinate.

Hence, a system of n equations (5.3) is obtained. This system has n unknowns that appear in the terms of generalized forces F_i^*, *i.e.*, joint torques for each generalized coordinate and the residual

wrench. Moreover, this system of equations is commonly written in the following matrix form:

$$[\mathbf{M}(\mathbf{q})]\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{U}(\mathbf{q}) = \mathbf{F}^*, \tag{5.4}$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}} \in \mathbb{R}^n$ are the vectors of generalized coordinates, velocities, and accelerations, respectively; $[\mathbf{M}(\mathbf{q})] \in \mathbb{R}^{n \times n}$ is the system mass matrix; $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ is the vector of inertial forces (Coriolis and centrifugal forces); $\mathbf{U}(\mathbf{q}) \in \mathbb{R}^n$ is the vector associated with gravitational forces; finally, $\mathbf{F}^* \in \mathbb{R}^n$ is the vector of generalized forces that contains the unknown joint torques and residual wrench.

Once OpenSim finishes the calculations, results of the inverse dynamics analysis must be assessed. In order to evaluate whether the process has been satisfactory or not, residual forces and moments should be plotted and analyzed. The gait cycle recorded in the walking motion capture has been analyzed, using the 2D model described in Section 3.1. Figure 5.9 shows the residual wrench obtained after doing the inverse dynamic analysis.

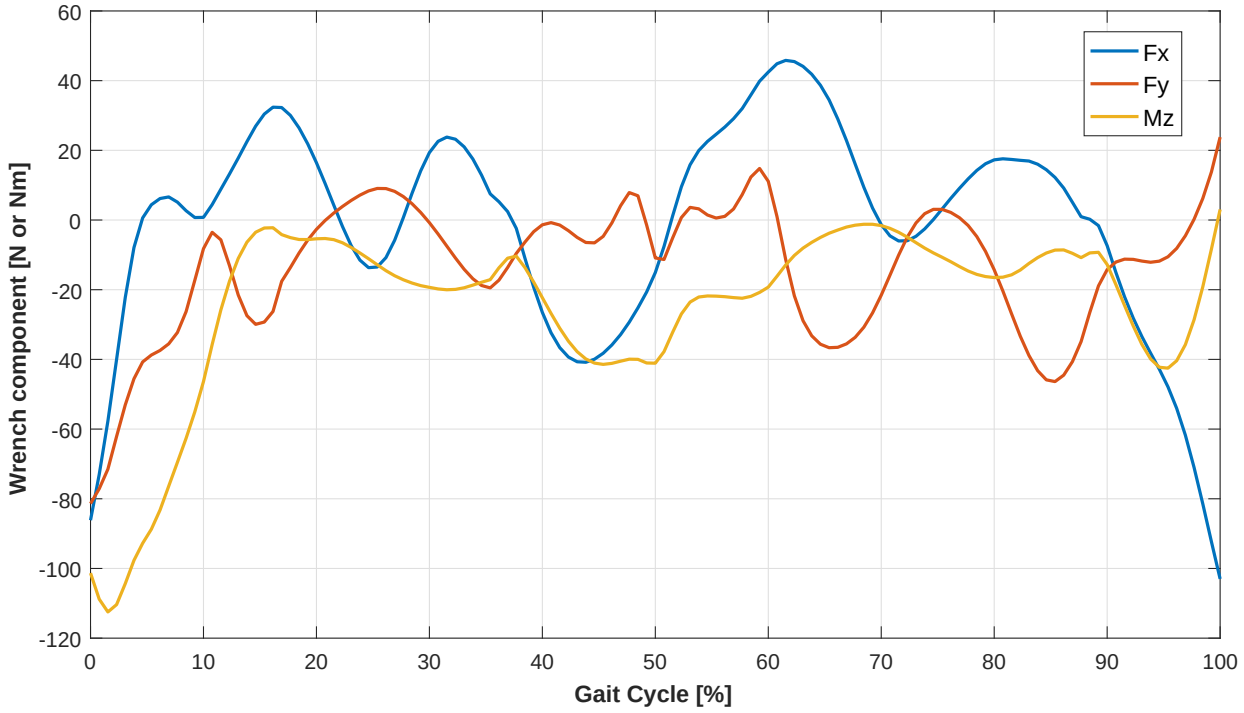


Figure 5.9 Residual forces (Fx, Fy) and moment (Mz) during all the gait cycle. Data is presented in percentage of gait cycle.

Apart from the beginning and the end of the gait cycle, the residual wrench varies from values that could be accepted for a model of 65 kg. However, if residual forces are wanted to be minimized further, one option can be the implementation of optimal control techniques, which are capable to reduce the residual wrench by predicting a new motion. Those techniques are explained in the next chapter.

6. PREDICTION FRAMEWORK

The optimal control problem is introduced as an approach to predict motions. Also, different methods to solve this problem are presented. Then, the program used in this project, GPOPS-II, and its solving method are explained in more detail. Finally, the simulation methodology followed to obtain the results is described.

6.1. Optimal control

The optimal control problem consists in determining the control signals that will cause a process to satisfy some defined constraints, and at the same time, minimize (or maximize) some performance criterion [24]. It is sometimes referred as trajectory optimization. Optimal control problems are used in different applications. For example, in aerospace engineering, optimal control formulations are employed to determine optimal trajectories and orbit transfers; or in economics, in which optimal investments of production strategies are wanted to be found [32].

As explained in Section 2.3, in biomechanical engineering an interest in motion prediction has increased. This has led to introduce optimal control formulations as a tool to determine new motions that minimize (or maximize) a certain criterion. In this thesis, optimal control problems are studied to minimize the residual wrench by obtaining new walking motions.

6.1.1. Optimal control problem statement

In a more mathematical framework, the optimal control problem can be described in the following form [36]: determine the state vector, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$, the control vector, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$, the initial time, t_0, and the terminal time, t_f, on the interval $t \in [t_0, t_f]$ that minimize the cost functional

$$J = \Phi(\mathbf{y}(t_0), t_0, \mathbf{y}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{y}(t), \mathbf{u}(t), t)\ \mathrm{d}t, \tag{6.1}$$

subject to the dynamic constraints

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = f(\mathbf{y}(t), \mathbf{u}(t), t), \tag{6.2}$$

the inequality path constraints

$$\mathbf{c_{min}} \leq \mathbf{c}(\mathbf{y}(t), \mathbf{u}(t), t) \leq \mathbf{c_{max}}, \tag{6.3}$$

and the boundary conditions

$$\mathbf{b_{min}} \leq \mathbf{b}(\mathbf{y}(t_0), t_0, \mathbf{y}(t_f), t_f) \leq \mathbf{b_{max}}. \tag{6.4}$$

Equation 6.1 represents the cost functional that is wanted to be minimized in the Bolza form [5]. Dynamic constraints (Eq. 6.2) are the set of differential equations that govern states and controls. They are introduced as constraints in order to be satisfied during the optimization. Note that in optimal control, variables are separated in states and controls. Per definition, a state variable is a differentiated variable that appears on the left-hand side of the differential equations (Eq. 6.2). In contrast, a control variable is an algebraic variable [5].

Furthermore, inequality path constraints (Eq. 6.3) are constraints satisfied over all the interval of time. They can be simply bounds on states and controls or algebraic path constraints. Finally, boundary conditions (Eq. 6.4) are bounds on states and time. They are imposed at the beginning and at the end of the interval of time. It should be noted that if an equality constraint is desired, lower and upper bounds must be equal ($\mathbf{c_{min}} = \mathbf{c_{max}}$ or $\mathbf{b_{min}} = \mathbf{b_{max}}$).

6.1.2. Indirect versus direct approaches

Two main methods are used to solve the optimal control problem: indirect and direct methods. Being $\mathbf{x}^*$ the optimal solution, an indirect method attemps to find a root of the necessary condition $J'(\mathbf{x}^*) = 0$. On the other hand, a direct method constructs a sequence of points and compares values for the objective function $J(\mathbf{x_1}) > J(\mathbf{x_2}) \cdots > J(\mathbf{x}^*)$ [5].

Historically, optimal control problems have been solved using indirect methods. These methods use a branch of mathematics called *calculus of variations* to obtain a set of first-order necessary conditions for optimality. Also, a second order sufficiency check can be implemented to confirm that the extremal solution is a minimum or a maximum [24].

Although indirect methods are highly accurate, they have several disadvantages. For example, the optimality conditions are often difficult to formulate, or numerical methods that solve these equations require an accurate guess that usually is not intuitive. Because of these and other reasons [5], direct methods have risen and have become a very popular and useful tool. Direct methods convert (or transcribe) the optimal control problem to a nonlinear programming (NLP) problem, they solve the NLP problem and then estimate the error of the optimal control problem solution. Two of the most common transcription methods are explained in the next section.

6.1.3. Transcription methods

As Betts illustrates in [5], a NLP problem is characterized by a finite set of variables and constraints. In contrast, an optimal control problem involves continuous functions (the states, $\mathbf{y}(t)$, and controls, $\mathbf{u}(t)$). Thus, an optimal control problem can be viewed as an infinite-dimensional nonlinear problem.

Therefore, the goal of direct methods is to transcribe or convert the infinite-dimensional problem into a finite-dimensional approximation. Then, this finite-dimensional NLP problem can be solved using well-known parameter optimization methods. Hence, a transcription method has three fundamental steps:

- Convert the dynamic system into a problem with a finite set of variables.
- Solve the finite-dimensional problem using a parameter optimization method.
- Asses the accuracy of the finite-dimensional approximation and if necessary repeat the transcription and optimization steps.

Two main classes of transcript methods can be distinguished: shooting methods and simultaneous methods. Shooting methods discretize only controls, and propagate the dynamics across the interval using the approximation of controls. The algorithm tries to find the initial value of states and the approximation of controls that drives to zero the defect constraint (the error committed in the boundary conditions). There are two shooting methods: single shooting, and multiple shooting. Single shooting propagates the dynamics in all the trajectory. On the contrary, multiple shooting breaks up the trajectory into segments and uses single shooting for each segment. Figure 6.1 shows the differences between single and multiple shooting.

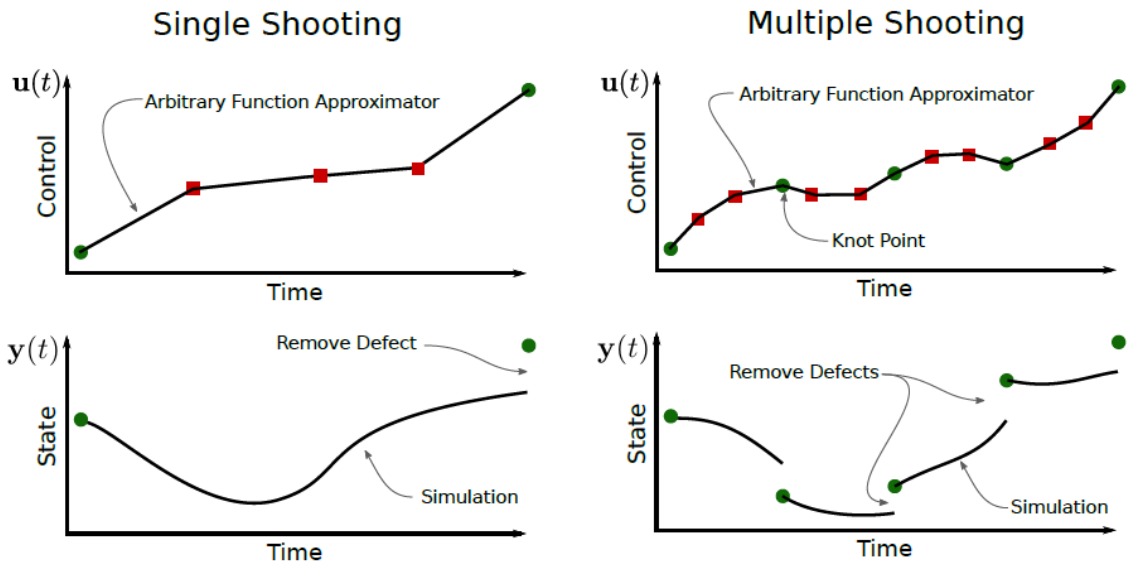


Figure 6.1 Single shooting versus multiple shooting methods. Adapted from [27].

On the other hand, simultaneous methods discretize both states and controls. These methods directly represent the state trajectory using an approximation, and then satisfy the dynamic constraints only at special points in the trajectory [27]. The most used simultaneous method is orthogonal collocation (also known as pseudospectral method [22]), which uses orthogonal polynomials to approximate states and controls. GPOPS-II, the program that is employed to solve the optimal control problem, uses an orthogonal collocation method.

6.2. GPOPS-II

GPOPS-II is a general-purpose MATLAB-based software for solving optimal control problems. It is developed by Michael Patterson and Anil V. Rao, from the University of Florida [31]. It solves general nonlinear multi-phase optimal control problems, where it is desired to optimize systems defined by differential-algebraic equations. The main advantage of this software is that the formulation of the problem is easy to understand. It allows to solve optimal control problems without a high knowledge in optimal control theory. Also, GPOPS-II has a user interface that permits to program the optimal control problem in an intuitive and compact way. It has an easy structure that any user with basic expertise in MATLAB can implement. Below, the main sections to construct the problem are briefly described (for further information see the user's guide [32]):

- **Setup:** In this section the main configurations are set, such as tolerances, maximum iterations, the NLP solver to use, etc. Also, auxiliary data is introduced if needed. For instance, in this project the captured motion is introduced as auxiliary data.
- **Bounds:** This section contains the boundary conditions defined in Eq. 6.4, and the simple bounds in states and controls that appear in Eq. 6.3. In this section initial and final time are determined. Furthermore, the range in which states and controls can vary are detailed.
- **Initial guess:** It is a structure that contains a guess of states, controls, time and integrals of the problem. It is used as an initial guess for the solution. In this thesis the experimental motion is introduced as initial guess to facilitate the convergence of the algorithm.
- **Continuous function:** The continuous function section contains the dynamic constraints (Eq. 6.2) and the algebraic path constraints of Eq. 6.3. Moreover, in this section the integrand of the cost functional (Eq.6.1) to be minimized is determined. Note that the integrand is the function to be integrated over the time interval of the optimization.

It should be noted that states and controls are not defined by the user. They are internally determined from the size of bounds and initial guess. Moreover, once the solution is found, GPOPS-II gives an output structure with the solution of time, states, controls and the minimized cost functional.

In order to solve the optimal control problem, GPOPS-II uses an *hp*-adaptive version of the *Legendre-Gauss-Radau (LGR) orthogonal collocation method* to convert the optimal control problem to a NLP problem. Then, the NLP problem is solved using an existing NLP solver. In this study, the NLP solver is IPOPT, which is an open-source software package for large-scale nonlinear optimization [40]. The flowchart of the GPOPS-II algorithm is shown in Figure 6.2.

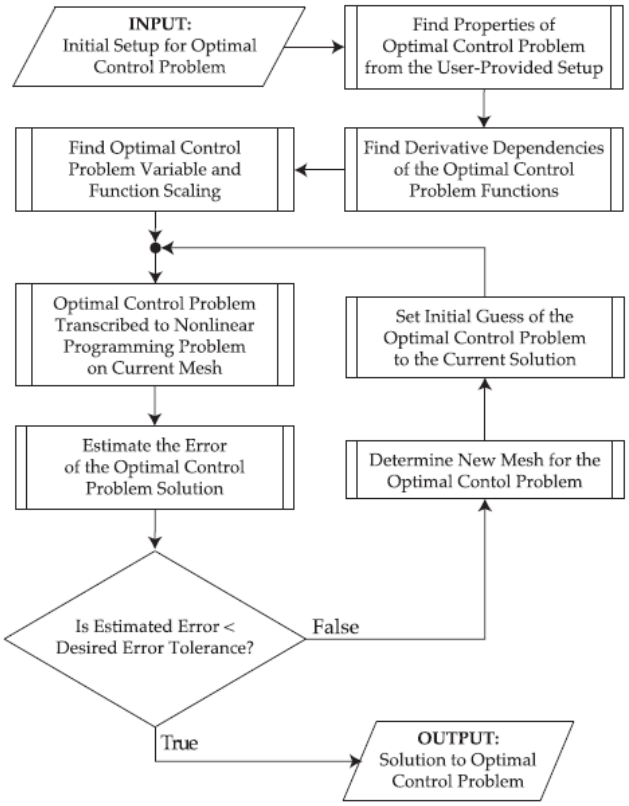


Figure 6.2 Flowchart of the GPOPS-II algorithm. Extracted from [31].

6.2.1. The *Legendre-Gauss-Radau (LGR) orthogonal collocation method*

GPOPS-II uses the *Legendre-Gauss-Radau (LGR) orthogonal collocation method* to transcribe the optimal control problem to a NLP problem. This transcription method is based on LGR points, which are a set of collocation points that are found from the roots of a group of orthogonal polynomials, called Legendre polynomials.

States and controls are approximated with Lagrange interpolating polynomials that use the Legendre-Gauss-Radau points found, plus the terminal point (the last point in the time interval). The cost functional is approximated using a Gauss-Radau quadrature. Then, dynamic and path constraints are discretized and evaluated at the LGR points [22]. It should be noted that dynamic and path constraints are not evaluated at the terminal point. This is due to the fact that LGR points include the initial point of the time interval, but not the final point. However, states and controls are approximated until the end of the interval since a terminal point is explicitly used [24]. This fact must be taken into account when assessing the results.

Note that intuitively, one could think that equidistant points might be suitable to approximate functions. However, they have worse properties than LGR points (which are not equidistant) to discretize functions (see this reference [24] for further information of LGR points). To illustrate this, Figure 6.3 shows the accuracy of an approximation of the function e^t.

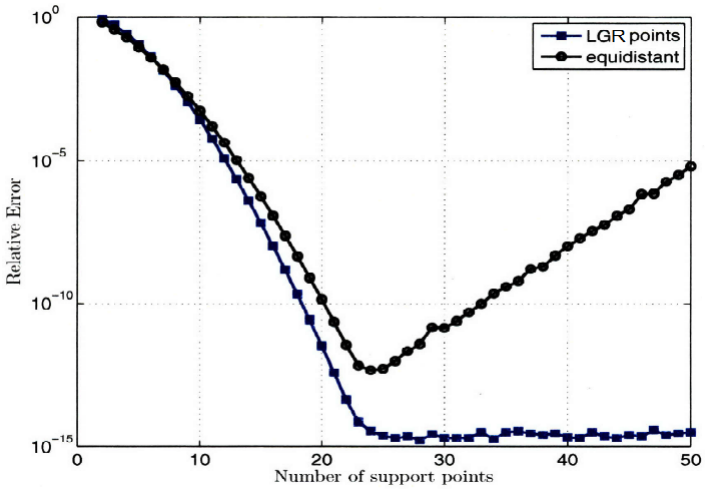


Figure 6.3 Lagrange polynomial approximation of e^t using equidistant and LGR points. The accuracy using equidistant points is lost if number of support points are increased. Adapted from [24].

6.2.2. An *hp*-adaptative method for orthogonal collocation

GPOPS-II divides the time interval forming a mesh that creates a group of reduced intervals, where the orthogonal collocation is applied at each of them. In order to create the mesh, and subsequently refine it if necessary, an *hp*-adaptative method is used [36]. This method varies the number/width of mesh intervals (h), and then, the degree of approximation polynomial (number of collocation points) within each mesh (p). In Figure 6.4 a representation of this method is shown. Recent research suggests that *hp* adaptative methods reduce number of collocation points in approximation, lead to smaller NLP problems and achieve convergence faster [36].

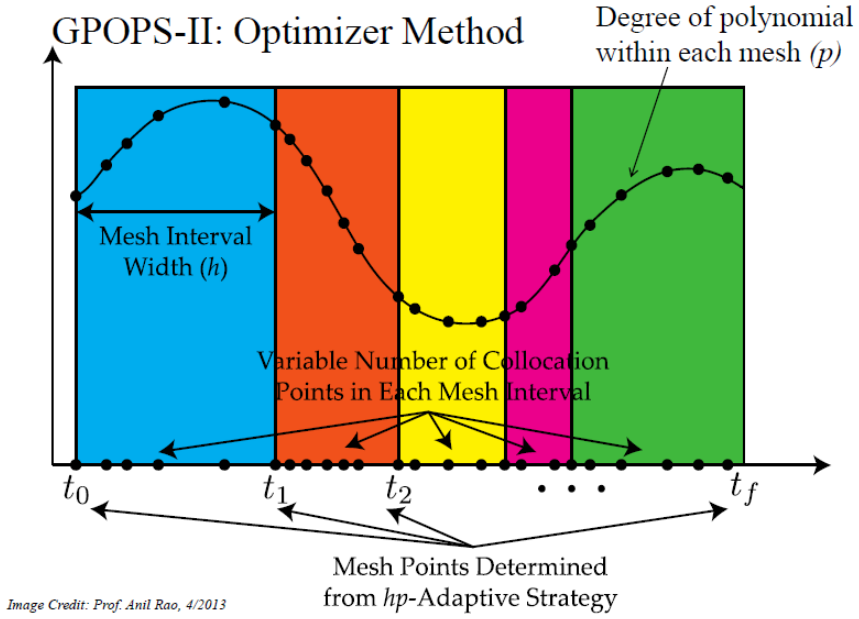


Figure 6.4 *hp*-adaptative method for orthogonal collocation. Mesh width and collocation points are variable. Adapted from [38].

6.3. Simulation methodology

In this section the initial optimal control formulation is presented. Furthermore, the process followed to obtain results through the different analyses is explained. Analyses can be set in two main groups: influence of parameter changes on the solution and study of different optimal control formulations.

6.3.1. Optimal control formulation: tracking workflow

An optimal control formulation is implemented to obtain a dynamically consistent walking motion by reducing the residual wrench. Since the reduction is wanted to perform by finding a similar motion to the captured one, experimental data is always tracked while the algorithm minimizes the residual wrench.

Moreover, in order to take advantage of OpenSim, the optimal control problem is written using the implicit form of the dynamics. This form consists in introducing the equations of motion as path constraints (Eq. 6.3), instead of writing them in the dynamic constraints (Eq. 6.2). Thereby, OpenSim is used in each iteration to solve an inverse dynamic analysis. Then, joint torques of the optimal control problem are imposed to be the same as the resulting joint torques from OpenSim. The following scheme (Figure 6.5) shows the optimal control workflow, in which experimental data is tracked and equations of motion are imposed as path constraints.

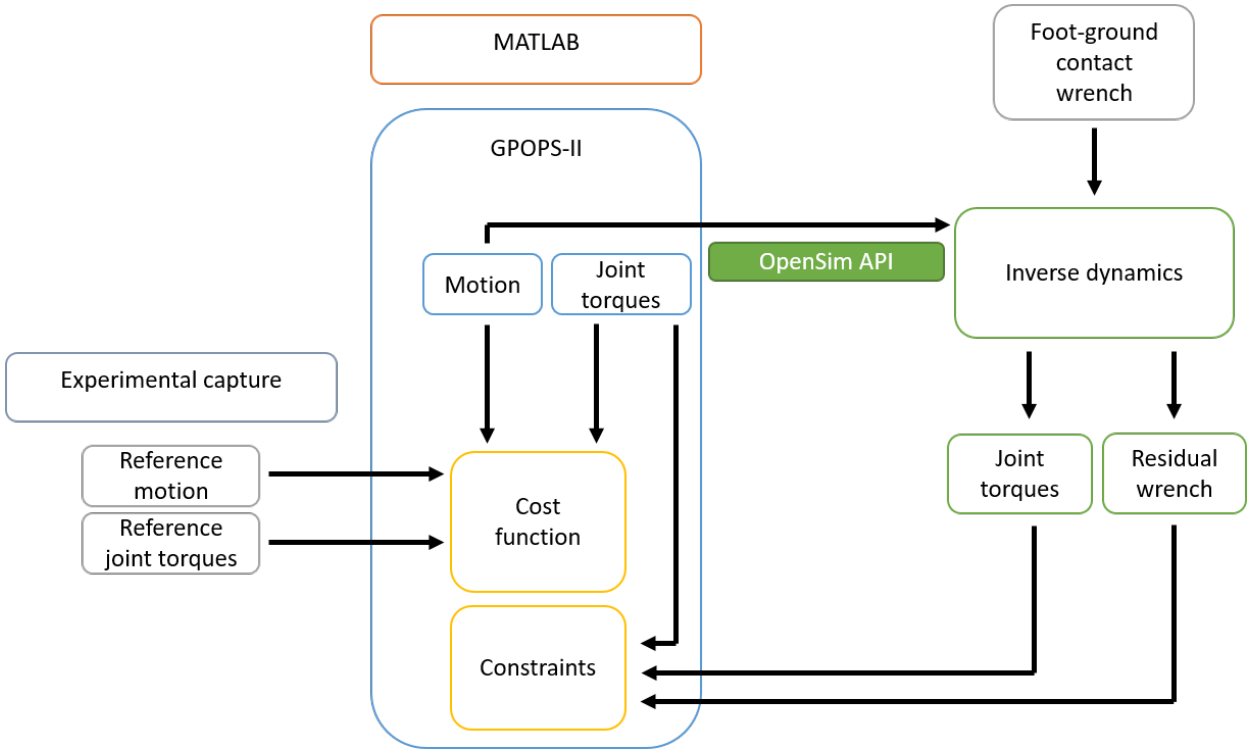


Figure 6.5 Tracking workflow of the optimal control formulation using the implicit form of dynamics.

Below the initial optimal control formulation is presented. It corresponds to the basis of the analysis and is a full tracking formulation, where generalized coordinates, velocities, accelerations and net joint torques are tracked.

States

The state vector, $\mathbf{y} \in \mathbb{R}^{2n}$, is defined as:

$$\mathbf{y} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}, \tag{6.5}$$

where states are the vector of generalized coordinates, $\mathbf{q} \in \mathbb{R}^n$, and the vector of generalized velocities, $\dot{\mathbf{q}} \in \mathbb{R}^n$, of the model.

Controls

The control vector, $\mathbf{u} \in \mathbb{R}^{2n-3}$, is defined as:

$$\mathbf{u} = \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\tau} \end{bmatrix}, \tag{6.6}$$

where controls are the vector of generalized accelerations, $\ddot{\mathbf{q}} \in \mathbb{R}^n$, and the vector of net joint torques, $\boldsymbol{\tau} \in \mathbb{R}^{n-3}$, of the model. Note that the vector of net joint torques is $n-3$ dimensional. This is due to the fact that the other three components are those of the residual wrench of the 2D model applied to the pelvis (Fx, Fy and Mz).

Cost functional

The cost functional to be minimized is the integral of the sum of squared differences between variables of the optimal control problem ($\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ and $\boldsymbol{\tau}$), and values obtained from the experimental capture ($\mathbf{q_{exp}}, \dot{\mathbf{q}}_{\mathbf{exp}}, \ddot{\mathbf{q}}_{\mathbf{exp}}$ and $\boldsymbol{\tau}_{\mathbf{exp}}$). Eq. 7.6 shows the expression of the cost functional, J:

$$J = \int_{t_0}^{t_f} \left(\sum_{i=1}^{n} \left[\left(\frac{q_i - q_{exp_i}}{q_{sf}} \right)^2 + \left(\frac{\dot{q}_i - \dot{q}_{exp_i}}{\dot{q}_{sf}} \right)^2 + \left(\frac{\ddot{q}_i - \ddot{q}_{exp_i}}{\ddot{q}_{sf}} \right)^2 \right] + \sum_{i=1}^{n-3} \left(\frac{\tau_i - \tau_{exp_i}}{\tau_{sf}} \right)^2 \right) \mathrm{d}t, \tag{6.7}$$

being $q_i, \dot{q}_i, \ddot{q}_i$ and τ_i the i^{th} component of the vector $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ and $\boldsymbol{\tau}$, respectively; $q_{exp_i}, \dot{q}_{exp_i}, \ddot{q}_{exp_i}$ and τ_{exp_i} the i^{th} component of the vector $\mathbf{q_{exp}}, \dot{\mathbf{q}}_{\mathbf{exp}}, \ddot{\mathbf{q}}_{\mathbf{exp}}$ and $\boldsymbol{\tau}_{\mathbf{exp}}$, respectively; $q_{sf}, \dot{q}_{sf}, \ddot{q}_{sf}$ and τ_{sf} scale factors for generalized coordinates, velocities, accelerations and joint torques, respectively; and finally, t_0 and t_f the initial and terminal time of the optimal control problem.

Dynamic constraints

Since the implicit form of the dynamics is used (equations of motion are introduced as path constraints), dynamic constraints are simple time derivative relations of the optimal control problem variables:

$$\ddot{\mathbf{q}} = \frac{\mathrm{d}\dot{\mathbf{q}}}{\mathrm{d}t}, \quad \dot{\mathbf{q}} = \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t}. \tag{6.8}$$

Inequality path constraints

In order to introduce equations of motion as path constraints, joint torques from the optimal control problem, $\boldsymbol{\tau}$, are forced to be the same as joint torques obtained from OpenSim, $\boldsymbol{\tau}_{\mathbf{IDA}}$. OpenSim finds these joint torques using the variables of the optimal control problem ($\mathbf{q}, \dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$) as inputs. Furthermore, residual wrench in the pelvis $\mathbf{R_{pelvis}} = [\text{Fx Fy Mz}]^{\text{T}}$ is constrained to be closer to zero. Thereby, inequality path constraints can be written as:

$$-\boldsymbol{\varepsilon}_{\boldsymbol{\tau}} \leq \boldsymbol{\tau} - \boldsymbol{\tau}_{\mathbf{IDA}} \leq \boldsymbol{\varepsilon}_{\boldsymbol{\tau}}, \tag{6.9}$$

$$-\boldsymbol{\varepsilon}_{\mathbf{R}} \leq \mathbf{R_{pelvis}} \leq \boldsymbol{\varepsilon}_{\mathbf{R}}, \tag{6.10}$$

being $\boldsymbol{\varepsilon}_{\boldsymbol{\tau}} \in \mathbb{R}^{n-3}$ and $\boldsymbol{\varepsilon}_{R} \in \mathbb{R}^3$ vectors of a specific tolerance.

Boundary conditions

Initial and terminal time, as well as states evaluated at initial and terminal time, are imposed to be the same as the experimental data. This is done to introduce to GPOPS-II initial values similar to the solution, and in that way, facilitating the convergence of the algorithm. Therefore, boundary conditions are:

$$t_0 = t_{exp_0}, \quad t_f = t_{exp_f}, \tag{6.11}$$

$$\mathbf{y}(t_0) = \begin{bmatrix} \mathbf{q_{exp_0}} \\ \dot{\mathbf{q}}_{\mathbf{exp_0}} \end{bmatrix}, \quad \mathbf{y}(t_f) = \begin{bmatrix} \mathbf{q_{exp_f}} \\ \dot{\mathbf{q}}_{\mathbf{exp_f}} \end{bmatrix}, \tag{6.12}$$

where t_{exp_0} and t_{exp_f} are the initial and terminal time of the captured gait cycle; and $\mathbf{q_{exp_0}}$, $\dot{\mathbf{q}}_{\mathbf{exp_0}}$, $\mathbf{q_{exp_f}}$, $\dot{\mathbf{q}}_{\mathbf{exp_f}} \in \mathbb{R}^n$ are the experimental values of generalized coordinates and velocities obtained from the capture at the initial and terminal time.

6.3.2. Influence of the parameter changes on the solution

In the first set of analyses, parameters of GPOPS-II have been studied. Changing their values, solutions have been assessed in order to find the best set of parameter values to perform the study of different optimal control formulations.

Since these analyses require a large number of simulations, the computational time using the OpenSim model (Section 3.1) would be very high, as it takes a lot of time to call OpenSim libraries and solve the inverse dynamic analysis in each iteration. Therefore, a skeletal model fully implemented in MATLAB, which is faster, has been used in this first group of analyses. This model corresponds to the *2D gait benchmark* [34], which is a 2D bar model with 14 degrees of freedom and 12 rigid bodies. A complete description of this skeletal model can be found in the Appendix, Section A.1. It should also be noted that in order to adapt this model as similar as possible to the OpenSim model, arms and head have been attached to trunk forming a HAT body.

The parameters that have been changed are: IPOPT tolerance, mesh tolerance, path constraint tolerances and amount of data introduced as initial guess. Note that IPOPT tolerance is the allowed

error when solving the NLP problem. On the contrary, mesh tolerance is the permitted error when the optimal control problem is approximated to the NLP problem. The following procedure has been carried out:

- IPOPT tolerance has been changed simultaneously with path constraint tolerances. Residual wrench only at collocation points has been evaluated to select the appropriate IPOPT and path constraint tolerances.
- Mesh tolerance has been varied to find the best result of residual wrench in all the frames of the gait cycle. This tolerance has been chosen to obtain the minimal residual wrench in a feasible computational time.
- The number of frames of the experimental data introduced as initial guess have been modified. Convergence of GPOPS-II has been apprised to find the suitable number of frames to introduce as initial guess.
- Once the above mentioned parameters have been selected, the initial optimal control formulation has been solved in order to evaluate the algorithm.

6.3.3. Study of different optimal control formulations

With the chosen set of parameter values and using the OpenSim skeletal model, different optimal control formulations have been analyzed. The goal has been to find a formulation that minimizes the residual wrench with the least computational time possible, and with resulting states and controls as close to experimental data as possible.

First, the initial optimal control formulation has been solved again. The solution has been compared with the previous analysis, in which the *2D gait benchmark* skeletal model has been used. Next, different strategies have been followed to formulate the optimal control problem:

- Convergence of GPOPS-II has been assessed by changing squared differences for absolute differences and differences to the fourth power in the cost functional.
- Jerks, the third time derivative of generalized coordinates, have been added to the cost functional. As proposed in [37, 28], jerks have been minimized to apprise if convergence and smoothness of states are improved.
- Kinematic variables have been removed from the cost functional one by one. It has been evaluated if a full tracking is needed or if tracking a specific kinematic variable is enough.
- Net joint torques have been removed from the cost functional. Then, they have also been removed from path constraints. It has been checked if they are needed to be tracked or even included in the optimal control algorithm.

Results from the different strategies have been studied by analyzing the plots of joint coordinates and torques, as well as the residual wrench obtained. It should be noted that unlike the OpenSim model, in the *2D gait benchmark* model, the coordinate that orientates the trunk is absolute (see the Appendix, Section A.1 for further information). Nevertheless, when analyzing the full tracking solution with this model, this coordinate has been also introduced, as it is an important coordinate to analyze in the results and to compare with the solution obtained with the OpenSim model.

Moreover, in order to compare the solution of the initial formulation using both models, as well as the solutions of the different strategies performed, differences between coordinates or torques of the optimal control problem, and coordinates or torques obtained from the experimental capture, have been quantified by calculating the Root Mean Square Error (RMSE) as Eq. 6.13 shows:

$$RMSE(x_i) = \sqrt{\frac{1}{K}\sum_{j=1}^{K}(x_{i,j} - x_{exp_{i,j}})^2}, \tag{6.13}$$

being K the total number of frames of the gait cycle; $x_{i,j}$ the i^{th} component of a variable vector of the optimal control problem $\mathbf{x} \in \{\mathbf{q}, \boldsymbol{\tau}\}$ at the j^{th} frame; and $x_{exp_{i,j}}$ the i^{th} component of the corresponding reference vector obtained from the experimental capture $\mathbf{x_{exp}} \in \{\mathbf{q_{exp}}, \boldsymbol{\tau}_{\mathbf{exp}}\}$ at the j^{th} frame.

In both studies (the "influence of parameter changes on the solution" and the "study of different optimal control formulations"), the optimal control problem has been solved with a maximum of 4000 iterations, on an initial grid with 10 mesh intervals and 4 LGR collocation points in each interval. Furthermore, optimal control variables have been scaled to achieve better convergence. They have been automatically scaled by GPOPS-II. At last, all the optimizations have been carried out using a DELL Precision 3420 computer (Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz, 8.00 GB RAM).

7. RESULTS & DISCUSSION

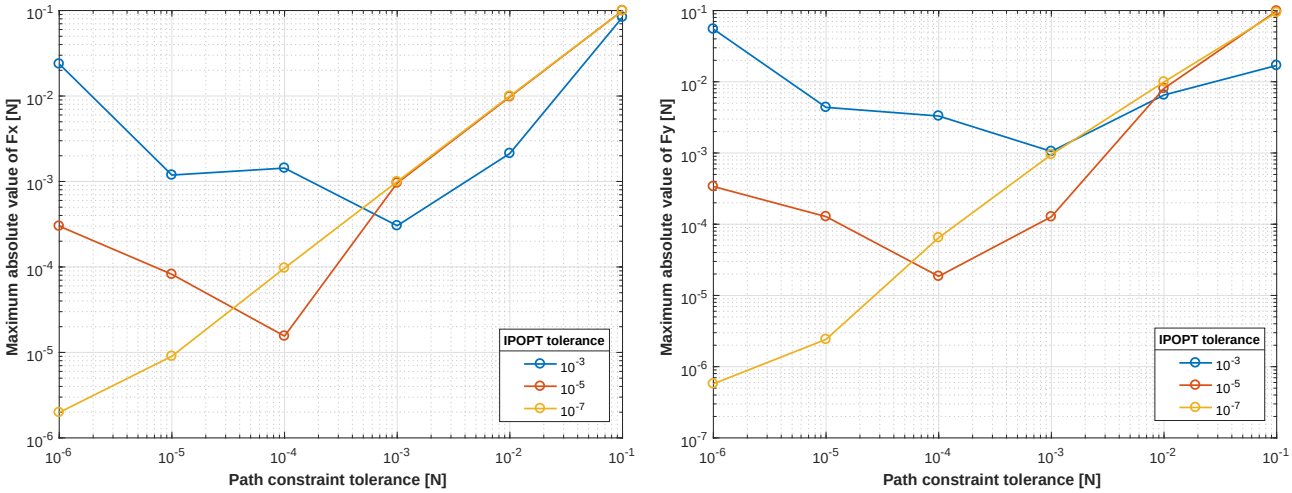
Results of the different analyses that have been performed through GPOPS-II are presented. First, the influence of GPOPS-II parameters is assessed. Then, solutions for different optimal control formulations are shown and discussed.

7.1. Influence of parameter changes on the solution

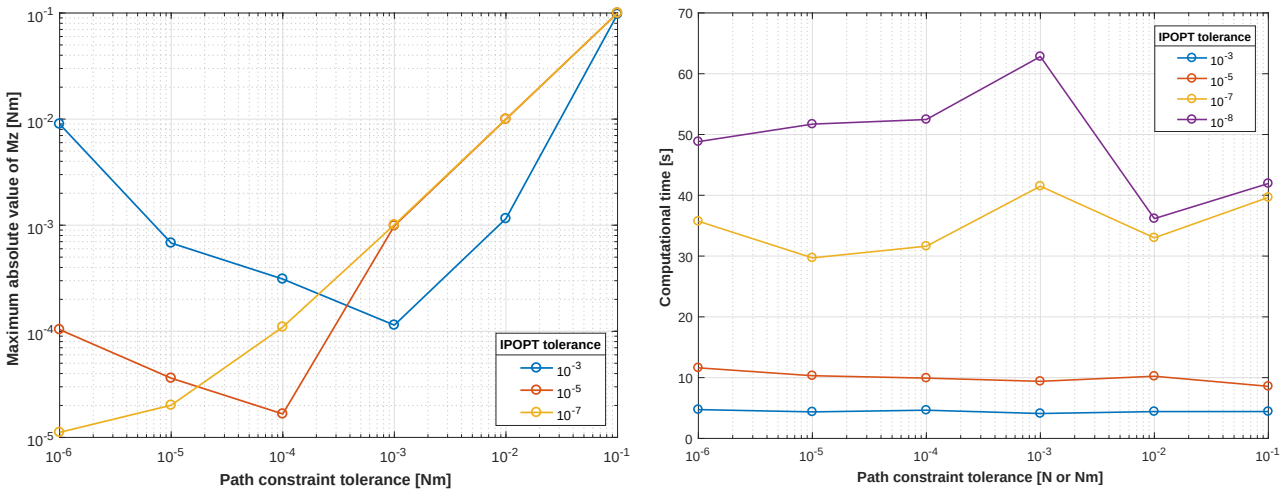
In this first set of analyses results of the modification of IPOPT tolerance, path constraint tolerance, mesh tolerance and initial guess are presented. After evaluating their influence on the solution, the most suitable values are pointed out.

7.1.1. IPOPT and path constraint tolerances

IPOPT and path constraint tolerances have been varied simultaneously. It has been found that depending on the value of IPOPT tolerance, path constraint tolerance is not accomplished. In order to evaluate this issue, residual wrench only at collocation points has been studied. To do so, the maximum absolute value of residual components Fx, Fy and Mz over all the solution has been taken from different combination of IPOPT and path constraint tolerances. IPOPT tolerance has been varied from 10^{-3} to 10^{-8}, path constraint tolerance has been varied from 10^{-1} to 10^{-6} N or Nm. In Figure 7.1 the results are shown.



(a) (b)



(c) **(d)**

Figure 7.1 Results of residual wrench depending on IPOPT and path constraint tolerances. In plots of maximum absolute Fx, Fy and Mz, IPOPT tolerance of 10^{-8} is not plotted since the trace is the same as IPOPT tolerance of 10^{-7}. (7.1a) Maximum absolute residual force Fx, (7.1b) maximum absolute residual force Fy, (7.1c) maximum absolute residual moment Mz, (7.1d) computational time.

As said at the beginning of this section, it can be seen that depending on the chosen IPOPT tolerance, path constraint tolerance is not fulfilled. For example, as seen in Figures 7.1a, 7.1b and 7.1c, if an IPOPT tolerance of 10^{-3} and a path constraint tolerance of 10^{-4} N or Nm are chosen, values larger than the path constraint tolerance appear. In general, the lower the path constraint tolerance with respect to the IPOPT tolerance is, the worse the results are. Therefore, in order to accomplish path constraints, IPOPT tolerance should be lower than path constraint tolerance. However, IPOPT tolarances much lower than path constraint tolerances are not appropriate. IPOPT tolerances close to path constraint tolerances show better results (maximum absolute values are lower), and the algorithm spends less time finding the solution.

Regarding the computational time, as seen in Figure 7.1d, the lower the IPOPT tolerance is, the more the program takes to converge. Moreover, for IPOPT tolerances of 10^{-3} and 10^{-5}, computational time is nearly independent of path constraints tolerance. On the contrary, for IPOPT tolerances of 10^{-7} and 10^{-8}, computational time depends on path constraint tolerance.

Taking into account all the points discussed above, an IPOPT tolerance of 10^{-5} and a path constraint tolerance of 10^{-4} N or Nm have been chosen. Selecting these pair of tolerances, path constraints are fulfilled and the algorithm takes $9,4$ s to converge. In Figure 7.2, the resulting residual wrench at collocation points is shown. It can be seen that peaks are smaller than the path constraint tolerance of 10^{-4} N or Nm. Moreover, note that the residual wrench does not reach the 100% of the gait cycle, as LRG points do not include the terminal point of the time interval (as explained in section 6.2.1).

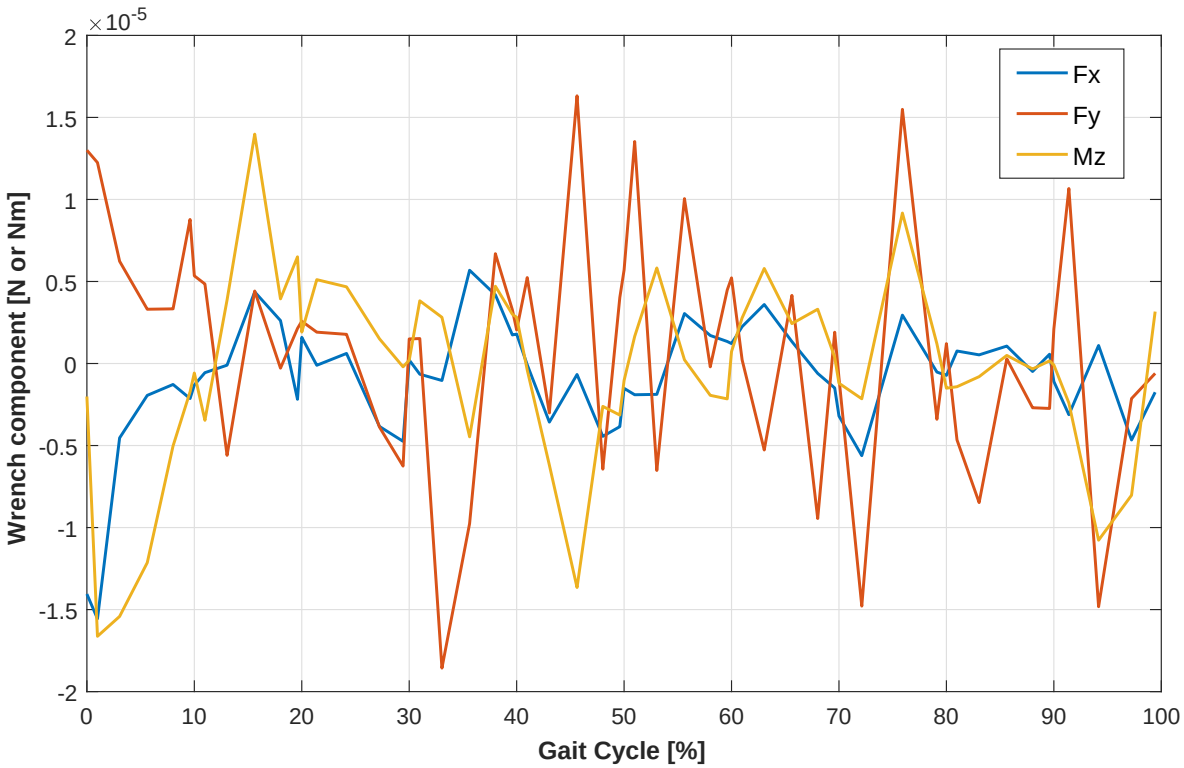


Figure 7.2 Residual wrench evaluated at collocation points for an IPOPT tolerance of 10^{-5} and a path constraint tolerance of 10^{-4}.

7.1.2. Mesh tolerance

Mesh tolerance has been modified to obtain the minimal residual wrench over all the frames of the experimental capture, in a reasonable computational time. This tolerance has been varied within a range from 10^{-2} to 10^{-7}. Figures 7.3 and 7.4 show the resulting residual wrench, number of collocation points and computational time.

Since frames of the experimental capture do not correspond to collocation points, the residual wrench is larger than the one evaluated at collocation points. However, by decreasing mesh tolerance, residual wrench over all the frames of the experimental capture can be reduced. This is due to the fact that for lower mesh tolerances, the NLP problem has to be better approximated to the optimal control problem. Thereby, more collocation points are used and path constraints are satisfied in more points of the gait cycle.

As shown in Figure 7.3, for a mesh tolerance of 10^{-2} and 10^{-3}, the residual wrench is the same. This is because of the tolerance is accomplished with the initial set of collocation points (4 within each of the 10 mesh intervals). Nevertheless, from a tolerance of 10^{-4}, more collocation points are needed to better approximate the optimal control problem to the NLP problem, and consequently, the residual wrench is minimized.

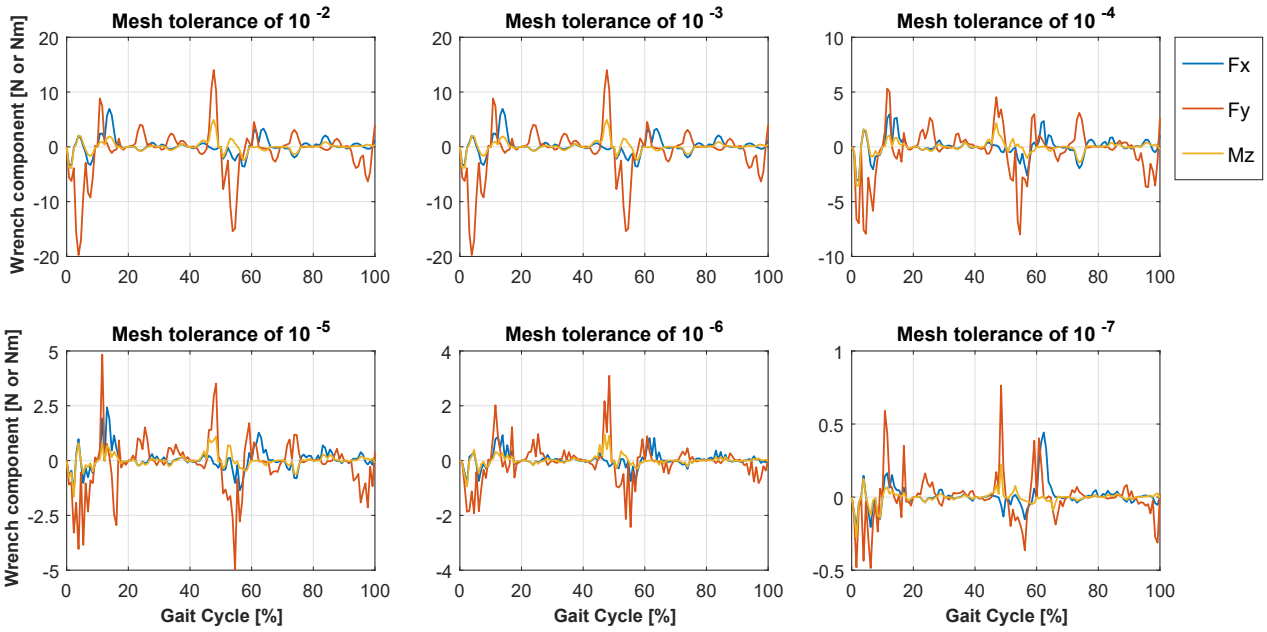
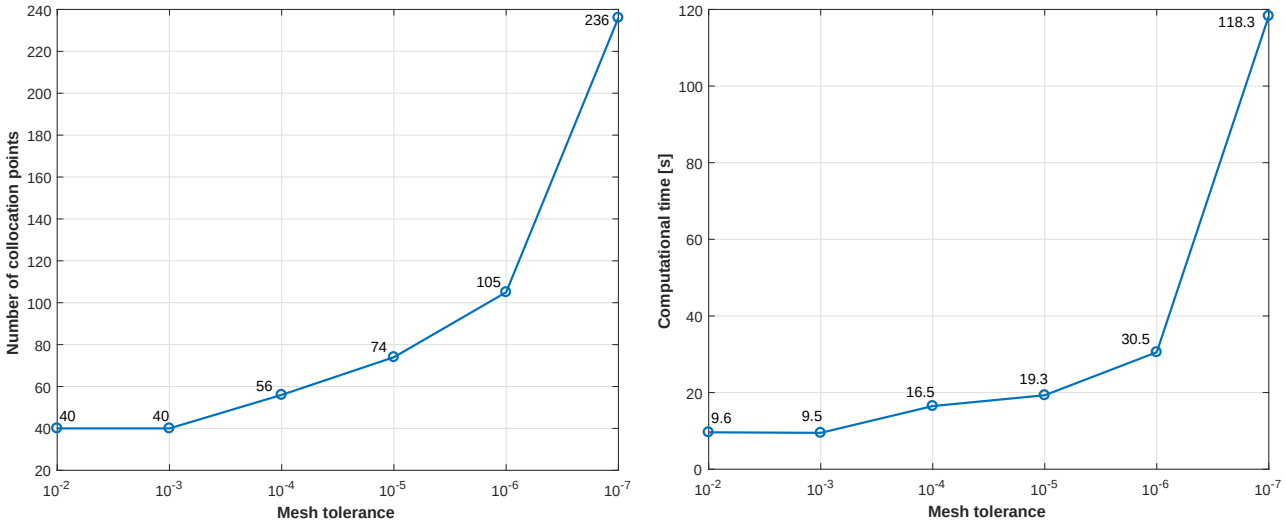


Figure 7.3 Residual wrench evaluated at frames of experimental capture for different values of mesh tolerance.



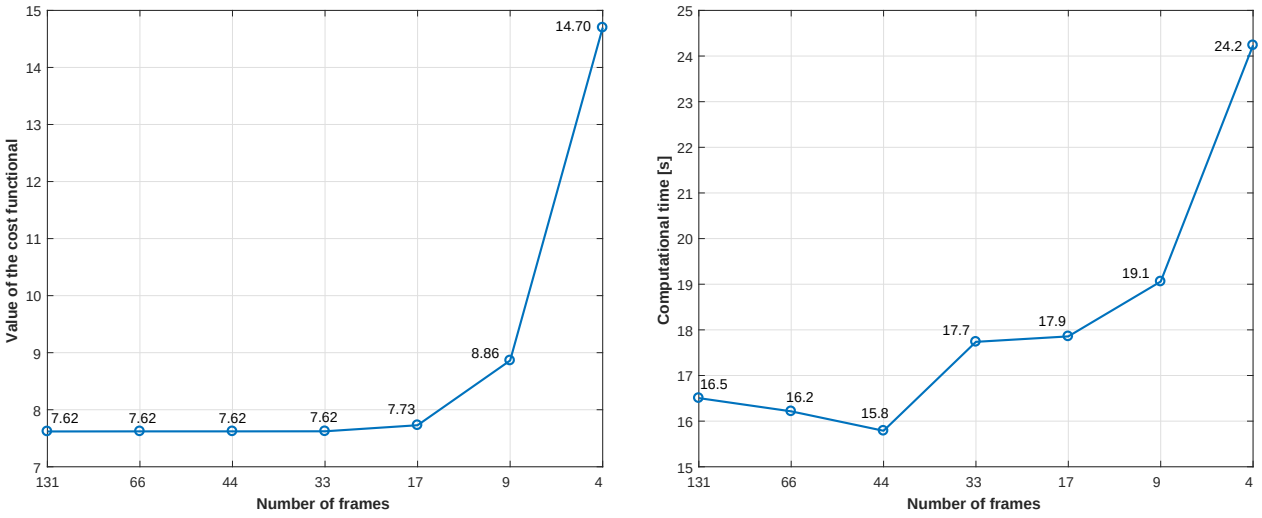
(a) **(b)**

Figure 7.4 Number of collocation points used to approximate the optimal control problem (7.4a) and computational time (7.4b) depending on the mesh tolerance.

Taking into account that is wanted to obtain the minimal residual wrench possible, but in a reasonable computational time, mesh tolerances of 10^{-4} and 10^{-5} seem to be the more appropriate. However, giving more importance to time, since when using the OpenSim model the computational time is highly increased, a mesh tolerance of 10^{-4} has been chosen. Using this tolerance the residual wrench is comprised between $-8,0$ and $5,3$ N or Nm, 56 collocation points are used and the optimal control problem takes $16,5$ s to be solved.

7.1.3. Initial guess

Convergence of GPOPS-II has been studied to find the suitable initial guess. To do so, number of frames of the experimental data introduced as initial guess for the solution have been changed. Then, computational time and cost functional value have been evaluated in each solution. Note that the cost functional of the initial optimal control formulation (Eq. 7.6) can be viewed as a global parameter that quantifies the similarity between optimal control variables and experimental data. The number of frames that have been taken are the following: 131 frames (all of them), 66 frames (one out of two), 44 frames (one out of three), 33 frames (one out of four), 17 frames (one out of eight), 9 frames (one out of 16) and 4 frames (one out of 33). Figure 7.5 shows the results for each initial guess.



(a) **(b)**

Figure 7.5 Cost functional value (7.5a) and computational time (7.5b) depending on the frames introduced as initial guess.

It can be seen that introducing all the frames as initial guess is not the best option for convergence. On the other hand, introducing a number of frames similar as the initial number of collocation points used, makes the program to find the solution faster. As shown in Figure 7.5, the best solution is to introduce 44 frames of experimental data as initial guess. Thereby, the optimal control problem converges in $15,8$ s. Note that if the number of frames is reduced, the program starts to spend more time solving the problem, and finds a solution that is worse. To the point that with less than 4 frames as initial guess, the optimal control problem do not converge.

7.1.4. Full tracking formulation

Results for the initial optimal control formulation have been obtained by using the chosen values of the previous analyses: IPOPT tolerance of 10^{-5}, path constraint tolerance of 10^{-4} N or Nm, mesh tolerance of 10^{-4} and 44 frames of experimental data as initial guess. Below, the resulting

generalized coordinates (Figure 7.6) and joint torques (Figure 7.7) are shown and compared with experimental data. Also, the minimized residual wrench is presented in Figure 7.8. In the Appendix (Section A.2.1), results of generalized velocities and accelerations can be found as well.

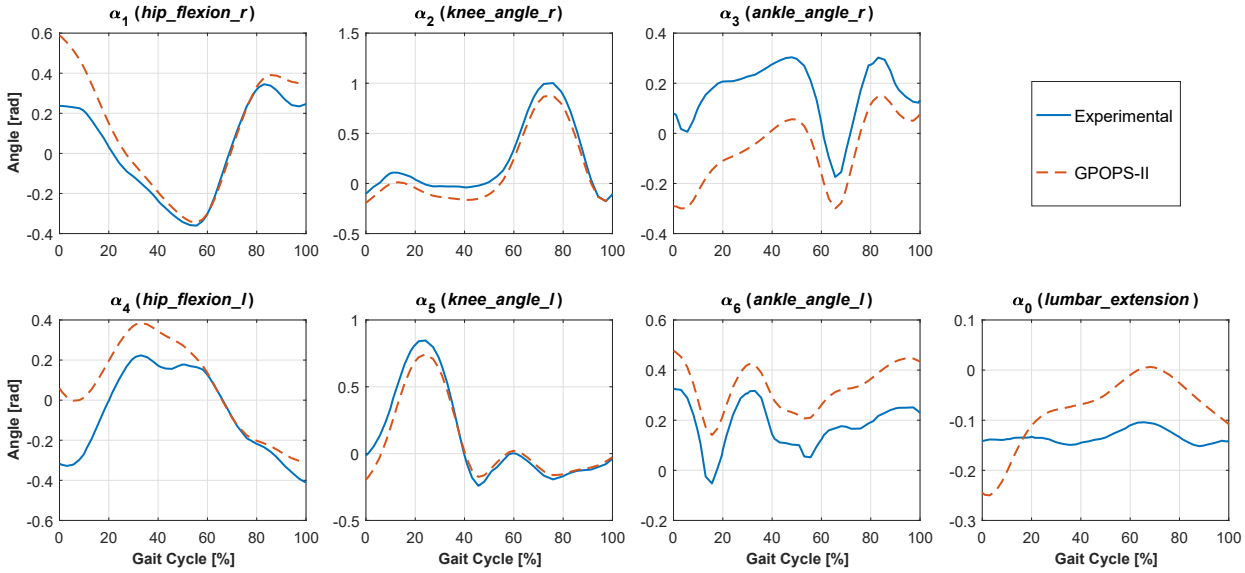


Figure 7.6 Comparison of generalized coordinates obtained from GPOPS-II and from the experimental capture. Relative coordinates and absolute lumbar coordinate are shown. The symbol above each plot corresponds to the symbol used to name each generalized coordinate in the *Gait 2D benchmark* model (See Figure A.1 of the Appendix). In brackets the corresponding name used in the OpenSim model.

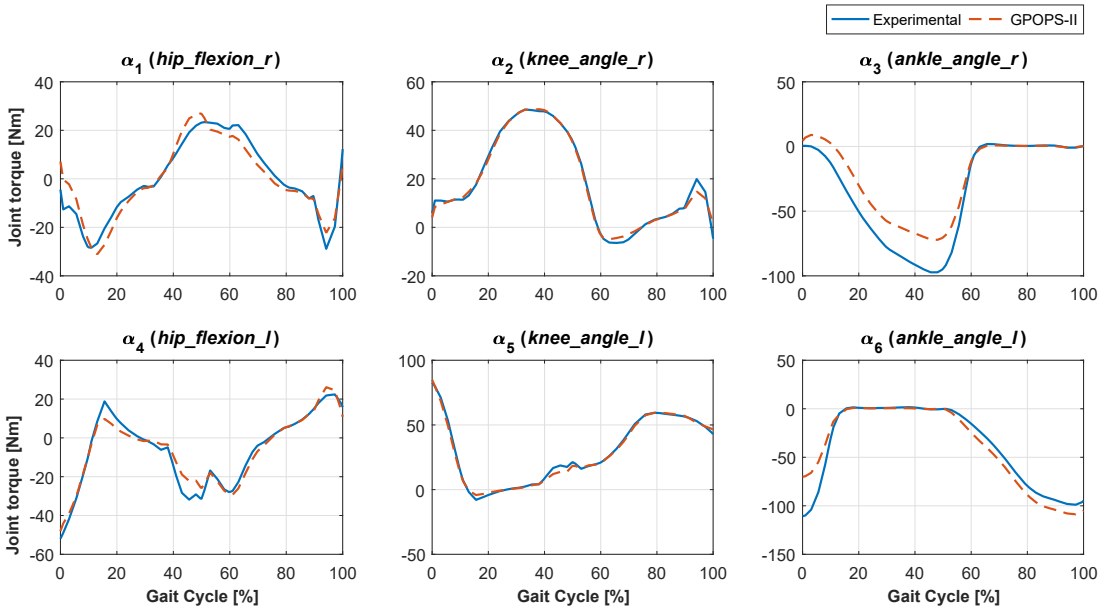


Figure 7.7 Comparison of joint torques obtained from GPOPS-II and from the experimental capture. The symbol above each plot corresponds to the symbol used to name each generalized coordinate in the *Gait 2D benchmark* model (See Figure A.1 of the Appendix). In brackets the corresponding name used in the OpenSim model. Note that the torque of lumbar coordinate is not shown, as this coordinate is absolute and its actuator represents the residual moment Mz, which is minimized.

The optimal solution has been found in $15,8$ s and 56 collocation points have been used. Figure 7.6 shows the new predicted motion by GPOPS-II, which accomplishes path constraints and reduces residual forces and moments. *Lumbar_extension*, the generalized coordinate that orientates the HAT body, is one of the variables that more changes its tendency with respect to the coordinate obtained from experimental data. Also, both "hip flexion" coordinates vary at the first half of the gait cycle, since at the beginning the residual wrench is larger. Finally, both "ankle angle" coordinates are shifted downwards (right foot) and upwards (left foot) in all the gait cycle. In Section 7.2.1, an explanation of why this behavior of generalized coordinates might occur is given.

Regarding joint torques, they are very similar to the experimental ones. As it can be seen in Figure 7.7, in general, their values are further from the experimental ones in the periods where the corresponding coordinates are more different from the captured motion.

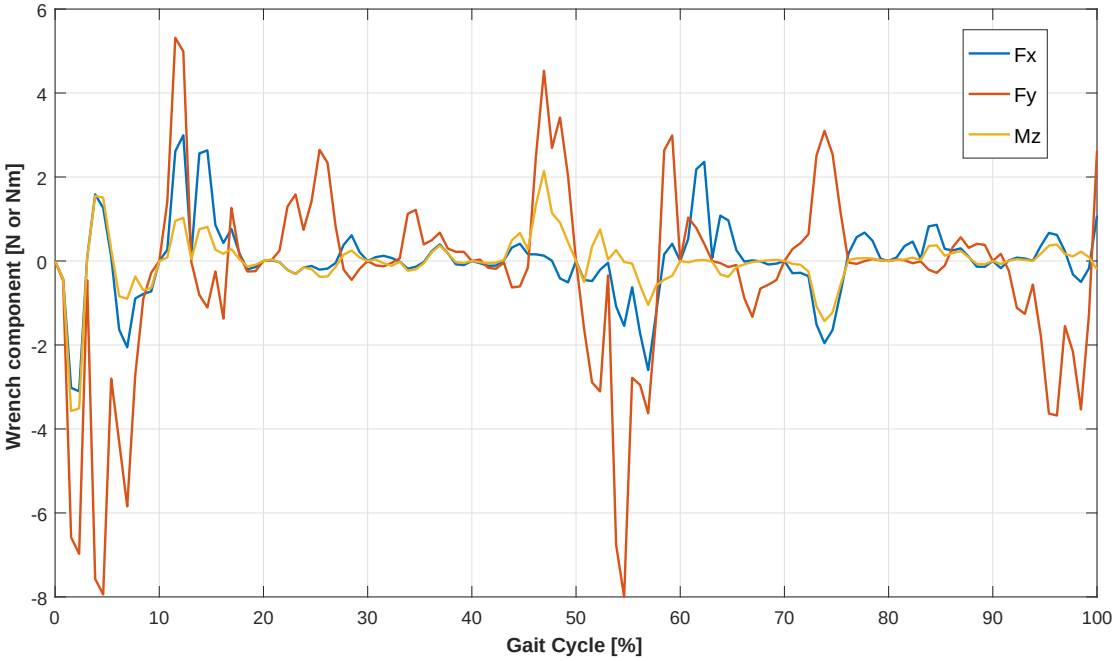


Figure 7.8 Residual wrench obtained from the new predicted motion using the full tracking formulation.

The residual wrench is highly minimized with respect to the one obtained before applying optimal control techniques (Figure 5.9). At the beginning, residual components varied between -115 and 50 N or Nm. Now, after applying the optimal control algorithm, residual wrench components vary between $-8,0$ and $5,3$ N or Nm. Concretely, vertical component Fy is the one that has the maximum and minimum peaks. Taking the maximum absolute value of this component, $8,0$ N, and considering a value of 9,8 m/s^2 of gravity, the maximum error committed can be calculated. Recalling that the subject's weight is 65 kg, the maximum error, e_{max}, with respect to its weight is:

$$e_{max}[\%] = \frac{8,0}{65 \times 9,8} \times 100 = 1,3\%, \tag{7.1}$$

which is an error acceptable when analyzing human gait in biomechanics [15].

7.2. Study of different optimal control formulations

The second group of analyses consists on studying variations of the initial full tracking formulation, using the OpenSim skeletal model, and then assess results in order to chose a final optimal control formulation. First, the full tracking is solved again, in order to point out differences from using the *Gait 2D benchmark* model or the OpenSim model in the optimal control algorithm. Next, different cost functionals are performed and a final formulation is presented.

7.2.1. Full tracking formulation

The full tracking solution has been solved again with the tolerances and initial guess chosen before. First, the resulting generalized coordinates (Figure 7.9) and joint torques (Figure 7.10) are shown and compared with experimental data. Then, in order to compare the results between both models, the RMSE of generalized coordinates is calculated for both solutions. Finally, the minimized residual wrench is presented in Figure 7.11. In the Appendix (Section A.2.2), results of generalized velocities and accelerations can be also found.

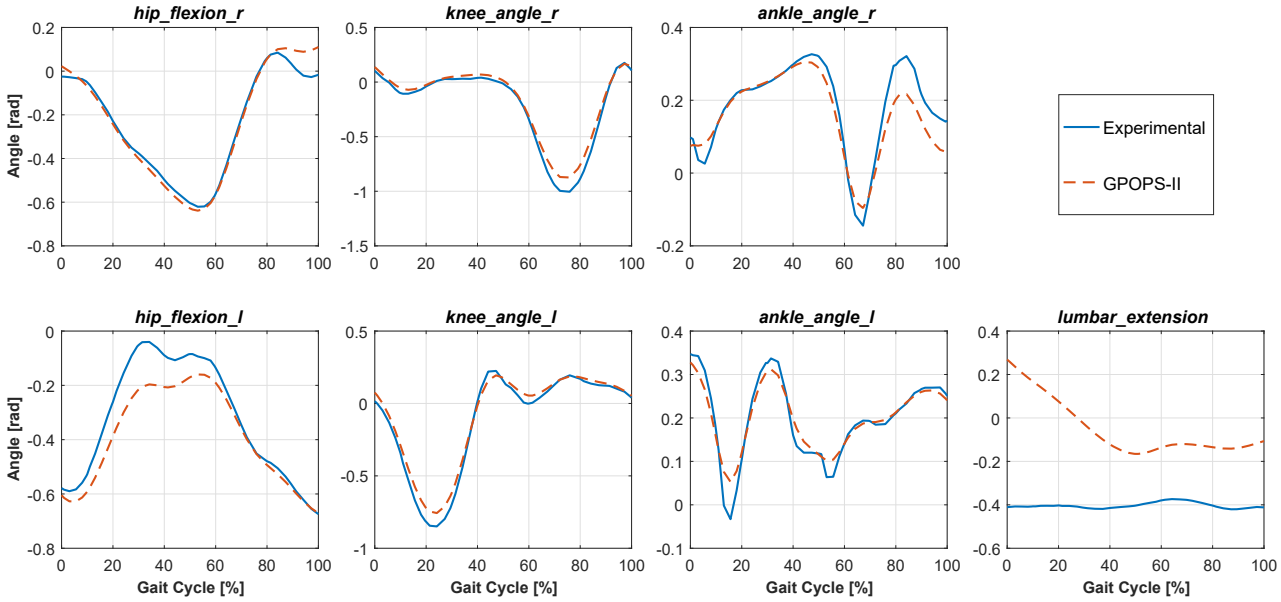


Figure 7.9 Comparison of generalized coordinates obtained from GPOPS-II and from the experimental capture. Joint coordinates of the OpenSim skeletal model are shown.

The optimal solution has been found in $598,3$ s, which are almost 10 minutes, and 52 collocation points have been used. Now, the computational time is nearly 38 times the spent time when using the model fully implemented in Matlab. This is because of the algorithm has to call OpenSim library and load the model in each iteration. With regard to coordinates, as can be seen in Figure 7.9, the new predicted motion is very similar in lower limbs. However, *lumbar_extension* coordinate is quite changed in order to minimize residuals, it goes from $0,28$ to $-0,18$ rad. At the beginning,

when the residual wrench is larger, is more different. Then, it gets closer to the experimental value. Moreover, as shown in Figure 7.10, *lumbar_extension* joint torque is the one that changes the most with respect to the experimental one, since the corresponding coordinate is quite different as well.

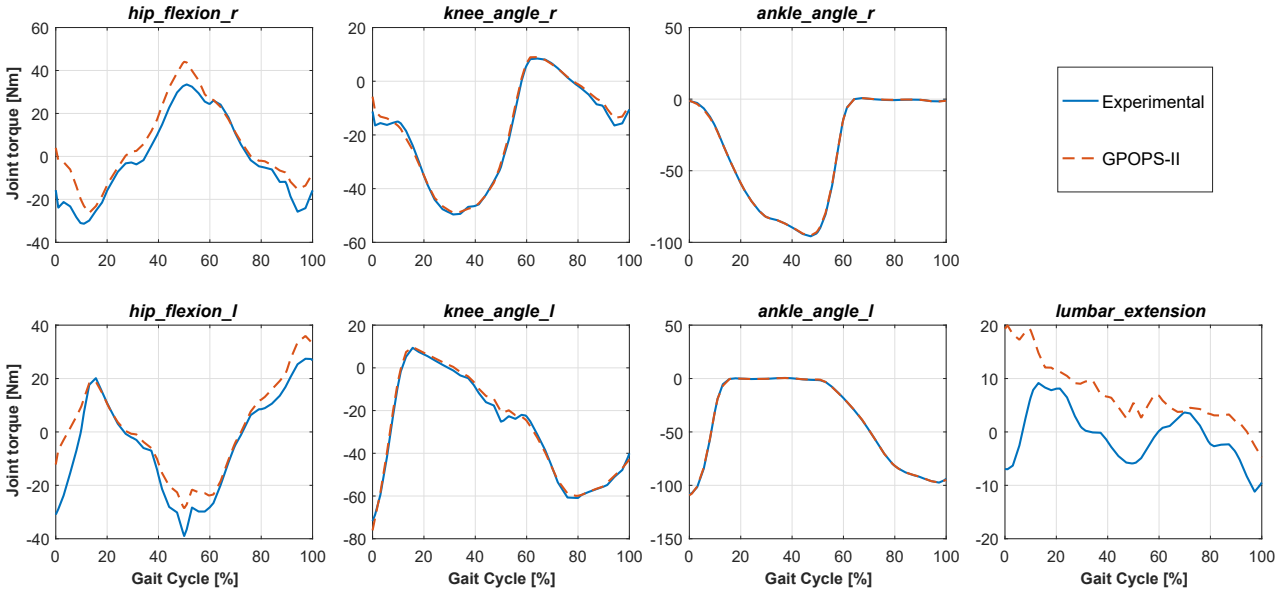


Figure 7.10 Comparison of joint torques obtained from GPOPS-II and from the experimental capture.

In order to compare the performance of both models, the RMSE of coordinates for both solutions is presented in Table 7.1. It can be seen that lower limb coordinates are more similar when using the OpenSim model, but *lumbar_extension* is more different. This might be related to the main difference that these two models have: the pelvis. As the OpenSim model has pelvis, *lumbar_extension* is a relative coordinate. Thereby, the HAT body can be oriented independently from lower limbs bodies. Then, as this body is the one with more mass, changes on its orientation causes more effect on the solution and the residual wrench can be easily minimized.

On the other hand, since the *Gait 2D benchmark* model does not have pelvis, the *lumbar_extension* coordinate is used as absolute. This causes that when changing this coordinate, not only the orientation of HAT body is changed, but lower limb bodies are moved as well. Thus, when the algorithm tries to reduce residuals by changing the orientation of HAT, it has to change lower limb coordinates too, resulting in a solution in which all the coordinates are more different from experimental data.

Model	*hip_flexion_r* [rad]	*knee_angle_r* [rad]	*ankle_angle_r* [rad]	*hip_flexion_l* [rad]	*knee_angle_l* [rad]	*ankle_angle_l* [rad]	*lumbar_extension* [rad]
Gait 2D	0,1233	0,1025	0,2426	0,1639	0,0815	0,1508	0,0832
OpenSim	0,0428	0,0581	0,0480	0,0846	0,0518	0,0256	0,3843

Table 7.1 RMSE of coordinates obtained from the full tracking formulation. Results from the *Gait 2D benchmark* model and the OpenSim model are presented.

Finally, Figure 7.11 shows the residual wrench. It can be observed that is very similar to the one obtained with the *Gait 2D benchmark*. However, in this case positive peaks are a little bit higher. Vertical force, which is still the residual wrench component that changes the most, varies between $-8,0$ and $8,5$ N. This might be caused by the lower number of collocation points used: now 52 instead of 56. However, an increase of $0,5$ N on the maximum value of the vertical component does not change the $1,3\%$ of maximum error, e_{max}, with respect to the subject's weight.

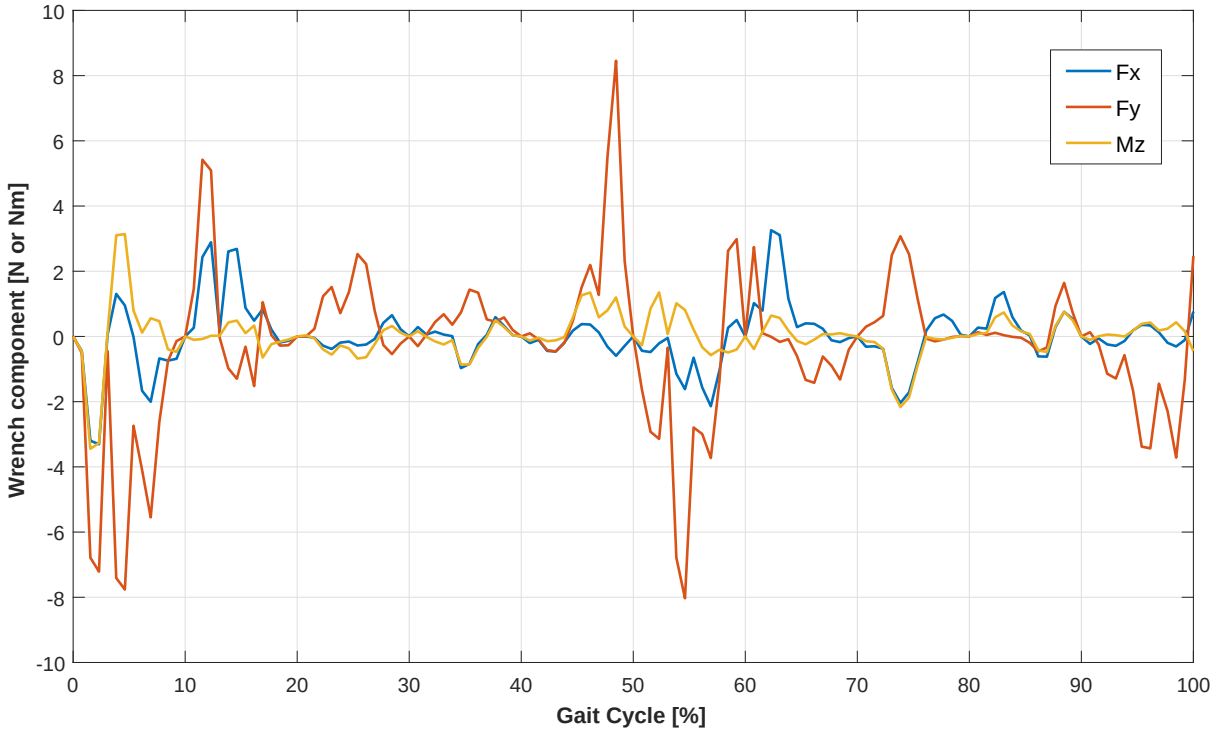


Figure 7.11 Residual wrench obtained from the new predicted motion using the full tracking formulation.

7.2.2. Evaluation of convergence using differences to other powers

In order to evaluate how the algorithm converges using other powers, the square difference terms of the cost functional (Eq. 7.6) have been changed to differences to the fourth power and absolute value of differences. Always making differences to be positive. Otherwise, the algorithm may minimize cost functional to high negative values, instead of near zero. The new cost functional formulations are respectively:

$$J = \int_{t_0}^{t_f} \left(\sum_{i=1}^{n} \left[\left(\frac{q_i - q_{exp_i}}{q_{sf}} \right)^4 + \left(\frac{\dot{q}_i - \dot{q}_{exp_i}}{\dot{q}_{sf}} \right)^4 + \left(\frac{\ddot{q}_i - \ddot{q}_{exp_i}}{\ddot{q}_{sf}} \right)^4 \right] + \sum_{i=1}^{n-3} \left(\frac{\tau_i - \tau_{exp_i}}{\tau_{sf}} \right)^4 \right) \mathrm{d}t, \qquad (7.2)$$

and

$$J = \int_{t_0}^{t_f} \left(\sum_{i=1}^{n} \left[\left\| \frac{q_i - q_{exp_i}}{q_{sf}} \right\| + \left\| \frac{\dot{q}_i - \dot{q}_{exp_i}}{\dot{q}_{sf}} \right\| + \left\| \frac{\ddot{q}_i - \ddot{q}_{exp_i}}{\ddot{q}_{sf}} \right\| \right] + \sum_{i=1}^{n-3} \left\| \frac{\tau_i - \tau_{exp_i}}{\tau_{sf}} \right\| \right) \mathrm{d}t. \qquad (7.3)$$

Cost functional with absolute value of differences has not converged. The 4000 iterations of IPOPT have been reached without finding a solution. On the contrary, cost functional with differences to the fourth power has converged. The optimal solution has been found in $7117,7$ s, which are nearly 2 hours, and 51 collocation points have been used. Results for residual wrench, torques, generalized velocities and accelerations have been practically the same as the solution presented in the previous section. However, as shown in Figure 7.12 and Table 7.2, generalized coordinates are worse than the full tracking solution with square differences.

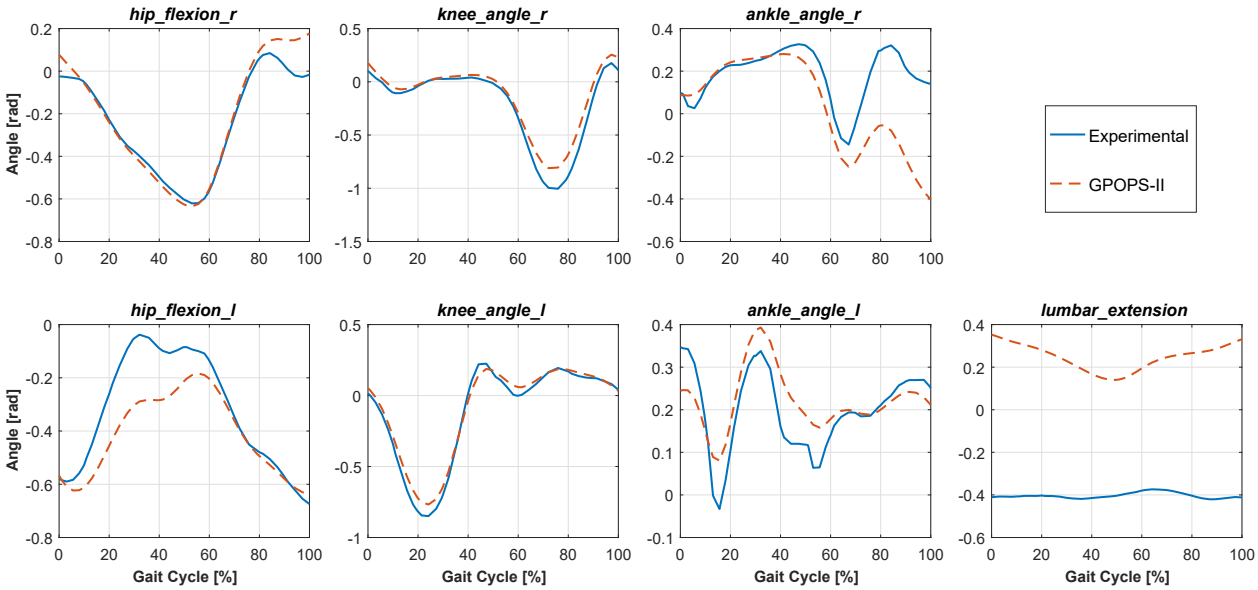


Figure 7.12 Comparison of generalized coordinates obtained from GPOPS-II and from the experimental capture. Joint coordinates of the OpenSim skeletal model are shown.

Power	*hip_flexion_r* [rad]	*knee_angle_r* [rad]	*ankle_angle_r* [rad]	*hip_flexion_l* [rad]	*knee_angle_l* [rad]	*ankle_angle_l* [rad]	*lumbar_extension* [rad]
Second	0,0428	0,0581	0,0480	0,0846	0,0518	0,0256	0,3843
Fourth	0,0672	0,0963	0,2279	0,1260	0,0501	0,0635	0,6567

Table 7.2 RMSE of generalized coordinates. Comparison between formulations of square differences and differences to the fourth power.

As can be observed, the RMSE is worse for all the coordinates. Specially, the *ankle_angle_r* and *lumbar_extension* coordinates. It can be concluded that differences to the fourth power and absolute value of differences on the cost functional are not appropriate for convergence. The second formulation have not converged, the first one has taken almost 2 hours to converge and results for coordinates have been worse. Cost functional with square differences seems to be the most suitable. This might be associated to the fact that the optimal control formulation with squared differences is similar to a quadratic programming (QP) problem, which is a specific NLP problem easier to solve than a general NLP problem.

7.2.3. Assessment of convergence and smoothness using jerks

Jerks, the third time derivative of generalized coordinates ($\dddot{\mathbf{q}} \in \mathbb{R}^n$), have been added to the problem. Convergence of the algorithm and smoothness of accelerations have been studied by minimizing them on the cost functional. When adding jerks as a variable of the optimal control problem, the state vector and the control vector change as follows:

$$\mathbf{y} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} \in \mathbb{R}^{3n}, \qquad \mathbf{u} = \begin{bmatrix} \dddot{\mathbf{q}} \\ \boldsymbol{\tau} \end{bmatrix} \in \mathbb{R}^{2n-3}. \tag{7.4}$$

Then, the new cost functional formulation is:

$$J = \int_{t_0}^{t_f} \left(\sum_{i=1}^{n} \left[\left(\frac{q_i - q_{exp_i}}{q_{sf}} \right)^2 + \left(\frac{\dot{q}_i - \dot{q}_{exp_i}}{\dot{q}_{sf}} \right)^2 + \left(\frac{\ddot{q}_i - \ddot{q}_{exp_i}}{\ddot{q}_{sf}} \right)^2 + \left(\frac{\dddot{q}_i}{\dddot{q}_{sf}} \right)^2 \right] + \sum_{i=1}^{n-3} \left(\frac{\tau_i - \tau_{exp_i}}{\tau_{sf}} \right)^2 \right) \mathrm{d}t. \tag{7.5}$$

The optimal solution has been found in $28540,0$ s, which are almost 8 hours, and 85 collocation points have been used. This time has been considered unfeasible for finding a solution, and it has been found that is because of the difficulties of the algorithm to reach the mesh tolerance of 10^{-4}. Therefore, mesh tolerance has been reduced to 10^{-3} when adding jerks. With this new tolerance, the solution has been found in $2397,7$ s, almost 40 minutes, and 60 collocation points have been used. Results of generalized coordinates (Figure 7.13), the RMSE of coordinates (Table 7.3) and the *lumbar_extension* acceleration (Figure 7.14) are presented. Results of torques, generalized velocities and all the accelerations can be found in the Appendix (Section A.2.3).

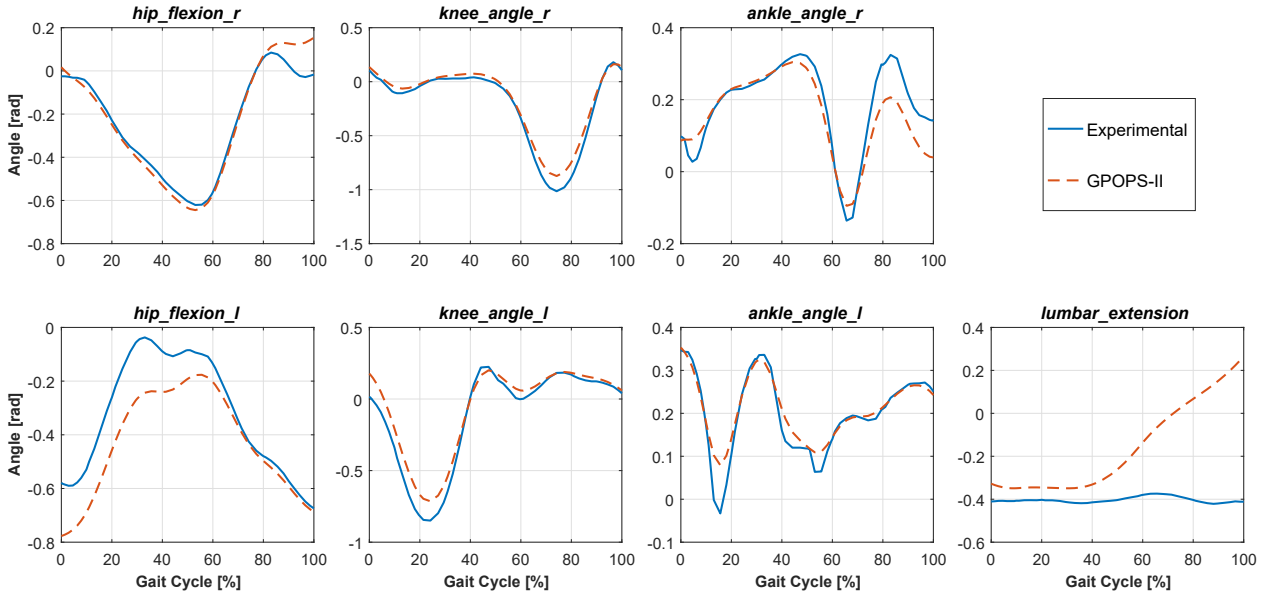
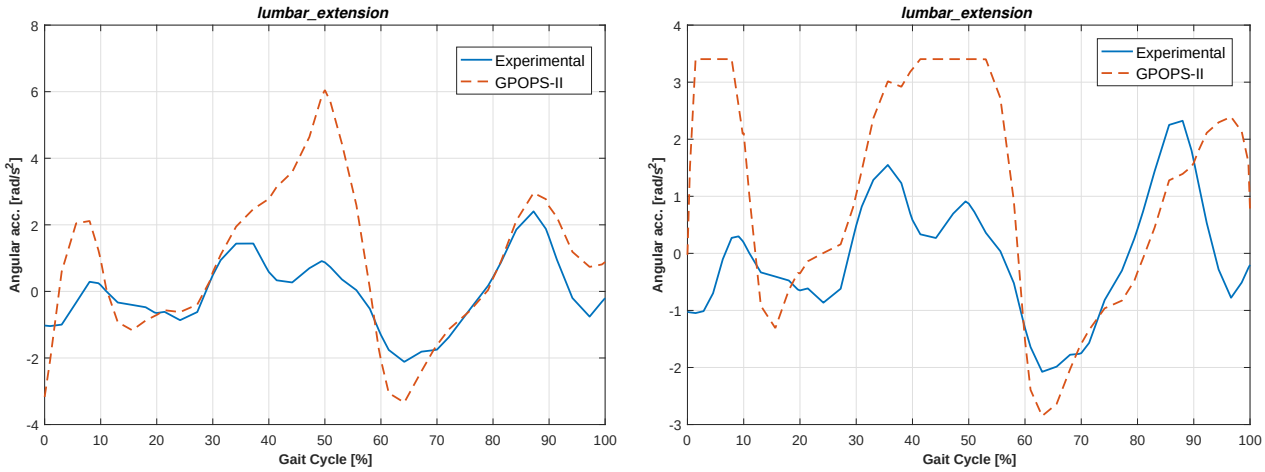


Figure 7.13 Comparison of generalized coordinates obtained from GPOPS-II and from the experimental capture. Joint coordinates of the OpenSim skeletal model are shown.

Jerks	*hip_flexion_r* [rad]	*knee_angle_r* [rad]	*ankle_angle_r* [rad]	*hip_flexion_l* [rad]	*knee_angle_l* [rad]	*ankle_angle_l* [rad]	*lumbar_extension* [rad]
Without	0,0428	0,0581	0,0480	0,0846	0,0518	0,0256	0,3843
With	0,0595	0,0648	0,0568	0,1305	0,0885	0,0281	0,3240

Table 7.3 RMSE of generalized coordinates. Comparison between the full tracking solution without using jerks and with jerks.

As observed in Table 7.3, in general, the RMSE of coordinates is slightly larger when jerks are added to the cost functional. However, for the *lumbar_extension* coordinate results are better. The RMSE is lower, and as shown in Figure 7.13, the evolution is different from the solution without using jerks. The coordinate starts being similar to the experimental one, at a value of $-0,35$ rad, and then, it increases reaching the $0,2$ rad, just the contrary that happens to the full tracking formulation without jerks.



(a) **(b)**

Figure 7.14 *lumbar_extension* acceleration: 7.14a without adding jerks, 7.14b with minimization of jerks.

The *lumbar_extension* acceleration is the one that changes the most when adding jerks to the optimal control problem (in the Appendix it can be seen that the other accelerations are very similar for both solutions). Without jerks, the acceleration has a maximum of 6 rad/s^2. On the contrary, with jerks, this maximum peak is reduced and smoothed to $3,4$ rad/s^2.

In conclusion, minimizing jerks on the cost functional has a clear effect on the accelerations. Nevertheless, the convergence is not improved and the mesh tolerance has to be reduced, as the algorithm has more difficulties to approximate the optimal control problem to the NLP problem. Since complexity of this skeletal model is low and the accelerations do not tend to oscillate, a positive effect in convergence of jerks may not be appreciated in this specific problem. It would be interesting to analyze the addition of jerks in a problem with more complexity or with oscillating accelerations in the solution.

7.2.4. Variations with kinematic variables

Kinematic variables from the cost functional (Eq. 7.6) have been removed one by one or in pairs. It has been assessed if a full tracking formulation of kinematic variables may be redundant. Also, it has been checked which kinematic variables are the most important for convergence and for tracking the experimental motion. To do so, the tracking formulation has been reduced by removing from the cost functional, one by one: coordinates, velocities and accelerations. And then, in pairs: coordinates and velocities, coordinates and accelerations and velocities and accelerations. Computational time and the RMSE of generalized coordinates have been evaluated to identify the best formulations. Results are shown in Tables 7.4 and 7.5.

Removed Variables	-	Coordinates	Velocities	Accelerations	Coordinates & velocities	Coordinates & accelerations	Velocitites & accelerations
Computational time [s]	598,3	578,5	678,2	12462,2	588,7	2287,9	6523,4

Table 7.4 Computational time spent by the algorithm for the different formulations. The hyphen represents the full tracking formulation in which no kinematic variables have been removed.

Removed Variables	*hip_flexion_r* [rad]	*knee_angle_r* [rad]	*ankle_angle_r* [rad]	*hip_flexion_l* [rad]	*knee_angle_l* [rad]	*ankle_angle_l* [rad]	*lumbar_extension* [rad]
-	0,0428	0,0581	0,0480	0,0846	0,0518	0,0256	0,3843
Coordinates	0,0437	0,0576	0,0492	0,0844	0,0515	0,0256	0,4320
Velocitites	0,0434	0,0583	0,0541	0,0865	0,0526	0,0257	0,3372
Accelerations	0,0800	0,0557	0,0354	0,0734	0,0647	0,0273	0,6724
Coordinates & velocitites	0,0444	0,0579	0,0564	0,0857	0,0522	0,0258	0,3625
Coordinates & accelerations	0,0820	0,0591	0,0414	0,0763	0,0620	0,0272	0,6703
Velocities & accelerations	0,2411	0,3552	0,0770	0,0578	0,2175	0,1603	0,6498

Table 7.5 RMSE of generalized coordinates. Comparison of results among the different optimal control formulations, in which kinematic variables have been removed.

Taking a look on the results, it can be concluded quickly that accelerations are always necessary. Convergence without tracking accelerations is highly worsened, and therefore the computational time increases a lot. Also, if accelerations are removed, results of generalized coordinates are more different from experimental data, in all the cases *lumbar_extension* coordinate is worse compared with the full tracking solution. The worst solution is the one in which accelerations and velocities have been removed. In this case, computational time is increased to $6523,4$ s, and the RMSE of the most of the coordinates is highly rised, as for instance the RMSE of *ankle_angle_r* coordinate that in the full tracking is $0,0581$ rad and now $0,3552$ rad.

On the other hand, removing coordinates and/or velocities have a positive impact on the solution. In general, resulting generalized coordinates are similar to experimental data. If coordinates are removed, the computational time decreases from $598,3$ s to $578,5$ s, but it has the drawback that the RMSE of *lumbar_extension* coordinate increases from $0,3843$ rad to $0,4320$ rad. If velocities are removed, the RMSE of *lumbar_extension* is reduced to $0,3372$ rad, but the computational time is augmented to $678,2$ s. Then, if both coordinates and velocities are removed, both positive effects are obtained: computational time is slightly reduced to $588,7$ s and the RMSE decreases to $0,3625$ rad. It seems that an optimal control formulation in which only accelerations and torques are tracked is a good alternative to the full tracking formulation.

7.2.5. Variations with dynamic variables

Maintaining all kinematic variables, torques have been removed from the cost functional (Eq. 7.6), in order to study if they are needed in both the cost functional and in path constraints. Then, they have been also deleted from path constraints. This second formulation corresponds to an exclusive kinematic tracking, in which torques are not a design variable of the optimal control problem, and the inverse dynamic analysis is performed only to obtain the residual wrench. Therefore, path constraints are reduced to Eq. 6.10, as the inverse dynamics constraint (Eq. 6.9) dissapears. It should be noted that this formulation is only possible when tracking experimental motion, since in general, when a foot-ground contact model is used to predict a new motion, a minimization of torques, muscle activations or muscle excitations is performed.

This second formulation has been carried out in order to assess if the algorithm is able to reduce the residual wrench, and at the same time, obtain kinematic variables very close to the experimental ones. Thus, the torques obtained from these kinematic variables should be similar to the experimental ones. Both formulations have been assessed by analyzing the computational time, the RMSE of torques and at last, the RMSE of generalized coordinates. Tables 7.6, 7.7 and 7.8 show the results.

Torques removed in	**-**	**Cost functional**	**Cost functional + constraints**
Computational time [s]	598,3	1433,0	482,4

Table 7.6 Computational time spent by the algorithm for the different formulations. The hyphen represents the full tracking formulation in which torques have not been removed.

Removing torques from cost functional worsens the convergence of the problem, increasing the computational time of the algorithm. This might be caused by an increase of difficulty when fulfilling path constraints (implicit form of dynamics), since torques are not tracked and their value

might be less guided. On the contrary, if torques are not design variables, the computational time is reduced more than 100 seconds. This happens because the algorithm has less design variables and one less path constraint.

Torques removed in	*hip_flexion_r* torque [Nm]	*knee_angle_r* torque [Nm]	*ankle_angle_r* torque [Nm]	*hip_flexion_l* torque [Nm]	*knee_angle_l* torque [Nm]	*ankle_angle_l* torque [Nm]	*lumbar_extension* torque [Nm]
-	8,2668	1,6453	0,4010	7,5608	1,9912	0,3716	9,8529
Cost functional	13,2100	8,1613	0,6694	20,5977	14,6607	1,5615	2,8091
Cost functional + path constraints	12,5748	12,3826	1,3738	17,0030	23,6485	2,4834	5,8084

Table 7.7 RMSE of joint torques. Comparison of results among the different optimal control formulations, in which torques have been gradually removed.

Torques removed in	*hip_flexion_r* [rad]	*knee_angle_r* [rad]	*ankle_angle_r* [rad]	*hip_flexion_l* [rad]	*knee_angle_l* [rad]	*ankle_angle_l* [rad]	*lumbar_extension* [rad]
-	0,0428	0,0581	0,0480	0,0846	0,0518	0,0256	0,3843
Cost functional	0,0511	0,0441	0,0896	0,0599	0,0606	0,0900	0,0736
Cost functional + path constraints	0,0511	0,0481	0,0253	0,0564	0,0575	0,0243	0,0665

Table 7.8 RMSE of generalized coordinates. Comparison of results among the different optimal control formulations, in which torques have been gradually removed.

It can be seen that in both cases, when torques are removed from the cost functional or from the cost functional and path constraints, the RMSE of the resulting torques increases. Except for the RMSE of *lumbar_extension* torque, which decreases. Regarding coordinates, they are very close to the experimental ones, being closer in the case that torques are removed from both cost functional and path constraints. It is remarkable that the RMSE of *lumbar_extension* coordinate decreases significantly, it goes from $0,3843$ rad to $0,0665$ rad, becoming the resulting coordinate very similar to the experimental one.

In conclusion, removing torques only in the cost functional is not a good strategy, since convergence is worse and the computational time increases. On the contrary, removing torques from both cost functional and path constraints, makes the algorithm to converge faster and focus more on tracking the experimental motion. If the price of obtaining torques more different from the experimental ones is accepted, this formulation allows to find a solution, in a shorter time, that minimizes residuals with resulting coordinates very similar to the captured motion.

7.2.6. Final optimal control formulation

Once all the different strategies have been studied, a final optimal control formulation has been chosen. First, it has been checked whether a formulation that combines both strategies of removing kinematic variables, and torques could be optimal or not. It has been found that in all the cases, computational time increases. And in general, resulting coordinates are less similar to the experimental motion. It has been concluded that strategies of removing kinematic variables or torques have to be performed separately. Thereby, two final alternative formulations have been considered.

Formulation 1

This first formulation corresponds to a tracking of accelerations and torques strategy, in which coordinates and velocities are removed from the cost functional. States, controls, dynamic constraints, path constraints and boundary conditions remain the same as the initial optimal control formulation (Section 6.3.1). With regard to the cost functional, it becomes the following expression:

$$J = \int_{t_0}^{t_f} \left(\sum_{i=1}^{n} \left(\frac{\ddot{q}_i - \ddot{q}_{exp_i}}{\ddot{q}_{sf}} \right)^2 + \sum_{i=1}^{n-3} \left(\frac{\tau_i - \tau_{exp_i}}{\tau_{sf}} \right)^2 \right) \mathrm{d}t, \tag{7.6}$$

Formulation 2

The second formulation corresponds to a full kinematic tracking strategy, in which torques are removed from the cost functional and from path constraints, *i.e.*, they are not a variable of the optimal control problem. States, dynamic constraints and boundary conditions remain the same as the initial optimal control formulation (Section 6.3.1). Regarding controls, cost functional and path constraints, they change. The control vector only contains the accelerations:

$$\mathbf{u} = [\ddot{\mathbf{q}}] \in \mathbb{R}^n. \tag{7.7}$$

The cost functional does not contain a tracking term of torques:

$$J = \int_{t_0}^{t_f} \left(\sum_{i=1}^{n} \left[\left(\frac{q_i - q_{exp_i}}{q_{sf}} \right)^2 + \left(\frac{\dot{q}_i - \dot{q}_{exp_i}}{\dot{q}_{sf}} \right)^2 + \left(\frac{\ddot{q}_i - \ddot{q}_{exp_i}}{\ddot{q}_{sf}} \right)^2 \right] \right) \mathrm{d}t. \tag{7.8}$$

Path constraints only contain the residual wrench, as the implicit form of dynamics is not performed in the algorithm:

$$-\boldsymbol{\varepsilon}_{\mathbf{R}} \leq \mathbf{R}_{\mathbf{pelvis}} \leq \boldsymbol{\varepsilon}_{\mathbf{R}}, \tag{7.9}$$

Results

The resulting generalized coordinates (Figure 7.15) and joint torques (Figure 7.16) are shown for both solutions and compared with experimental data. Also, the minimized residual wrench is presented in Figure 7.17. Only one plot is presented for both solutions as the resulting residual wrench is the same. In the Appendix (Section A.2.4), results of generalized velocities and accelerations can be found as well.

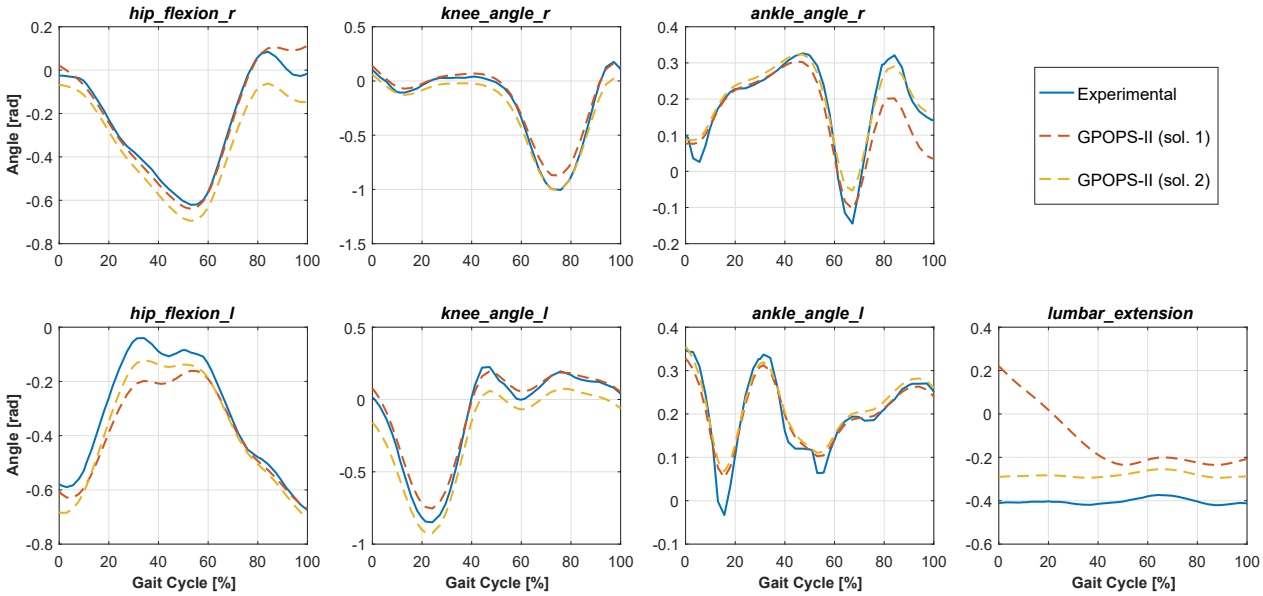


Figure 7.15 Comparison of generalized coordinates obtained from both solutions of GPOPS-II and from the experimental capture. Joint coordinates of the OpenSim skeletal model are shown.

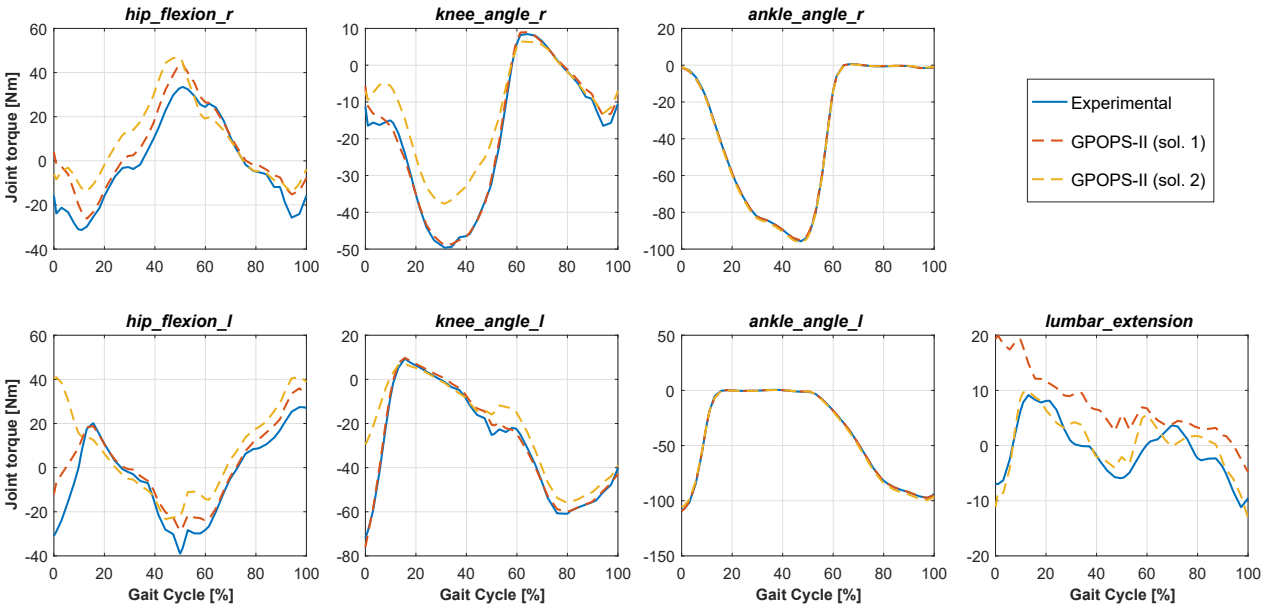


Figure 7.16 Comparison of joint torques obtained from both solutions of GPOPS-II and from the experimental capture.

For the first formulation, the optimal solution has been found in $588,7$ s and 52 collocation points have been used. For the second formulation, the algorithm converged in $482,4$ s and 52 collocation points have been used too. Regarding generalized coordinates, both formulations have very good results of lower limb coordinates. However, comparing the *lumbar_extension* coordinate in both solutions, it can be appreciated that when using the second formulation, the resulting generalized coordinate (sol. 2) is much better than the resulting generalized coordinate of the first formulation

(sol. 1). In the second solution, this coordinate is always similar to the experimental one, with an approximate value of $-0,28$ rad. Whereas in the first solution, it starts at $0,2$ rad, before it gets closer to the experimental coordinate.

With respect to torques, in general, the results obtained from the second formulation (sol. 2) are more different from the experimental ones than the results obtained from the first formulation (sol. 1). It can be remarked the *hip_flexion_l* torque, which in the second solution it starts at 40 Nm, while the experimental one is -35 Nm. Nevertheless, the *lumbar_extension* torque is more similar in the second solution than in the first one, since the corresponding coordinate is more similar to the experimental motion as well.

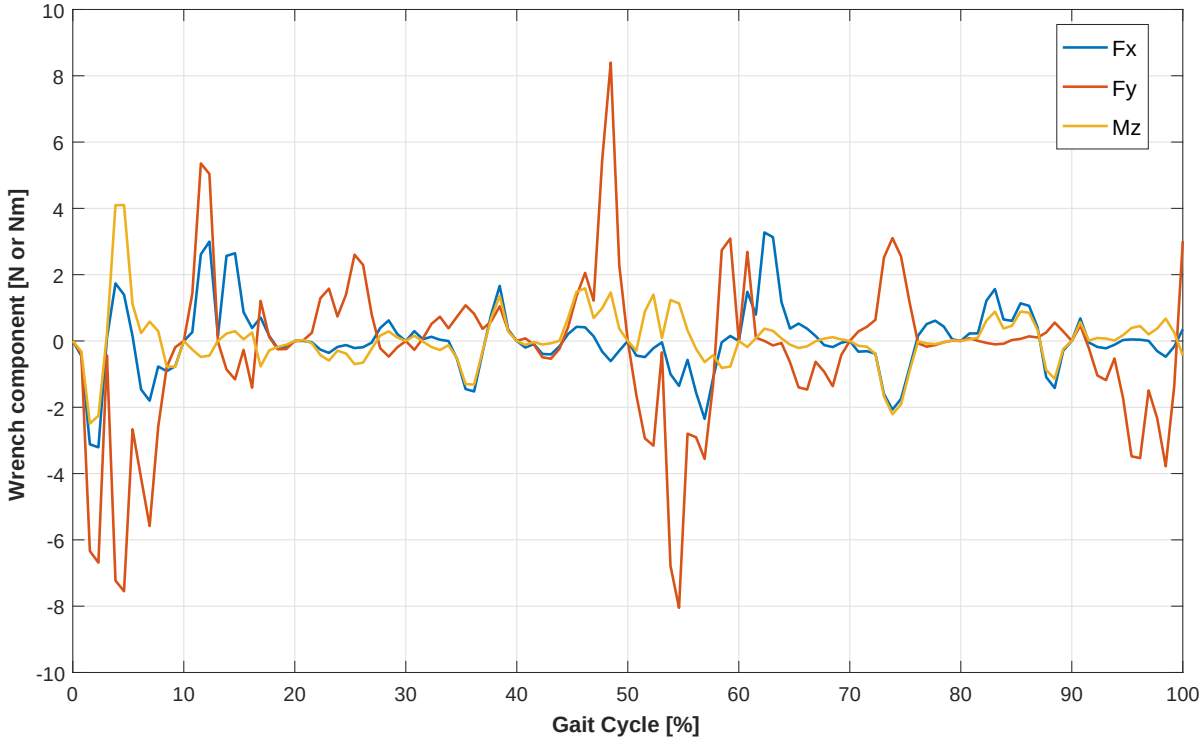


Figure 7.17 Residual wrench obtained from the new predicted motion using both solutions 1 and 2.

As observed in Figure 7.17, the residual wrench of both solutions is practically the same as the one obtained in the full tracking formulation (Figure 7.11). Vertical force, being the component that changes the most, varies between $-8,0$ and $8,5$ N. Thereby, obtaining the same $1,3\%$ of maximum error, e_{max}, with respect to the subject's weight.

To sum up, these formulations correspond to two alternatives to the full tracking formulation, in which the computational time is reduced and the *lumbar_extension* coordinate is more similar to the experimental one (as seen in Tables 7.5 and 7.8). Depending on the strategy that is wanted to follow, one might chose the first or the second formulation. For example, if torques are wanted to be very similar, the first formulation (the tracking of accelerations and torques) may be more

appropriate. On the other hand, if it is accepted that torques can vary, the second formulation (the full kinematic tracking with no torques) offers a very good solution in terms of computational time, with resulting coordinates very similar to the experimental ones. Below, in Figures 7.18 and 7.19, the solutions of both formulations are illustrated with the OpenSim skeletal model.

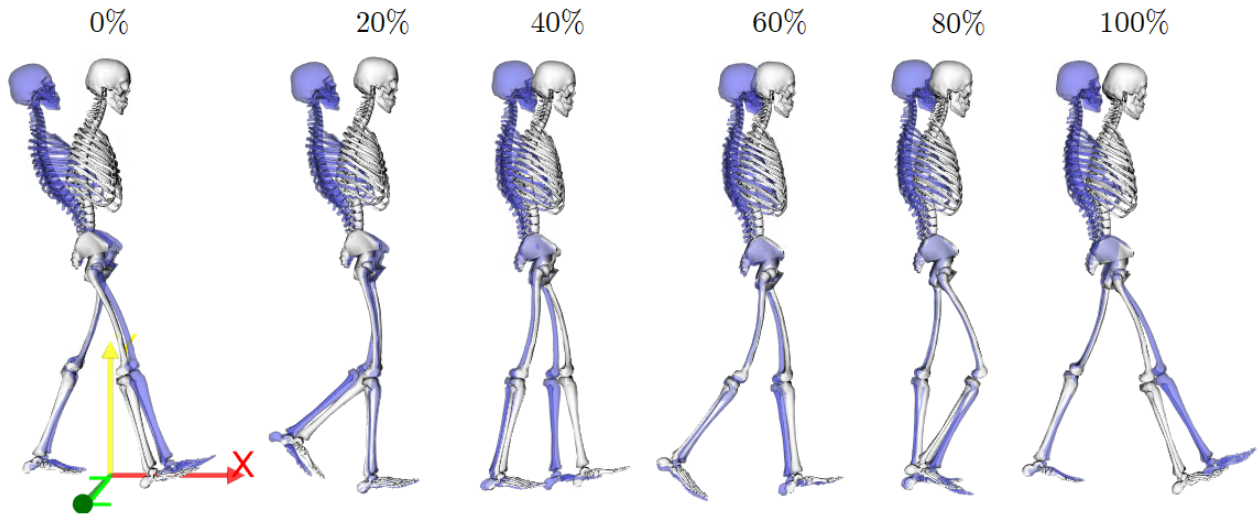


Figure 7.18 Representation of the solution found by the final formulation 1. The blue model represents the new motion, while the white one the experimental motion. Percentage of cycle is shown above the models.

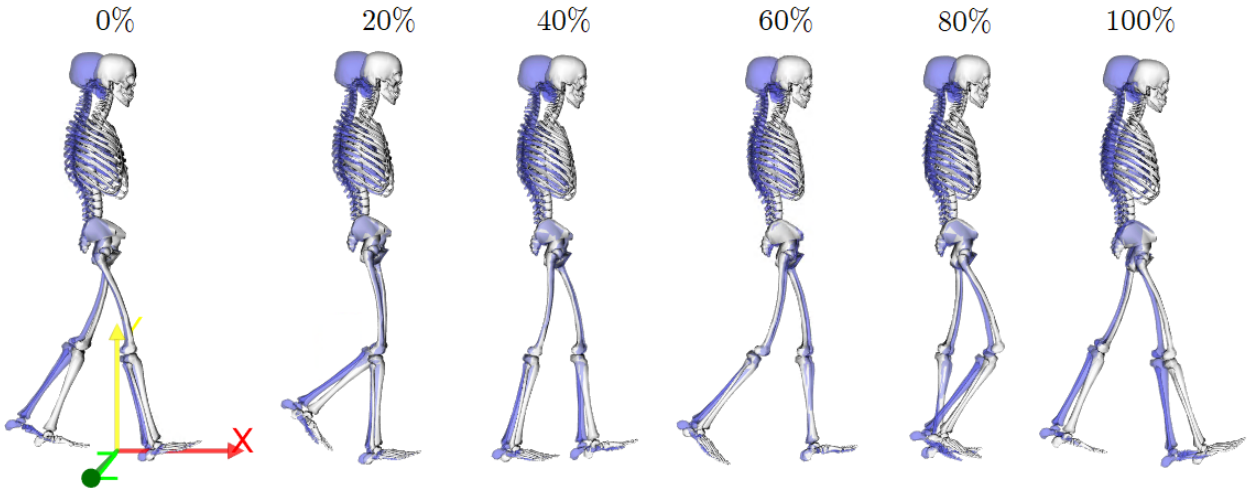


Figure 7.19 Representation of the solution found by the final formulation 2. The blue model represents the new motion, while the white one the experimental motion. Percentage of cycle is shown above the models.

8. PROJECT IMPACT & IMPLEMENTATION

In this chapter, environmental and social impacts that may cause this project are contemplated. Then, the work plan and the economic cost of the project are explained.

8.1. Environmental impact

The environmental impact of this project is minimal, since the work has been basically based on simulations using a computer. However, electricity, computers and the equipment needed to capture the motion can be considered.

First, it should be noted that electrical consumption has been low. Only one capture session was carried out, and no power equipment was employed. Also, a computer and a light bulb has assumed to be open while working in the thesis. Nevertheless, their consumption is marginal. Therefore, it is considered that the realization of this project has not been a significant increase of electrical consumption of the Biomechanics Laboratory of UPC.

During the motion capture, stick tape was used to attach reflective markers to the body. 22 markers were used, and more or less, 4 cm of tape for each marker. Thus, a total of 88 cm of stick tape were wasted. Furthermore, deterioration of the electrical and electronic equipment (lab equipment and computers) must be taken into account. Once their useful life is finished, they must be treated individually, in agreement with the Regulation 2017/699, and with the Directive 2012/19/EU of the European Parliament and of the Council of 04 July 2012 on waste electrical and electronic equipment (WEEE).

8.2. Social impact

This thesis represents an analysis of a method which can facilitate the acquisition of results through simulation. The incorporation of these kind of methods to biomechanics involves an important breakthrough to the motion prediction, allowing professionals to assess outcomes before they occur.

Motions can be studied and analyzed without the limitation of reproducing real movements. Thereby, different cases can be studied and the best solution can be chosen without executing a trial and error methodology. For example, predictions can be performed when a surgical intervention modi-

fies some biomechanical characteristic of a patient, and then, asses how she or he will walk. Also, motion results of patients wearing orthoses can be predicted, allowing to personalize and optimize the design of them. In this way, the adaptation process of patients is highly improved, people who may need them for rehabilitation or for facilitating their day-life.

In short, the progress in human motion prediction may cause an impact that could be quantified both in economic savings (preserving resources and time), as in the benefit of those patients who directly could take advantage of these methods to improve their quality of life.

8.3. Work plan

In the following page, the work plan of this project is illustrated with a Gantt diagram (Figure 8.1). The whole semester contains 19 weeks, and activities are mainly separated in writing, reading, learning and execution tasks.

As it can be seen, in the first 4 weeks, a general knowledge of biomechanical concepts, optimal control concepts, lab equipment and the capture procedure was acquired. Also, the motion capture and the subsequent acquisition of the experimental variables were carried out. Next, from week 5 to 7, optimal control techniques, GPOPS-II and a general knowledge of its mathematical background were learned. Furthermore, a first GPOPS-II code was studied. Week 8 corresponds to the midterm exams week and no activities were scheduled. Then, from week 9 to 12, while keeping learning GPOPS-II, the first chapters of the thesis were written. From week 12 to 17, a GPOPS-II code was built to extract results from the experimental data. Furthermore chapters of GPOPS-II and results were written. Finally, in the last 2 weeks, general improvements in the thesis writing were made in order to get ready to handed-in it.

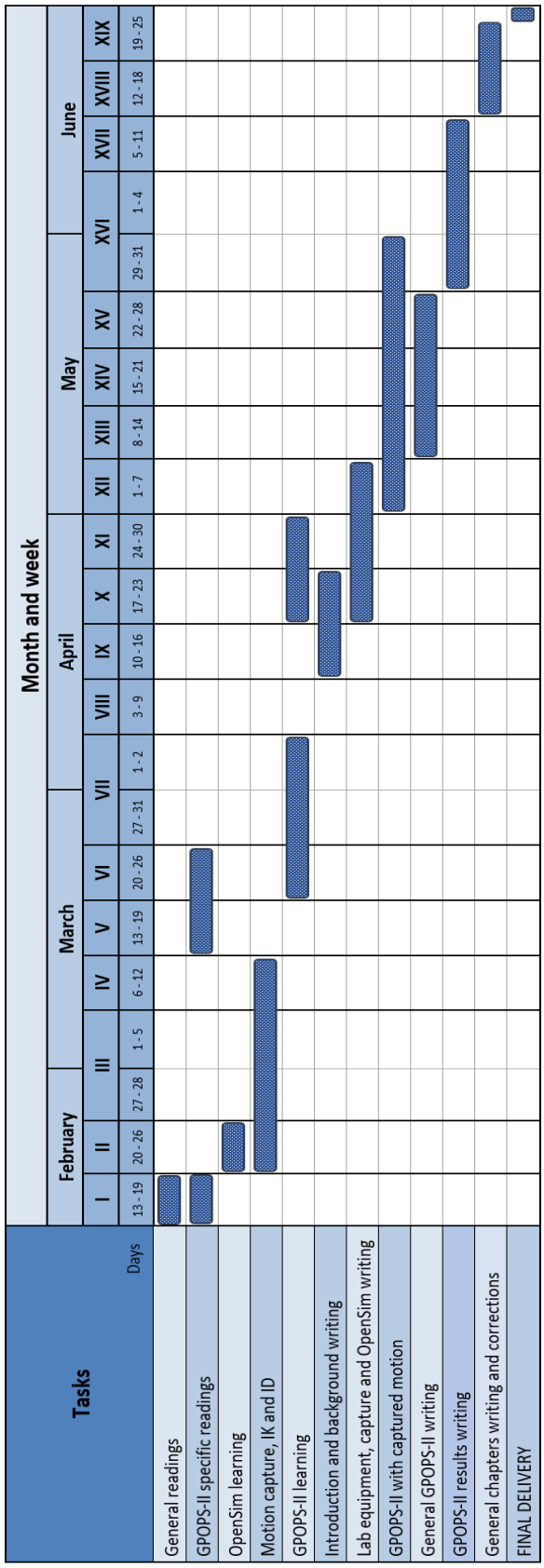


Figure 8.1 Gantt diagram of the project. Tasks are mainly split in writing, reading, learning and execution tasks. Note that each week of the semester is expressed with a roman number.

8.4. Economic cost of the project

The economic cost of the project consist on four different aspects: the depreciation of the laboratory equipment and computers, the cost of MATLAB and GPOPS-II licenses, the cost of working time of the student and supervisors, and finally, an estimation of the electrical energy consumed.

Depreciation can be calculated from the total price of the equipment and computers, the knowledge of their useful life and the total time that they have been used. It has been estimated a useful life of 8 years for the laboratory equipment and 5 years for the computers. Lab equipment is considered to be used 4 hours per day, 5 days per week and 48 weeks per year, which is a total of 7680 hours of useful life. On the other hand, computers are considered to be used 12 hours per day, 5 days per week and 48 weeks per day, which is a total of 14400 of useful life.

Only one capture session was done. Thus, the laboratory equipment was only working for a couple of hours. Moreover, during all the project two computers have been employed: a department computer and a personal laptop. Department computer has been used 14 hours per week, during 15 weeks. Hence, a total of 210 hours during the thesis. On the other hand, personal computer has been utilized a little bit less, it has been estimated a total of 200 hours.

MATLAB and GPOPS licenses expire after one year. Taking the whole year (24 hours per day, 7 days per week, 52 weeks per year), licenses last 8736 hours. MATLAB has been running for 210 hours, the same amount of time that the department computer has been open. Considering that GPOPS needs MATLAB to be used, it has been approximated that GPOPS has been working 100 hours of the total 210 hours.

As for the student's hours of dedication, it should be taking into account the amount of time that computers have been used (410 hours), plus 40 hours of meetings, calculations and reflections. Then, as ETSEIB recommends, the salary for an engineering student has been considered of 8 €/h. Also, 35 hours of support and supervision have been approximated.

Finally, the cost of electrical energy consumed has been estimated. Since the lab equipment was only used for two hours and no power equipment was employed, the consumption during the capture session has been neglected. Regarding to the student's hours of dedication, it has been supposed that a light bulb has been open all the 450 hours, and the computers have been consuming the 410 hours. Power of light and computers has been estimated to 40 W for each of them. The price of electricity has been assumed constant, with a value of 0,14 €/kWh.

In table 8.1 the cost of the project is presented. Each cost related to the factors described above is shown in detail. The total cost of the project is 5407,47 €.

Cost factor	Fixed cost [€]	Life expectancy [years]	Variable cost [€/h]	Time referred to project [h]	Cost related to project [€]
OptiTrack capture system and software Motive	17000	8	2,2135	2	4,43
Force plates AMTI Accugait (2 units)	18000	8	2,3438	2	4,69
Department computer	1000	5	0,0694	210	14,58
Personal laptop	750	5	0,0521	200	10,42
Matlab license	500	1	0,0572	210	12,02
GPOPS-II license	600	1	0,0687	100	6,87
Supervisors	-	-	50	35	1750,00
Student	-	-	8	450	3600,00
Electrical energy	-	-	0,0099	450	4,47
TOTAL COST					**5407,47 €**

Table 8.1 Calculation of the final project cost. Variable costs of equipment, computers and licenses are obtained from dividing their fixed cost by their life expectancy in hours. Variable cost of electrical energy is found from its price multiplied by the power consumption of computers and light.

CONCLUSIONS

In the present bachelor thesis, optimal control techniques have been applied to predict dynamically consistent walking motions. Concretely, a tracking strategy has been performed to obtain new motions, similar to the walking motion capture carried out at the Biomechanics Laboratory of UPC. Thus, there has been a process of getting acquainted with the Lab equipment and the related software.

The software OpenSim and a 2D skeletal model developed by the OpenSim team have been used. The captured motion has been projected to the sagittal plane. Then, inverse kinematic and dynamic analyses have been performed to obtain the experimental coordinates, velocities, accelerations and torques.

GPOPS-II, which works in MATLAB, has been employed as the tool to implement the optimal control algorithm. First, the influence of parameters on the solution has been analyzed to learn how the algorithm behaves and how the desired results can be obtained. Then, different optimal control formulations have been studied. Starting from a full tracking formulation, in which all the experimental data is tracked, modifications have been done to check if there is an optimum strategy.

Results have been compared and assessed in terms of computational time or convergence, similarity between the experimental and the resulting motion, and reduction of residual wrench. This comparison and assessment between different formulations have shown which strategies could be better to perform a prediction with tracking experimental data.

As a general conclusion, results obtained in this thesis are valued positive. Applying an optimal control technique, it has been possible to obtain new motions, similar to the experimental capture, in which the residual wrench has been highly minimized from the one obtained in the first inverse dynamic analysis. Furthermore, alternatives to the full tracking solution have been found satisfactory, and a knowledge in optimal control, GPOPS-II and different tracking strategies have been gained.

Throughout the work, a number of difficulties have been presented. From all of them, it can be remarked the learning of optimal control theory and its environment, as well as the implementation of a program that allows to call OpenSim libraries when GPOPS-II is running. Nevertheless, all of them have been solved with meetings, tests, team work, literature search and a lot of hours of dedication.

It has been checked that optimal control techniques correspond to a very powerful strategy in biomechanics in order to predict new motions, while accomplishing a set of desired restrictions. However, in this project, the analysis has been a first test of the tool. Consequently, different proposals that will be carried out in the future are considered.

In the first place, it is proposed to extend the knowledge gained in this project to a 3D skeletal model. Thus, assessing convergence and similarity between resulting coordinates and experimental motion, to adjust the optimal control formulation to a problem of higher complexity. Secondly, in order to adapt the tracking strategy to incomplete SCI patients, it is suggested to introduce orthoses and crutches to the model. In this way, starting to analyze unhealthy motions and adding external forces from the orthoses and crutches to the formulation. Finally, as a future achievement, it is contemplated to develop a foot-ground contact model, so as to predict new motions without the need to depend on measured foot-ground forces. Thus, being able to generate the contact forces in predictive formulations, in which an "optimal gait" is aimed to be obtained for a specific subject.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisors, Josep Maria Font Llagunes and Míriam Febrer Nafría, for their support, advise and strong dedication during the project. Their guiding and involvement in this project have been fundamental to consolidate the thesis. Furthermore, I really appreciate how they treated me from the first moment, making me feel not only a student, but also a member of the research group.

Moreover, I would like to thank to the BIOMEC group, for giving me a place to work and letting me use the laboratory equipment. I give thanks to the lab meetings with them, and any advice and help that they could have given me to make my thesis improve.

REFERENCES

[1] Ackermann, M., and Schiehlen, W. Dynamic Analysis of Human Gait Disorder and Metabolical Cost Estimation. *Archive of Applied Mechanics 75* (2006), 569–594. DOI: 10.1007/s00419-006-0027-7.

[2] Ackermann, M., and van-den Bogert, A. J. Optimality principles for model-based prediction of human gait. *Journal of Biomechanics 43* (2010), 1055–1060. DOI: 10.1016/j.jbiomech.2009.12.012.

[3] Agulló-Batlle, J. *Introducció a la mecánica analítica, percussiva i vibratòria.* Publicacions OK punt, Barcelona, 1997. ISBN: 84-920850-3-7.

[4] Anderson, F. C., and Pandy, M. G. Dynamic Optimization of Human Walking. *Journal of Biomechanical Engineering 123(5)* (May 2001), 381–390. DOI: 10.1115/1.1392310.

[5] Betts, J. T. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2 ed. SIAM, The Boeing Company, Seattle, Washington, 2010. ISBN: 978-0-89871-688-7.

[6] Bosch-Soldevila, E. *Anàlisi dinàmica inversa del moviment humà mitjançant el programari Opensim.* Master's thesis, Universitat Politècnica de Catalunya (UPC), Spain, December 2013.

[7] Carnicero-Carmona, A. *Anàlisi cinemàtica i dinàmica del moviment dels braços en un llançament de bàsquet.* Bachelor's thesis, Universitat Politècnica de Catalunya (UPC), Spain, July 2014.

[8] Czamara, A., Markowska, I., and Hagner-Derengowska, M. Three-dimensional kinematic analysis of ankle, knee, hip, and pelvic rotation during gait in patients after anterior cruciate ligament reconstruction - early results. *BMC Musculoskeletal Disorders 16* (2015), 1–9. DOI: 10.1186/s12891-015-0726-8.

[9] De-Groote, F., Kinney, A. L., Rao, A. V., and Fregly, B. J. Evaluation of Direct Collocation Optimal Control Problem Formulations for Solving the Muscle Redundancy Problem. *Annals of Biomedical Engineering 44(10)* (October 2016), 2922–2936. DOI: 10.1007/s10439-016-1591-9.

[10] de Jalon, J. G., and Bayo, E. *Kinematic and Dynamic Simulation of Multibody Systems*, 1 ed. Springer-Verlag, 1994. ISBN: 978-1-4612-7601-2.

[11] de Rosario, H., Page, A., Besa, A., Mata, V., and Conejero, E. Kinematic description of soft tissue artifacts: quantifying rigid versus deformation components and their relation with bone motion. *Medical & Biological Engineering & Computing 50* (2012), 1173–1181. DOI: 10.1007/s11517-012-0978-5.

[12] del-Ama, A. J., Koutsou, A. D., de-los-Reyes, A., Gil-Agudo, Á. and Pons, J. L. Review of hybrid exoskeletons to restore gait following spinal cord injury. *The Journal of Rehabilitation Research and Developments 49* (2012), 497–514. DOI: 10.1682/JRRD.2011.03.0043.

[13] Delp, D. Getting started with inverse dynamics. http://simtk-confluence.stanford.edu:8080/display/OpenSim/Getting+Started+with+Inverse+Dynamics. [Online; accessed: 5-May-2017].

[14] Delp, D. Getting started with inverse kinematics. http://simtk-confluence.stanford.edu:8080/display/OpenSim/Getting+Started+with+Inverse+Kinematics. [Online; accessed: 5-May-2017].

[15] Delp, D. Getting started with rra. http://simtk-confluence.stanford.edu:8080/display/OpenSim/Getting+Started+with+RRA. [Online; accessed: 15-June-2017].

[16] Delp, D. Getting started with scaling. http://simtk-confluence.stanford.edu:8080/display/OpenSim/Getting+Started+with+Scaling. [Online; accessed: 3-May-2017].

[17] Delp, D. How scaling works. http://simtk-confluence.stanford.edu:8080/display/OpenSim/How+Scaling+Works. [Online; accessed: 3-May-2017].

[18] Delp, D. Musculoskeletal models. http://simtk-confluence.stanford.edu:8080/display/OpenSim/Musculoskeletal+Models/. [Online; accessed: 22-Apr-2017].

[19] Delp S. L., Anderson F. C., Arnold A. S., Loan P., Habib A., John C. T., Guendelman E. and Thelen D. G. *OpenSim: Open-source Software to Create and Analyze Dynamic Simulations of Movement*. IEEE Transactions on Biomedical Engineering, 2007.

[20] Dumas, R., Cheze, L., and Verriest, J.-P. Adjustments to McConville et al. and Young et al. body segment inertial parameters. *Journal of Biomechanics 40* (2007), 543–553. DOI: 10.1016/j.jbiomech.2006.02.013.

[21] Febrer-Nafría, M. *Application of Optimal Control in the Simulation of Human Motion*. Master's thesis, Universitat Politècnica de Catalunya (UPC), Spain, March 2016.

[22] Garg, D., Patterson, M. A., Darby, C. L., Francolin, C., Huntington, G. T., Hagerk, W. W., and Rao, A. V. Direct trajectory optimization and costate estimation of infinite-horizon optimal control problems using collocation at the flipped legendre-gauss-radau

points. *IEEE/CAA Journal of Automatica Sinica 3* (2016), 174–183. DOI: 10.1109/JAS.2016.7451105.

[23] Hatze, H. The meaning of the term 'Biomechanics'. *Journal of Biomechanics 7* (1974), 189–190. DOI: 10.1016/0021-9290(74)90060-8.

[24] Huntington, G. T. *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems.* PhD thesis, Massachusetts Institute of Technology (MIT), United States of America, June 2007.

[25] Jones, O. Anatomical planes. http://teachmeanatomy.info/the-basics/anatomical-terminology/planes/. [Online; accessed: 15-Apr-2017].

[26] Kaplan, M. L., and Heegaard, J. H. Predictive algorithms for neuromuscular control of human locomotion. *Journal of Biomechanics 34* (2001), 1077–1083. DOI: 10.1016/S0021-9290(01)00057-4.

[27] Kelly, M. P. *Transcription Methods for Trajectory Optimization.* Cornell University, 2015.

[28] Meyer, A. J., Eskinazi, I., Jackson, J. N., Rao, A. V., Patten, C., and Fregly, B. J. Muscle Synergies Facilitate Computational Prediction of Subject-Specific Walking Motions. *Frontiers in Bioengineering and Biotechnology 4* (2016). DOI: 10.3389/fbioe.2016.00077.

[29] Mombaur, K. Optimal Control for Applications in Medical and Rehabilitation Technology: Challenges and Solutions. *Springer Optimization and Its Applications 109* (January 2016), 103–145. DOI: 10.1007/978-3-319-30785-5_5.

[30] Muñoz-Farré, A. *Gait analysis of paediatric patients with hemiparesis*. Bachelor's thesis, Universitat Politècnica de Catalunya (UPC), Spain, June 2016.

[31] Patterson, M. A., and Rao, A. V. GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming. *ACM Transactions on Mathematical Software (TOMS) 41* (2014). DOI: 10.1145/2558904.

[32] Patterson, M. A., and Rao, A. V. *GPOPS-II, A General-Purpose MATLAB Software for Solving Multiple-Phase Optimal Control Problems*, version 2.3 ed. University of Florida, December 2016.

[33] Pàmies-Vilà, R. *Application of Multibody Dynamics Techniques to the Analysis of Human Gait.* PhD thesis, Universitat Politècnica de Catalunya (UPC), Spain, December 2012.

[34] Pàmies-Vilà, R., Font-Llagunes, J. M., Lugrís, U., Alonso, F. J., and Cuadrado, J. A Computational Benchmark for 2D Gait Analysis Problems. *New Trends in Mechanism and Machine Science 24* (2015), 689–697. DOI: 10.1007/978-3-319-09411-3_73.

[35] Pätkau, O. *Application of different Control Strategies to the FD Simulation of Human Gait*. Master's thesis, Universitat Politècnica de Catalunya (UPC), Spain, June 2014.

[36] Rao, A. V., Patterson, M. A., and Hager, W. A *ph* Collocation Scheme for Optimal Control. *Automatica* (January 2013). Submitted for Publication.

[37] Serrancolí, G., Schutter, J. D., and De-Groote, F. Analysis of Optimal Control Problem Formulations in Skeletal Movement Predictions. *Converging Clinical and Engineering Research on Neurorehabilitation II. Biosystems & Biorobotics 15* (2017), 1269–1273. DOI: 10.1007/978-3-319-46669-9_207.

[38] Spangelo, S. GPOPS-II, Guest Aero 575 Lecture.

[39] Vaughan, C. L., Davis, B. L., and O'Connor, J. C. *Dynamics of Human Gait*, 2 ed. Kiboho Publishers, South Africa, 1992. ISBN: 0-620-23558-6.

[40] Wächter, A. *Short Tutorial: Getting Started With Ipopt in 90 Minutes*. IBM T.J. Watson Research Center.

[41] Xiang, Y., Arora, J. S., Rahmatalla, S., and Abdel-Malek, K. Optimization-based Dynamic Human Walking Prediction: One Step Formulation. *Numerical Methods in Engineering 79* (2009), 667–695. DOI: 10.1002/nme.2575.