Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials (GETI)

Predicting Human Motion Assisted by Wearable Hybrid Devices That Combine Robotics and Neuroprostheses

MEMÒRIA

September 13, 2021

Autor: Berta Pons Vilà

Director: Albert Peiret Gimenez

Josep Maria Font Llagunes

Convocatòria: Setembre 2021

ETSEIB

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona


UPC


ETSEIB

Abstract

Spinal cord injury is a neurological disorder that affects millions of people worldwide that can cause motor disabilities. It seriously affects the patients and the people around them.

Advancements in robotics have developed assistive devices such as wearable exoskeletons for rehabilitation and assisted motion. Moreover, research on functional electrical stimulation (FES), makes it possible to use neuroprostheses to induce muscle contraction using electrical stimulation.

The objective of this bachelor thesis is to use optimal control to design hybrid strategies for assisted walking combining wearable exoskeletons and neuroprostheses for patients with motor disabilities. Optimal control techniques are used to find and optimize assisted walking strategies.

The first part of this thesis describes the theoretical background of human movement biomechanics and analysis, spinal cord injury, and optimal control techniques and applications. It explains in detail the methods and resources used. Afterward, using the software OpenSim and Matlab, optimal control problems are designed for assisted walking motion. Lastly, the results of the different assistive walking cases are presented and discussed.

The results obtained show various combinations of assistive walking. Both exoskeleton and neuroprosthesis options are analyzed individually and then, different combinations of hybrid strategies are analyzed. The results present how using more exoskeleton and less neuroprosthesis and vice versa affects assisted walking. In the long run, further research on the field can lead to significant improvements in the lives of patients.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

There are many people with motor disabilities caused by a damaged neural system, such as spinal cord injury. That can seriously affect their lives and the people around them. Assistive devices are designed for rehabilitation to recover partial or total mobility of patients. Wearable robotics like exoskeletons can help patients perform their daily activities. Moreover, recent studies on functional electrical stimulation (FES), have developed neuroprostheses that can induce muscle contraction and generate movement.

The motivation of this project is to explore hybrid strategies of assisted walking to find improved solutions for patients with motor disabilities. Combining wearable robotics and neuroprostheses can have a great improvement on assistive devices and patients' lives.

1.2 Objectives

The main objective of this thesis is to design hybrid strategies for assisted walking combining wearable robotics and neuroprostheses using optimal control techniques. More specifically:

- Understanding biomechanics of human motion, different motion analyses, and optimal control problems.
- Analyzing the walking motion from experimental data from the Biomechanics Laboratory of the UPC.
- Develop Matlab code using OpenSim Moco to design optimal control problems with a patient-device model. different assisted walking strategies including wearable exoskeletons and neuroprostheses.

1.3 Project Scope

The scope of this project includes the use of optimal control motion prediction techniques to design hybrid strategies for assisted walking. However, taking into account the timing and resources of the project, there are some limitations to be considered.

Due to the current circumstances, it has not been possible to obtain the experimental data during this project, so it is used previously captured data. The musculoskeletal model does not include a contact model for the foot-ground contact forces. Regarding the assisted devices, it is only considered the ability to generate a motor torque for the exoskeleton, and muscle activation for the neuroprosthesis. The design and control of these devices are out beyond the scope of this project.

Finally, due to timing and complexity, it is out of scope of the project the energetic analysis of the different strategies studied. Nonetheless, potential and possible further steps of the project are presented.

2 Theoretical Background

2.1 Biomechanics of human motion

The human body is a very complex machine. It has around 200 bones and 600 muscles. Biomechanics is the application of mechanics to study the functions and motions of biological systems [10]. To use biomechanics to study the different motions of the human body, the human body is simplified in biomechanical models. A biomechanical model is the description of a biological system as a mechanical device. The different elements of the biological system are defined as rigid bodies, joints, and actuators that comply with the laws of physics [27].

Various human models have been created to help understand the dynamics of human motion. Based on the complexity, each model has its advantages and limitations. For instance, a simpler model may be computationally cheaper, but it will have certain restrictions with simulation capabilities. Meanwhile, a more complex model may require the implementation of machine learning tools such as deep reinforcement learning but it will be more accurate and perform better [15].

When considering motions where the lower body generates most of the movement, such as walking, usually the main focus of the study is the lower part of the human body, while the upper body can be simplified as a single solid called HAT (head, arms, trunk). This is illustrated in the OpenSim model "gait2354", where there are only lower body muscles (Figure 1):

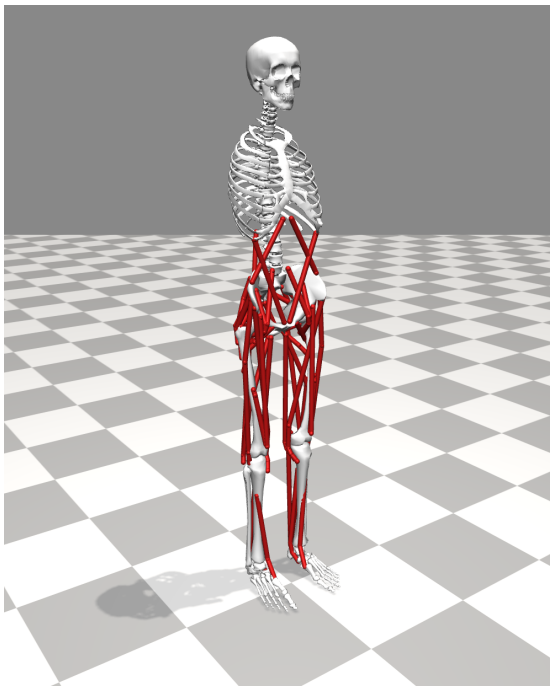

Figure 1: OpenSim model "gait2354"

In other words, the biomechanical model used varies depending on the objectives and resources of the study.

2.2 Gait cycle

One of the motions studied in biomechanics is the human gait. The normal gait movement consists of a harmonious movement between the upper and lower limb extremities [24]. Amongst various models that describe the gait motion, there is the Perry Model. The gait cycle (or stride) is the interval between two consecutive initial floor contacts by the same limb. According to the Perry Model, it consists of two periods: the stance and the swing, divided in 8 phases shown in Figure 2. The stance is formed by the first 5 first phases and the swing by the 3 remaining ones.

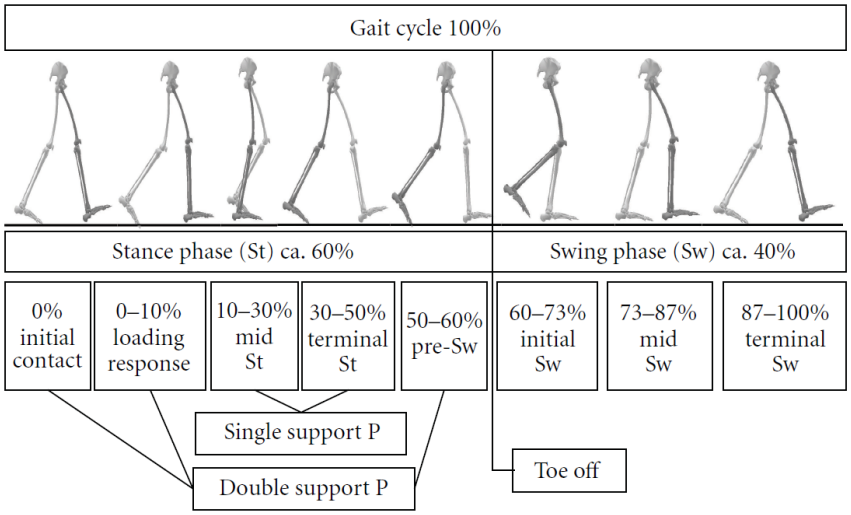

Figure 2: Normal Gait Cycle [24][9]

The Perry Model is symmetric, meaning that the right gait cycle is identical to the left one, and its 8 phases of the gait cycle are:

1. Initial contact: When the foot just touches the floor
2. Loading response: The initial double stance period from the initial floor contact until the other foot is lifted for the swing period.
3. Mid Stance: The first half of the single-limb support interval. It starts as the other foot is lifted and it ends when the bodyweight is aligned over the forefoot.
4. Terminal Stance: The other half of the single-limb support begins with heel rise until the other foot strikes the ground.
5. Pre-Swing: The double stance interval and also the final phase of the stance period. It starts with the initial contact of the opposite limb and ends with an ipsilateral toe-off.
6. Initial Swing: It begins with the lift of the foot from the floor and ends when the swinging foot is opposite the stance foot.
7. Mid swing: It begins when the swinging leg is opposite the stance limb and ends when the swinging limb is forward and the tibia is vertical.
8. Terminal Swing: It begins with a vertical tibia and ends when the foot strikes the floor.

2.3 Analysis of human motion

The generation of movement can be synthesized in the following loop diagram:

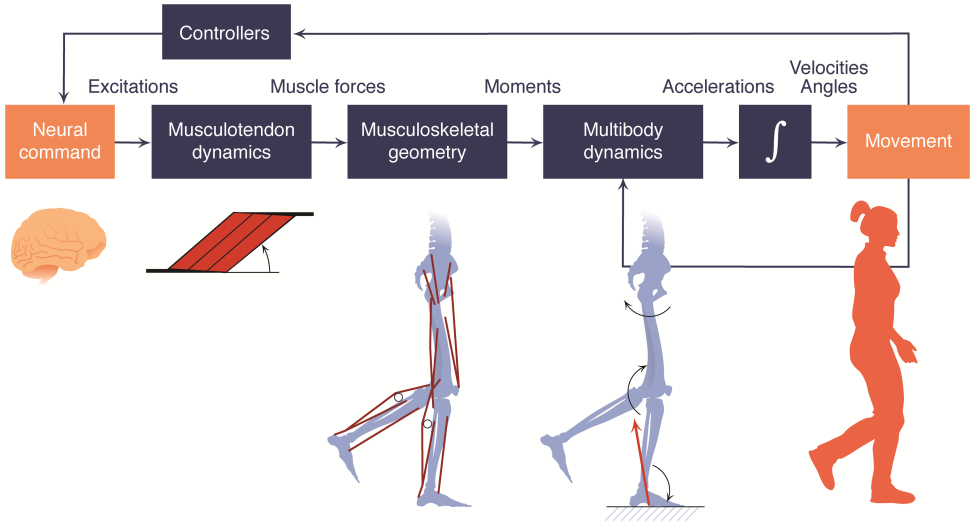

Figure 3: Schematic representation of the injured spinal cord [28]

The brain sends a neural command that activates the muscles, performing certain forces and torques that cause movement, defined by trajectories, velocities, angles, etc. Afterwards, a sensory feedback is sent to the brain.

In order to study the kinematics and dynamics of human motion, there are different methods developed to approach it. Broadly, they are two main types of analyses, inverse and forward methods. With the inverse methods, the starting point is the movement, which can be the trajectories of the joints, and from that the forces and movements are predicted (e.g., inverse kinematics and dynamics). Meanwhile, the forward methods approach the problem from forces and torques and from that, they predict movement (e.g., forward dynamics).

A particular analysis is the direct collocation method, that combines inverse and forward dynamics analyses and quantifies tasks with an objective function. It is an implicit method where the state and control variables are discretized. Dynamics equations are converted into algebraic constraints which makes functions continuous and differentiable so there is no need for explicit integration and therefore, the optimization converges quicker and more reliably [36].

2.4 Optimal Control Problem

It is a common assumption that natural motions of the human body are optimal, a product of evolution. As mentioned earlier, the human body is a very complex machine, so it could be considered as a control system where dynamic motion tasks such as walking can be formulated as optimal control problems. This gives the possibility to use optimality to predict human movement and improve performance with motion generation and analysis in rehabilitation [12].

As its name implies, an optimal control problem's main goal is to optimize a certain movement. This is achieved by finding the variables that minimize a determined cost function. Optimal control problems start with a biomechanical model of the human body, formed by bodies and the

links between them. The model has state variables such as generalized coordinates and velocities and control variables such as muscle activations. These variables are subjected to differential equations that describe the dynamics of the musculoskeletal system, and other interactions of the variables from the elements considered in the study (such as the interaction between feet and the ground). Additionally, they have to satisfy the path constraints and boundary conditions of the problem.

Finally, there is the objective function or cost function of the optimality problem to minimize, a formula that quantifies the cost of the motion. Depending on the objective of the problem, this function can consider different aspects, such as the tracking error, the metabolic cost or the muscle fatigue.

To summarize, an optimal control problem has the objective to find the states and controls that satisfy dynamics, path constraints and boundary conditions equations that minimize a certain cost function (see Table 1):

Optimal Control Problem			
Variables	Equations	Objective function	Goal
States: $x(t)$ Controls: $u(t)$	System dynamics: $\dot{x}(t) = f(x(t), u(t), t)$ Path constraints: $g(x(t), u(t)) \geq 0$ Boundary conditions: $c(x_o, x_1) = 0$	$J = \int_{t_o}^{t_1} f(x(t), u(t))dt$	$\min_{x,u} J(x, u)$

Table 1: Optimal Control Problem

2.5 Spinal Cord Injury

Spinal cord injury (SCI) is a neurological disorder that disrupts the normal functioning of the spinal cord such as motor deficit, sensory changes, and autonomic nervous system dysfunction [25]. It is a severe disease that has very serious effects on the patients and the people that surround them. The causes of SCIs can be traumatic or non-traumatic, the first one being the most common (90% of all causes of SCIs [2]). Around the world, there are between 250 000 and 500 000 cases per year [35]. Trauma is mainly a result of motor accidents, falls, violence, work-related injuries, sports-related injuries, or suicide attempts. Most patients are young males injured in road traffic collisions or elderly populations that have fallen [31].

The spinal cord is nervous tissue that provides communication between the brain and 31 spinal nerves associated with different parts of the body (shown in Figure 4). Additionally, it also produces reflexes called the spinal reflexes [16].

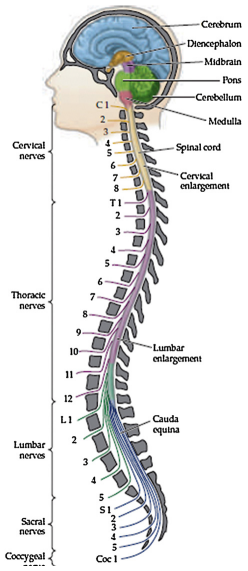

Figure 4: Set of the 31 spinal nerves [16]

As shown in the Figure 4, each spinal nerve is named with a letter and a number (i. e. C1).

Trauma on the spinal cord begins with the "primary injury", which can be caused by mechanical forces such as compression, or infarcted by a vascular injury [14]. Subsequently, neurological damage in the spinal cord starts (Figure 5 shows a schematic representation of the injured spinal cord).

Afterward, the physical injury triggers a series of biological events called "secondary injury", which can happen immediately or days later [16]. Secondary injuries may include systemic respiratory, cardiovascular, and immunological consequences of the primary injury [25]. Furthermore, there may be a "chronic phase" leading to neurological impairments [32]. As each spinal nerve corresponds to different nerves of the body, there is a relation between the spinal nerves damaged and the dysfunction caused. For example, injuries on the high cervical nerves are the most severe, causing paralysis in arms, hands, trunk, and legs (tetraplegia when all four limbs are affected). It can also cause breathing problems and trouble to speak. Whereas injuries on the sacral nerves (S1-S5) usually cause some loss of function in the hips and legs, but the patient is more likely able to walk. Most SCIs include loss of function in the hips and legs [29].

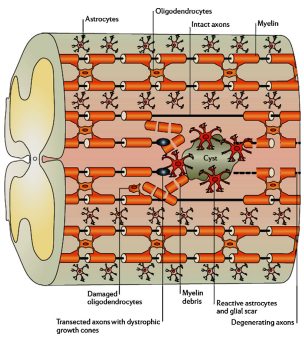

Figure 5: Schematic representation of the injured spinal cord [31]

The different levels of SCI damage are classified alphabetically by ASIA (American Spinal Injury Association). They are the following [25]:

- A: Complete (There is no sensory nor motor function)
- B: Sensory Incomplete (Sensory, but not motor)
- C and D: Motor Incomplete
- E: Normal (Normal sensory and motor functions with prior deficits)

Treatment and recovery/rehabilitation To this day, SCI still is a frequent and traumatic event without a cure. Nonetheless, a great deal of research has been made to understand SCI and develop recovery treatments. Rehabilitative, cellular, and molecular therapies have been tested in animal models, but none has proved to treat SCI successfully [32]. Being a very complex injury, several therapies are combined to treat the various conditions of SCI. Neuroprotection aims at protecting the functional neurons, neuroregeneration works towards regulating the lesion to regenerate cells and neurons to restore functional connections, and neurorehabilitation is based on physical training to recover function. Exercise has been demonstrated to be beneficial at cellular and biochemical levels [8].

Rehabilitation is fundamental to minimize and prevent complications, enhance function, and help patients deal with their new disability. Physical therapy focuses on strength building and optimizing mobility, respiratory and cardiovascular training, and muscle stretching [25]. Rehabilitation potential depends on the level of injury. There are many adaptive and assistive devices developing and improving, such as wheelchairs and exoskeletons, to allow the patients to live as independently as possible [25].

2.6 Assistive devices

There is an estimated amount of 65 million people worldwide who suffer from motor disabilities. Consequently, advancements in robotics have developed assistive devices with the ability to, for instance, cure spinal cord injuries. When implementing assistive devices to rehabilitation techniques, it is fundamental to model the physical properties of the patient and define the optimal control problem parameters that correspond with the patient [15].

Wearable exoskeletons are robotic devices attached to the human body that combine different technologies to help execute and improve determined motions. Their performance depends on many factors, such as the structure, the actuators, or the measurement devices [13].

There is a wearable exoskeleton currently being developed called "ABLE". It is an active orthosis to assist SCI patients when standing up and walking (See Figure 6).

Figure 6: ABLE weareble exoskeleton [1]

Another technique used for assisted motion is functional electrical stimulation (FES) and it is used for restoring function in people with motor disabilities. It consists of inducing muscle contraction with electrical stimulation to generate controlled movement. The nerves are stimulated through surface electrodes applied to the skin. Its functioning is modeled as a hysteresis curve with a maximum neural excitation value, which is around 40% of the maximum muscle activation.

3 Methodology

3.1 OpenSim

3.2 Biomechanical Model

The model used in this thesis is the OpenSim model *Gait10dof18musc* created by Ajay Seth, Darryl Thelen, Frank C. Anderson, and Scott L. Delp. This model is oriented towards lower extremity studies. It is composed of a trunk, pelvis, and leg segments. In total it has 10 degrees of freedom (DOF) and 18 muscles, as its name implies. The model is defined in *XML* language in a *.osim* file [19]. The OpenSim model represents the neuromuscular and musculoskeletal dynamics of the human body, each component corresponding to the different parts of the body, divided into the following categories: reference frames, bodies, joints, constraints, forces, contact geometry, markers, and controllers.

3.2.1 Bodies

Each body has a name, mass properties, and visible objects associated with it. The model is formed by 12 rigid bodies and the ground (See Table 2 and Figure 7).

Bodies	
Body part	OpenSim name
HAT (head, arms and trunk)	*torso*
Pelvis	*pelvis*
Right femur	*femur_r*
Right tibia	*tibia_r*
Right talus	*talus_r*
Right calcaneus	*calcn_r*
Right toes	*toes_r*
Left femur	*femur_l*
Left tibia	*tibia_l*
Left talus	*talus_l*
Left calcaneus	*calcn_l*
Left toes	*toes_l*
Ground	*ground*

Table 2: Model bodies

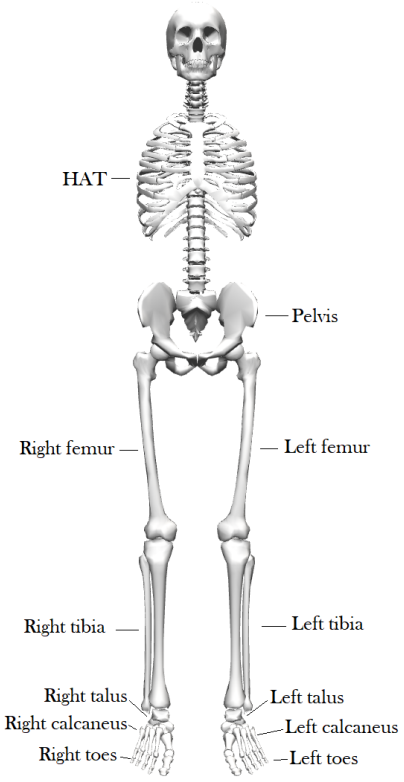

Figure 7: Gait10dof18musc bodies

3.2.2 Joints

Joints define the kinematic relation between two bodies, the parent and the child. The purpose of this nomenclature is to set one of the bodies as the reference when defining the relative motion between the bodies. The parent body is the reference body while the child body moves with respect to it. The kinematic constraints describe the freedom of movement allowed between the two bodies. There are defined types of joints available, but they can also be customizable. Table 3 describes the different joints of the model with their respective DOFs and corresponding generalized coordinates. This model is holonomic since the number of DOFs is the same as the number of generalized coordinates.

Joint	**OpenSim Joint Name**	**Child Frame**	**Parent Frame**	**DOF**	**OpenSim Coordinates**
Ground - Pelvis	*ground_pelvis*	Pelvis	Ground	3	*pelvis_tx* *pelvis_ty* *pelvis_tilt*
Lumbar Joint	*back*	HAT	Pelvis	1	*lumbar_extension*
Right Hip	*hip_r*	Right Femur	Pelvis	1	*hip_flexion_r*
Right Knee	*knee_r*	Right Tibia	Right Femur	1	*knee_angle_r*
Right Ankle	*ankle_r*	Right Talus	Right Tibia	1	*ankle_angle_r*
Right Subtalar	*subtalar_r*	Right Calcn	Right Talus	0	-
Right Metatarsal	*mtp_r*	Right Toes	Right Calcn	0	-
Left Hip	*hip_l*	Left Femur	Pelvis	1	*hip_flexion_l*
Left Knee	*knee_l*	Left Tibia	Left Femur	1	*knee_angle_l*
Left Ankle	*ankle_l*	Left Talus	Left Tibia	1	*ankle_angle_l*
Left Subtalar	*subtalar_l*	Left Calcn	Left Talus	0	-
Left Metatarsal	*mtp_l*	Left Toes	Left Calcn	0	-

Table 3: Model Joints

3.2.3 Forces

The forces applied to the model are also defined in the model file. There are two types of forces: passive forces like springs, dampers, and contact, and active forces like actuators and muscles. OpenSim allows ideal actuators that apply pure forces or torques directly proportional to the input control via is optimal force.

There are multiple models of muscle forces and they usually include muscle activation and contraction dynamics. The gait10dof18musc uses the Hill-type muscle model as Millard2012Equilibrium Muscle made by Dr. Matthew Millard, Tom Uchida, and Ajay Seth (cita1, cita2). It is a configurable equilibrium muscle model, which means that the forces generated by the fiber and tendon are equal. However, to solve optimal control problems with OpenSim Moco (section), the muscle model is switched to the DeGrooteFregly2016 Muscle by DeGroote et al. [21][7].

The model also includes the acceleration due to gravity so the gravitational forces are also taken into account.

3.3 Experimental Data

Due to various circumstances, it has not been possible to collect experimental data in this project. Nonetheless, this section presents the methodology to obtain the used experimental data.

The data used was obtained at the Biomechanics Laboratory of the Polytechnic University of Catalonia (a more detailed explanation of the whole process can be found in [23]). An OptiTrak™ motion capture equipment from NaturalPoint Inc was used to capture marker positions in a 3D space. It uses cameras (V100:R2 model) that emit infrared light, which is reflected on the markers, and captured by the cameras. The markers consist of small spheres covered by reflective material and positioned on strategic points of the subject. The capture frequency is 100Hz. Two captures are obtained with this equipment: a static capture of the subject in a static position and a motion capture of the subject performing a gait cycle. The cameras are strategically located around the laboratory to guarantee a precise capture.

Additionally, the laboratory includes two AMTI Accugait force plates, placed at the center of the laboratory. They provide the ground reaction forces by measuring the contact wrench applied to the body in contact. They obtain the ground reaction forces during the gait cycle that are used in the optimization control problem.

3.4 Scaling

When working with experimental data of motion capture, the model dimensions have to be as similar as possible as the subject ones. This process is called scaling and it consists of matching the distances of body markers locations and other parameters such as mass between the model and the real subject. This process can be done with the OpenSim 4.1 software using the Scale Model tool [20]. Figure 8 shows a diagram of the files involved in the process.

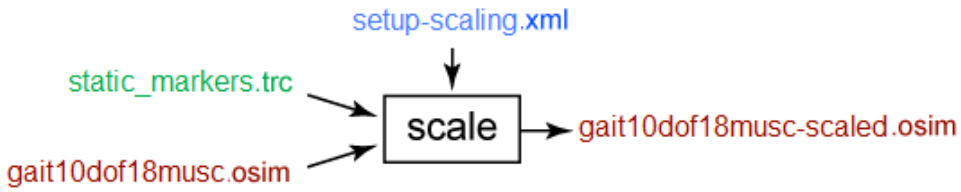

Figure 8: Files involved in the Scaling tool in OpenSim. (Adapted from [20])

The model (*gait10dof18musc.osim*) is loaded on the OpenSim graphic user interface (GUI) by selecting *File* > *Open Model* (See Figure 9).

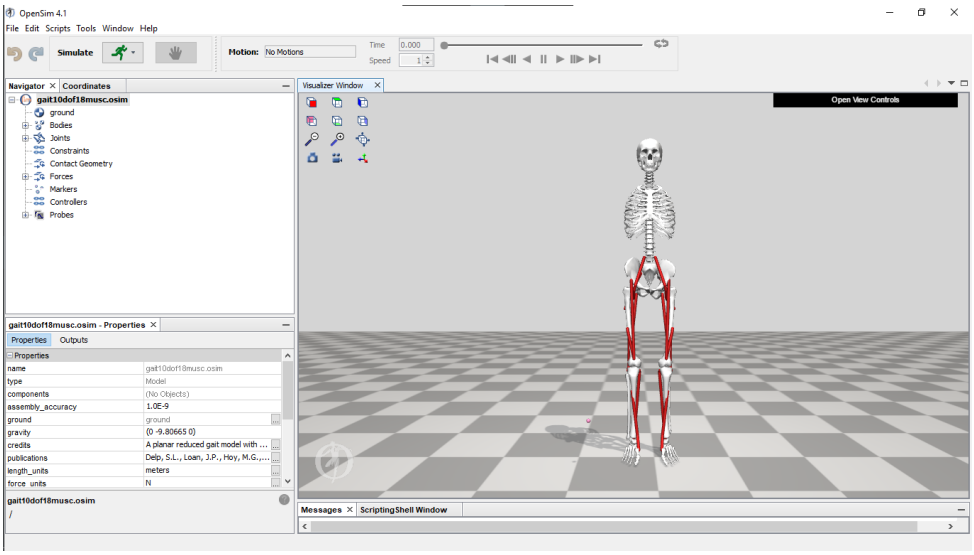

Figure 9: Loaded model in OpenSim GUI

Then, the Scale Model window is opened selecting *Tools* > *Scale Model*. It is where the input files and scaling settings are defined.

The experimental markers are the ones on the subject's body and their locations are obtained with a static capture using motion capture equipment. The static capture usually is a few captures of the subject in a static position and it is stored in a *.trc* file (*static-markers.trc*).

Selecting the *Load button*, the setup file (*setup-scaling.xml*) is added. This file contains the execution, model, and subject parameters, and scaling and marker placement properties. Among them are the following:

- The subject mass is specified in the *mass* property. To maintain the distribution of mass within the different body segments the *preserveMassDistribution* property is set to *True*.
- In the *MarkerPlacer* tag are the marker placement properties. To find the model configuration that matches the static pose of the subject, an inverse kinematics problem is presented. In the tag, the marker and coordinates weights used to compute the IK problem are specified (they can be edited and visualized in the Static Pose Weights tab).
- The name of the marker file is in the *markerFile* tag, and the time range of the static capture is in the *timeRange* tag.

Afterward, a file with the locations of the virtual markers, *markerSet.xml*, is added with the add markers from file button. These markers must be in anatomical correspondence with the experimental ones, as seen in Figure 10.

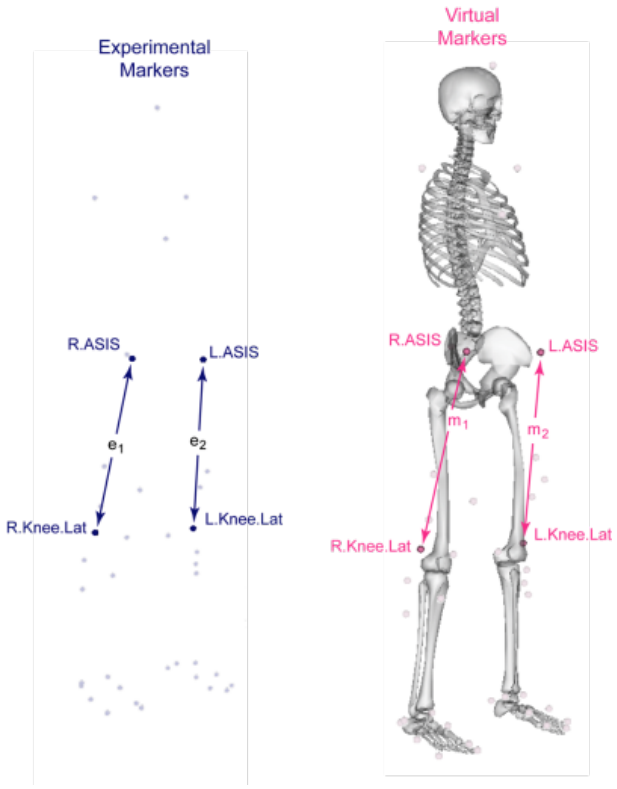

Figure 10: Correspondance between experimental and virtual markers [20]

Comparing the distances between the experimental and the virtual markers, scale factors are computed by dividing the experimental and virtual distances. These factors can also be added manually and visualized in the *Scale factors* tab. With these scale factors, the *Scale Model* tool algorithm scales the joint frame locations, mass center location, force application points, and muscle attachment points.

Lastly, before running the scaling, the *Preview Static Pose* button is checked to be able to visualize the results of the scaling and to make sure that it has been done accurately. Figure 11 shows the original and the scaled model. .

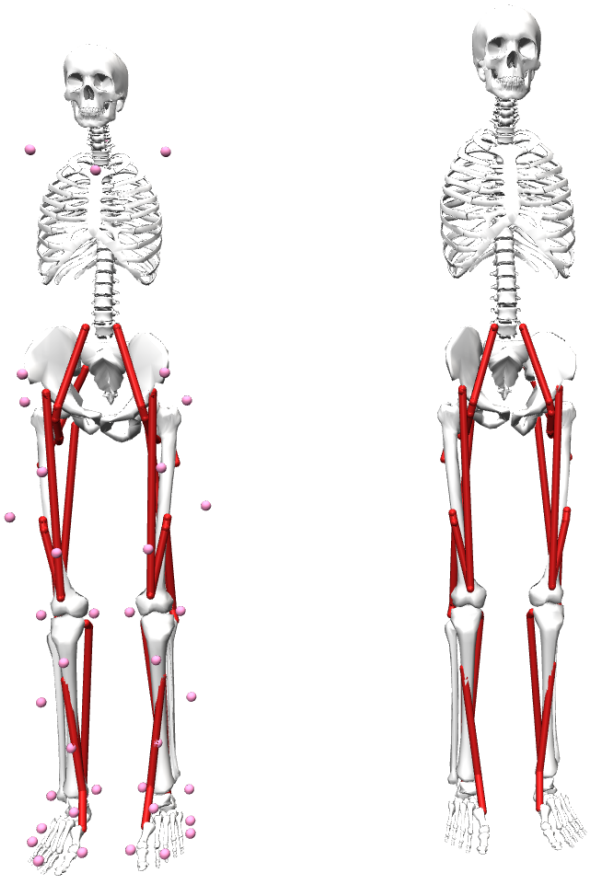

Figure 11: Original (right) and scaled (left) model

3.5 OpenSim Moco

Moco is a software toolkit, part of OpenSim, focused on solving optimal control problems in an easy and customizable way. The core library is written in C++, and it has Matlab, Python, and XML interfaces. It can be downloaded for Windows or Mac ([30]). In this thesis, Moco is used in Matlab [22].

Using the direct collocation method, it can solve optimal control problems, also the ones with kinematic constraints, for OpenSim models. This includes motion tracking, motion prediction, and parameter optimization problems. A diagram of how Moco works is shown in Figure 12.

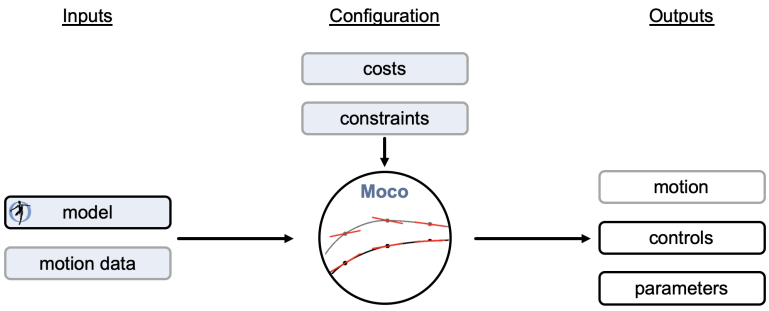

Figure 12: OpenSim Moco general structure [22]

Moco uses a library of cost and constraint modules, implemented with software classes that describe the problem and how it must be solved. The *MocoStudy* class is the main one that encompasses the optimal control problem (See Figure 13). It's divided into two other classes, *MocoProblem* and *MocoSolver*, and they don't depend on each other.

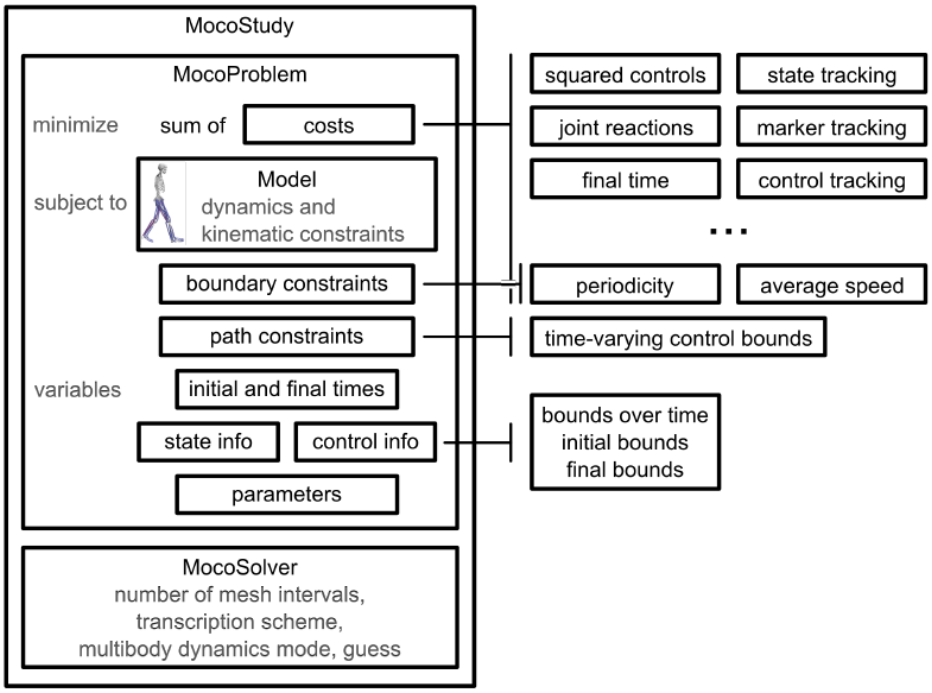

Figure 13: *MocoStudy* elements diagram [22]

The *MocoProblem* class contains all the elements that describe an optimal control problem: Cost terms, multibody and muscle dynamics, kinematic and boundary constraints, model properties, and variable bounds. Table 10 presents a generalized optimization problem that Moco can solve [6]:

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

Table 4: Moco optimal problem (extracted from [6])

The main cost function of the problem (J) is the sum of weighed (w_j) cost functions (J_j). The weight terms indicate how important is to minimize its cost function. These cost functions depend on the initial and final time bounds (t_0, t_f), states (y_0, y_f), controls (x_0, x_f), Lagrange multipliers (λ_0, λ_f), the time-invariant parameters, and integrals ($S_{b,k}$). All the variables have lower (L) and upper (U) bounds, and there are also bounds on the initial and final states and controls.

As stated in previous sections, the goal of the problem is to find the states ($y(t)$) and controls ($x(t)$) that minimize the cost function (J) while being subject to the multibody dynamics (which involve M, the mass matrix), the gravitational, muscle and other forces applied ($f_{applied}$), inertial forces ($f_{inertial}$), and auxiliary dynamics (that can be expressed as explicit

$f_{\dot{z},ex}$ or implicit $f_{\dot{z},im}$ differential equations). As mentioned above, the system might include kinematic constraints caused by forces applied by parts of the modeled system. To solve for these constraint forces, time-varying Lagrange multiplier variables (λ) are used. The derivative of the kinematic constraints (ϕ) introduces the kinematic constraint Jacobian (G). The system also includes boundary (V_k) and path (g) constraints with their corresponding lower ($V_{L,k}$ and g_L) and upper ($V_{U,k}$ and g_U) bounds.

The *MocoSolver* class is where the numerical method to solve the problem is detailed. It uses the *CasADI* library [3] to transcribe the continuous problem to a finite-dimensional nonlinear program with gradient-based nonlinear programs such as *IPOPT* [34] and *SNOPT* [26].

Moco uses the direct collocation method to solve optimal control problems. It is a nonlinear optimization technique to solve optimal control problems that approximates state and control variables using polynomials to simplify the integration calculations. Moco implements two transcription schemes: the trapezoidal scheme, which implements trapezoidal integration, and the Hermite-Simpson scheme, which uses parabolic integration and Hermite interpolation [11].

When solving a *MocoStudy*, a *MocoSolution* is generated. This *MocoSolution* is used as the initial guess of the *MocoTrajectory* in the next iteration (Shown in Figure 14). This breaks down a complex study into a series of simpler ones.

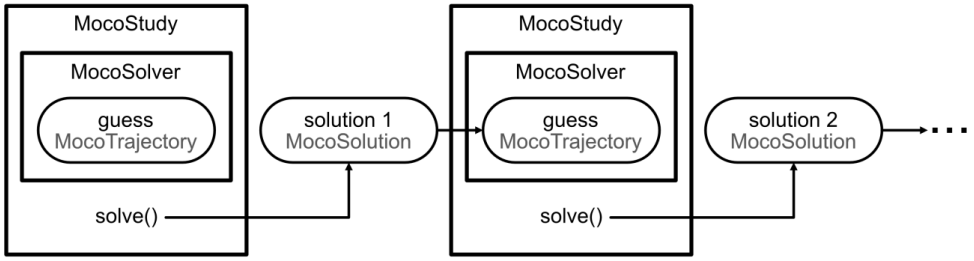

Figure 14: OpenSim Moco iteration structure (extracted from [6]).

For standard problems, Moco provides two solving tools, *MocoInverse* and *Mocotrack* (See Figure 15):

MocoInverse: finds the controls of muscles or actuators that achieve a given motion and minimize the cost function. That is called the muscle/actuator redundancy problem [7]. Prescribing a motion has the advantages of the problem becoming more robust and faster to solve since the non-linear multibody dynamics stop taking part. The disadvantages are that it is not possible to predict deviations from the provided motion or to use contact models, so it must apply measured external forces instead. That usually requires adding non-existent actuators to resolve inconsistencies when solving the problem.

MocoTrack: finds the motion (states) and controls that minimize the error between motion data and the associated model quantities. It can use contact models and it is useful for predicting deviations from experimental data. Both solving tools only require the OpenSim model and motion data as inputs.

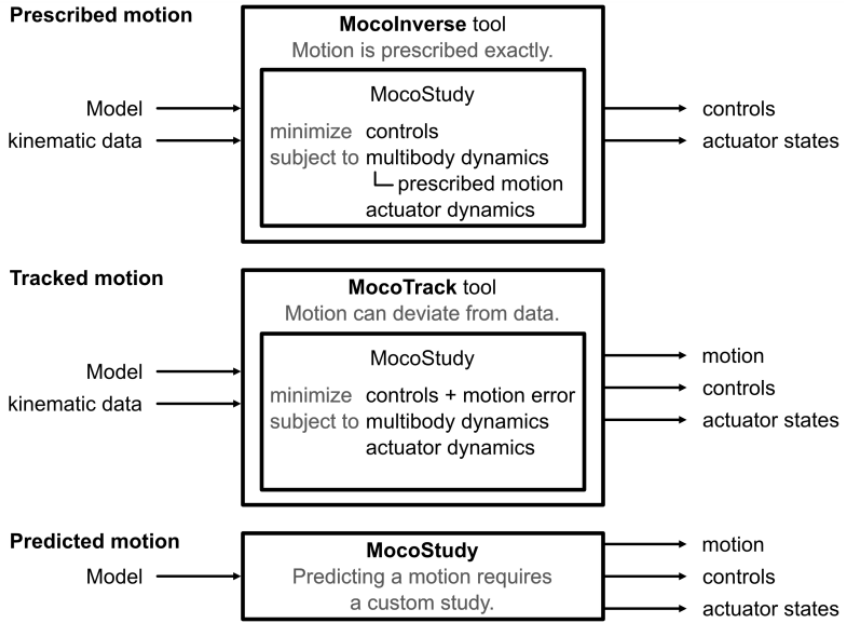

Figure 15: Solving prescribed motion, tracked motion, and predicted motion problems (extracted from [6]).

3.6 Inverse Kinematics

After scaling the model and with the experimental walking motion data, an inverse kinematics process is done. Inverse kinematics (IK) is a tool that finds the model configurations that fit best with the recorded motion over time. More specifically, for every time step of the motion capture data, it finds the values of the generalized coordinates of the model that best match with the experimental markers data [18].

An IK problem can be presented as an optimal control problem, where for each time step (t_i) the generalized coordinates (q) are found while minimizing the following cost function:

$\min_q \sum_{i=1}^{m} w_i \parallel x_i^{exp} - x_i \parallel^2, i = 1, 2, ..., m$

Where $x_i(q)$ is the position vector of the i^{th} virtual marker; q is the generalized coordinates vector; x_i^{exp} is the i^{th} experimental marker; m is the number of markers, and w_i is the marker's weight.

To solve an IK problem with Moco, there is the *InverseKinematicsTool*, a tool for performing IK [18]. The inputs necessary are the *modelFile* (*.osim*), the *markersFile* (*.trc*), and the *timeBounds* of the problem. Additionally, markers and coordinates weights can be added. After running the tool, it returns the output of the IK, which is the *coordinatesFile* (*.sto*), which can also be converted into a *statesFile* (*.sto*). The predicted motion can be visualized with Moco or at the OpenSim GUI. Figure 16 shows captures of the IK solution visualized in the OpenSim GUI:

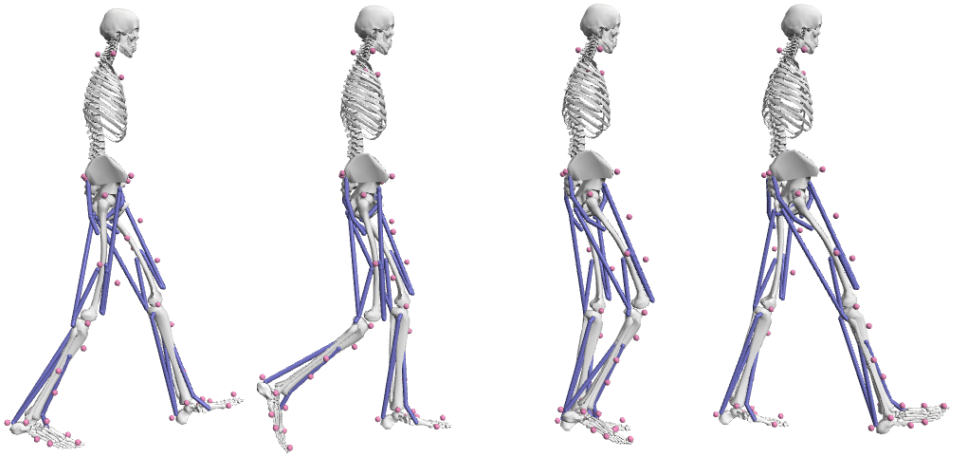

Figure 16: Inverse kinematics solution (extracted from [6]).

3.7 Tracking Optimization

Once the IK is finished, the next step is to conduct a tracking optimization problem. As explained in previous sections, it consists of finding the states and controls that minimize a cost function that includes a tracking term, among others.

First, the optimization problem is defined. A *MocoStudy* is created and the *MocoProblem* and *MocoSolver* are defined.

The input files needed for the *MocoStudy* are:

- The model file: The *.osim* file of the scaled model.
- The states file: the *.sto* file obtained after the IK problem that contains the reference motion for the tracking goal.
- External force file: a *.xml* file containing the experimental ground contact forces for the contact tracking goal.

The *MocoProblem* is set with the following parameters:

- The scaled model: *gait10dof18musc-scaled.osim*
- The initial and final time bounds: the same time bounds used with the IK problem.
- The experimental ground contact forces are added to the model as external loads.

The cost function is defined as the sum of a series of weighted cost functions using MocoGoal tools [17]:

- *StateTrackingGoal* (J_1): This function computes the difference between the state variable x and a reference x_{exp}, obtained from the previous IK problem.

  $J_1 = \int_{t_0}^{t_1} \left\| x^2 - x_{exp}^2 \right\| dt$

  This cost function can also be applied for tracking activations, forces, etc.

- *ControlGoal* (J2): This function is the sum of the absolute value of the control variables (x_c), raised to an exponent (p), integrated over the phase (with time bounds t_i, t_f), and divided by the displacement of the system (d). The formula used is the following:

  $J_2 = \frac{1}{d} \int_{t_i}^{t_f} \sum_{c \in C} w_c \mid x_c(t) \mid^p dt$

  In this equation d is the displacement of the system, C is the set of control signals, w_c is the weight for control c, $x_c(t)$ is the control signal c, and p is the exponent.

  In this tracking optimization problem, the weights of the controls vary. The controls that correspond to reserve, residual, and exoskeleton forces or torques each have their control weight.

- *ActivationSquaredGoal* (J_3): This function calculates the sum of squared activations (cita).
- *InitialActivationGoal* (J_4): This function is to ensure that muscle initial activations and excitations are different. It is an endpoint constraint goal (cita)

The total cost function (J_{total}) is the following:

$J_{total} = \sum_1^n w_i * J_i$

Where n is the number of smaller cost functions and w_i is the i-th function's weight. Depending of the objective of the optimization, the weights of each function vary. If it is more important to minimize a function than another, the first one will have a higher function weight.

Once the *MocoProblem* is all defined, the next step of the tracking optimization problem is to create the *MocoSolver*. The solver used is the *MocoCasADiSolver*, which uses the CasADi automatic differentiation and optimization library [5].

The following settings and parameters are added to the MocoSolver:

- The multibody dynamics is set as *implicit*.
- The transcription scheme, to transform the linear problem to a non-linear one, is set as the *Hermite-Simpson* one.
- The optimization convergence and constraint tolerances are set. They determine when the solver will consider that a solution converges, and it is optimal to stop the iterations.
- The optimization sparsity detector is set as *random*. Sparsity in the derivative matrices in the problem is what makes the direct collocation method fast. The CasADi solver determines the sparsity pattern of the matrices, to find "zeros", and solve the problem faster. When it is set as random, the trajectories used to detect "zeros" are random.
- The optimization finite difference scheme is set as *forward*. This scheme uses forward differences to approximate derivatives. It is faster than the central difference but less accurate.
- The number of mesh intervals used in the optimization and the optimization maximum number of iterations are set. When the optimization reaches the maximum amount of iterations, it stops.

Finally, the tracking optimization problem solver starts running.

4 Results and Discussion

In this section, the results of this thesis are presented. Three cases have been analyzed for the simulation of assistive walking: walking with an exoskeleton, walking with electrostimulation, and walking with a hybrid combination of both.

One of the design factors of each case and simulation is the activation bounds of the muscles variables. They vary depending on the type of resource is used for walking. The muscles are divided into three categories:

- Hip: Glutes and iliopsoas
- Knees: Hamstrings, biceps femoris, rectus femoris, and vastus intermedius
- Ankles: Gastrocnemius, soleus, and tibialis anterior

Another design factor is the control weights of the control variables that integrate the *Control-Goal* cost function. The controls have the same effort weight, but there are a few exceptions which are divided into the following types:

- Reserve weight: The weight of the control variables that correspond to extra actuators to help solve the optimization problem.
- Residual weight: The weight of the control variables of the pelvis.
- Exoskeleton weight: The weight of the control variables that correspond to the reserve actuators of the exoskeleton.

The simulations have been made with OpenSim Moco, using the *gait10dof18musc-scaled.osim*, solving the tracking optimization problem described in Section 3.7. The time bounds used correspond to half a gait cycle since it is considered a symmetric motion. Moco usually spends between 10-30 minutes solving each simulation.

4.1 Case 1: Walking with Exoskeleton

In this first case, the subject is walking with an exoskeleton. To simulate the subject's motor disability, leg muscles are debilitated. This translates to setting the model muscles' activations to zero. At the same time, the exoskeleton is added to the optimization problem. The exoskeleton considered is one able to generate torques on the knees and ankles. For that, reserve actuators are added to the model.

Table 5 shows the parameters of the simulation:

Activation Bounds:	Number of iterations	Time of simulation	Cost functions	Total cost function
Hip: [0.01, 1] Knees: [0.01, 0.01] Ankles: [0.01, 0.01]	212	19 min 49 sec	ControlEffort = 0.098750 MinActivations=0.000355 StateTracking=0.103452	0.202557

Table 5: Case 1 simulation parameters

The *MinActivations* function has the lowest value (0.0003) compared to the other functions, the *ControlEffort* and the *StateTracking* that are around the same value (0.1)

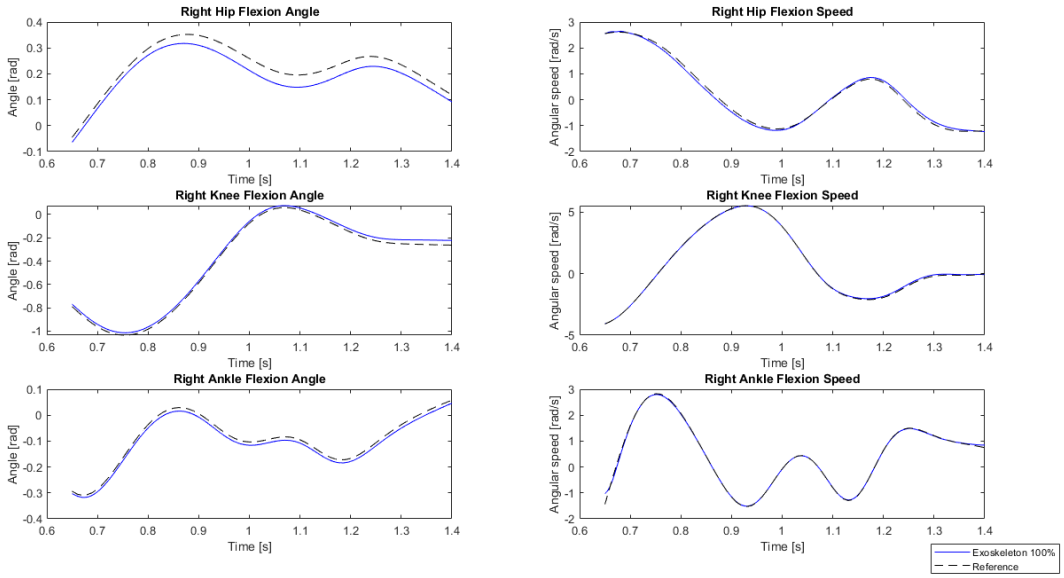

Figure 17: Case 1 States (generalized coordinates and velocities) for the right hip, knee, and ankle

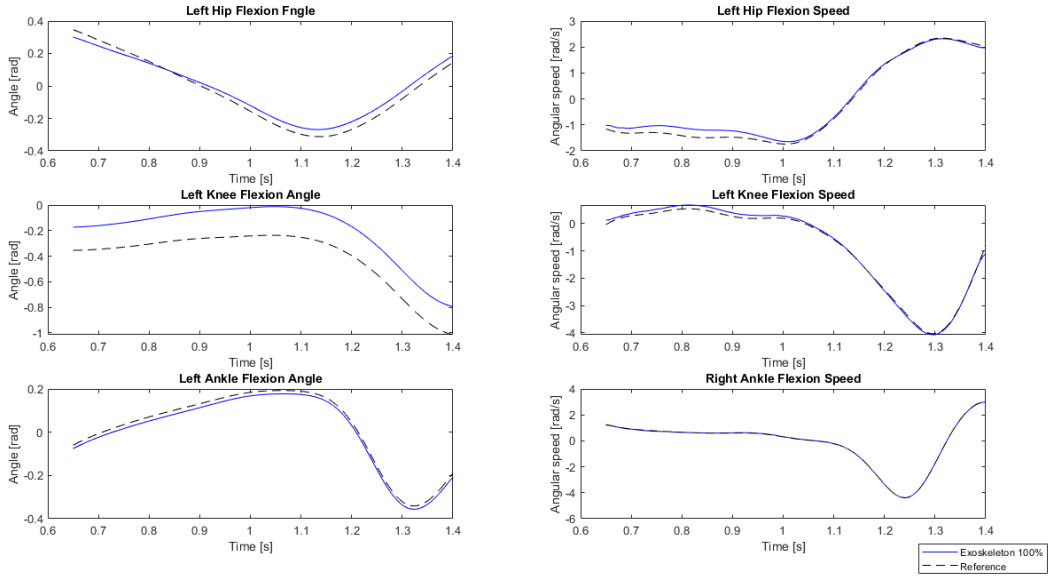

Figure 18: Case 1 States (generalized coordinates and velocities) for the left hip, knee, and ankle

Figures: Case 1 States (generalized coordinates and velocities) for the right/ left hip, knee, and ankle Figures 17 and 18 show the plots of the trajectories of the joints and their velocities. The trajectories in blue are the ones of the full exoskeleton assistance case and the ones in black are the reference trajectories obtained with the inverse kinematics problem. The reference trajectory is the healthy gait one. Both trajectories are very similar except on the left knee angle, where the trajectories have a higher offset.

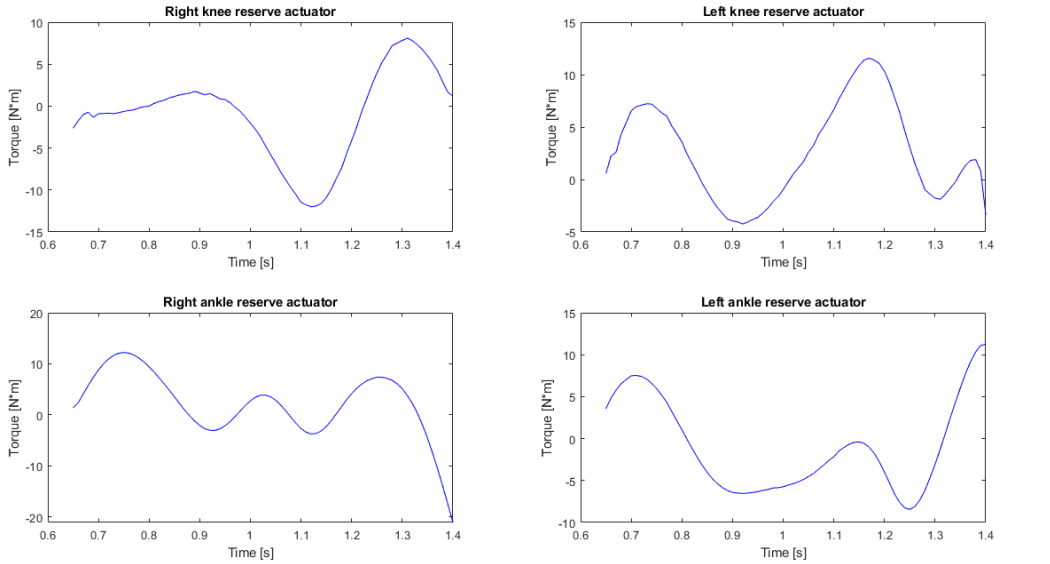

Figure 19: Case 1 Exoskeleton Torques

Figure 19 shows the torque of the reserve actuators of the knees and ankles that correspond to

the exoskeleton. The torques ranges go from -15 Nm to 8 Nm for the right knee actuator, from -4 Nm to 11 Nm for the left knee actuator, from -20 Nm to 11 Nm for the right ankle actuator, and from -8 Nm to 12 Nm for the left ankle actuator, approximately. The right ankle reserve actuator is the one that generates the higher absolute torque values.

4.2 Case 2: Walking with Neuroprosthesis

In the second case, the subject is walking with a neuroprosthesis assistive device. This means that the muscles are activated through functional electrical stimulation (FES). To simulate this case, the leg muscles' activations are limited since muscles cannot be fully activated with FES. Table 6 shows the parameters of the simulation:

Activation Bounds:	Number of iterations	Time of simulation	Cost functions	Total cost function
Hip: [0.01, 1] Knees: [0.01, 0.4] Ankles: [0.01, 0.4]	201	21 min 6 sec	ControlEffort = 0. 009669 MinActivations = 0. 009050 StateTracking = 0.028179	0. 046898

Table 6: Case 2 simulation parameters

ControlEffort function and the *MinActivations* functions have around the same value (0.009..) which is one-third of the *StateTracking* function value (0.028..)

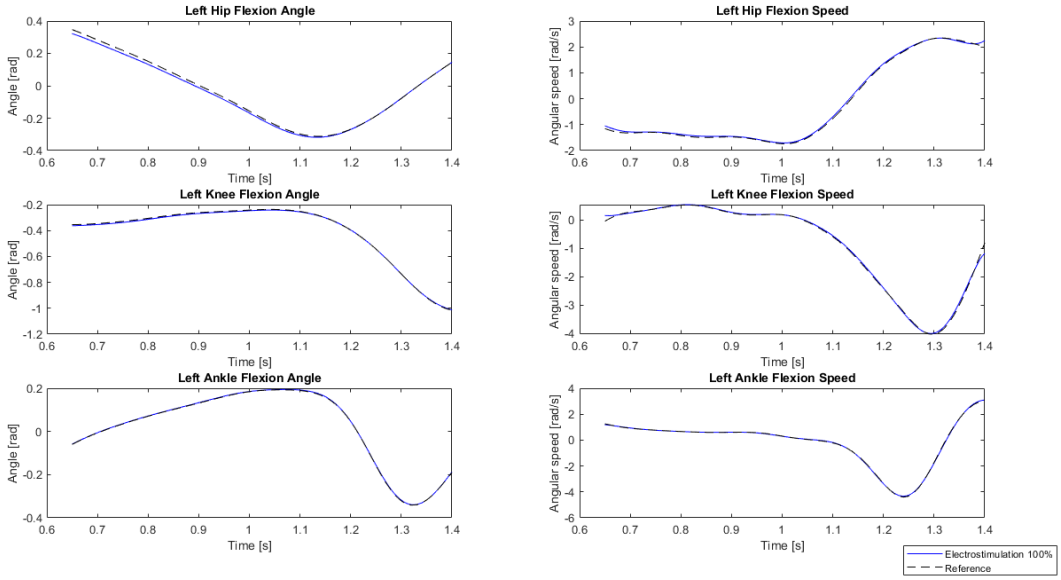

Figure 20: Case 2 States (generalized coordinates and velocities) for the right hip, knee, and ankle

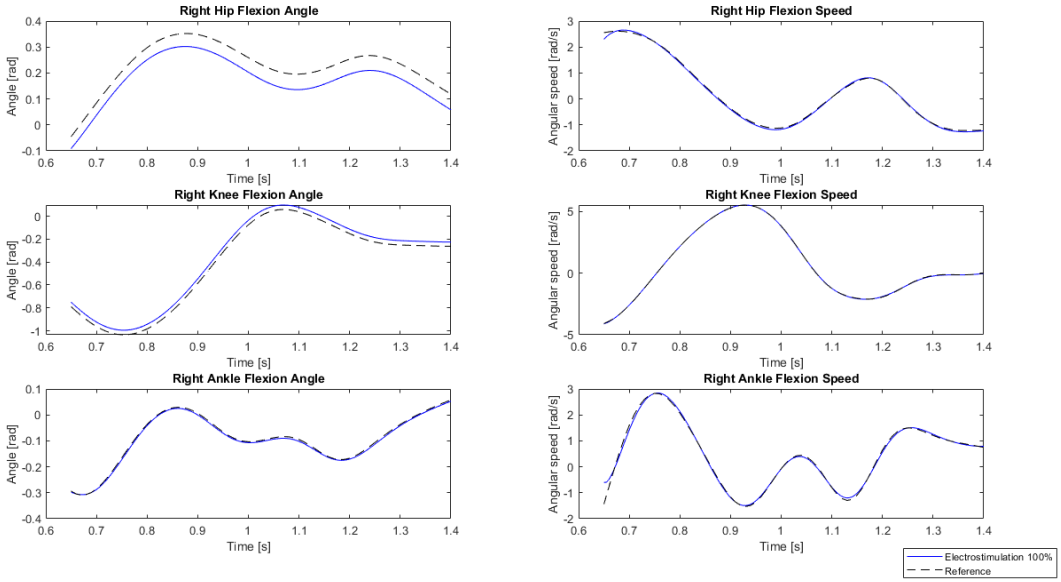

Figure 21: Case 2 States (generalized coordinates and velocities) for the left hip, knee, and ankle

Figures 20 and 21 show the plots of the trajectories of the joints and their velocities of case 2. The trajectories in blue are the ones of the full exoskeleton assistance case and the ones in black are the reference trajectories obtained with the inverse kinematics problem. As in case 2, both trajectories are very similar. An exception is the right hip angle, where the trajectory has a slight offset with the healthy gait trajectory.

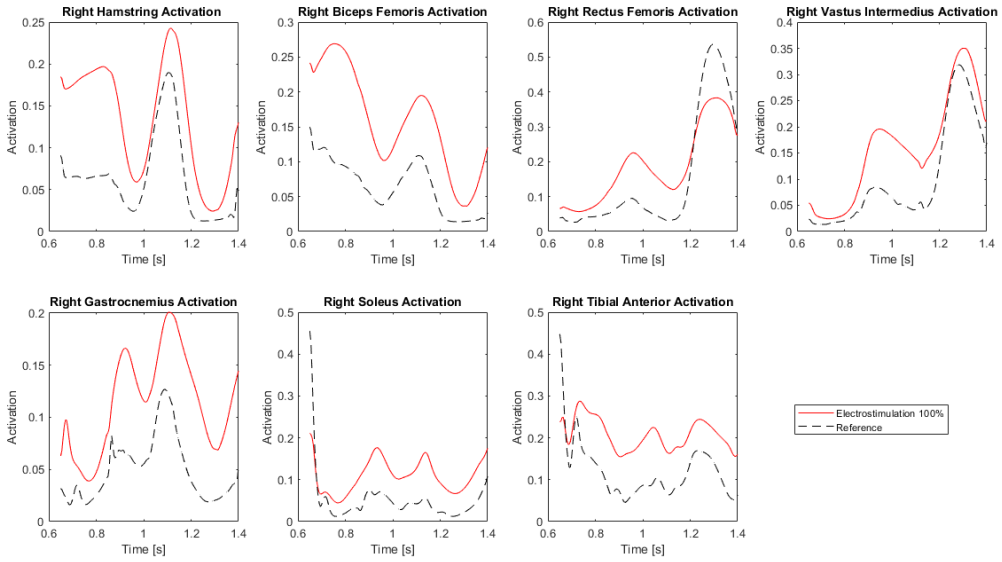

Figure 22: Case 2 Muscle Activations

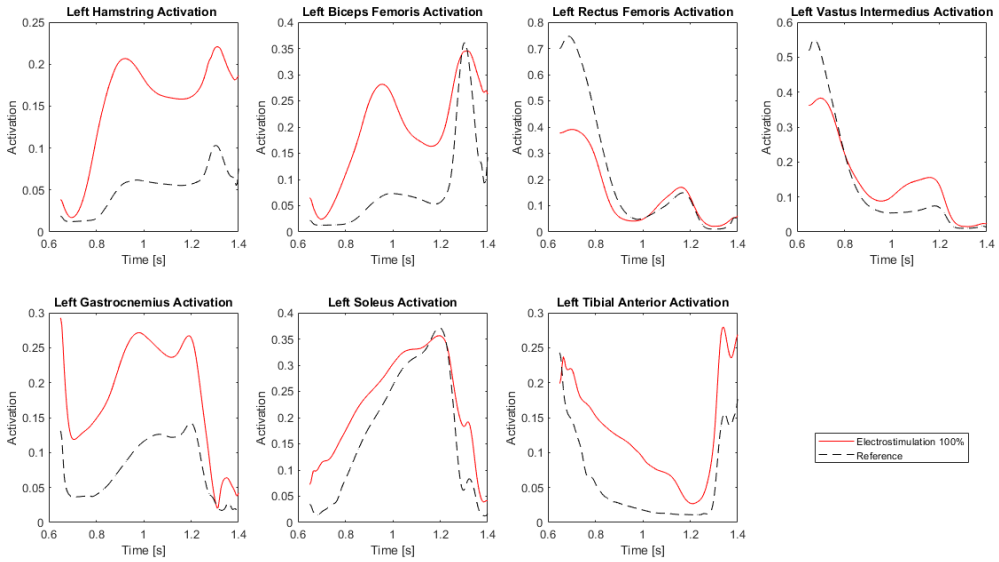

Figure 23: Case 2 Muscle Activations

Figures 22 and 23 show the muscle activations of the muscles that receive FES. The reference activations are from a simulation without muscle activation bounds. In most graphics, the red curve values (FES) are higher than the black curve values (reference). There are a few parts where the reference activation values are higher than the FES ones. That is when the reference activation values are above or near the FES muscle activation high bound (0.4). This can be appreciated on the "Right Rectus Femoris Activation" plot. The red curve starts above the black one but as both reach higher values (1.3 seconds), the reference one goes higher surpassing the FES one.

4.3 Case 3: Hybrid Assisted Walking

This last case studies different combinations of hybrid assistive walking. The subject walks with a combination of an exoskeleton and FES in the ankle, and FES in the knee. To simulate the ankle assisted combination, different values of muscle activation bounds (that corresponds to the FES) are applied while using the ankle reserve actuator (that corresponds to the exoskeleton generated torque). For the knee, the same muscle activation bounds as case 2 are used to simulate the full use of FES.

Table 7 shows the muscle activation bounds used, where *FESact* is the ankle muscles activation bound used:

Activation Bounds:
Hip: [0.01, 1]
Knees: [0.01, 0.4]
Ankles: [0.01, FESact]

Table 7: Case 3 Muscle Activation Bounds

Tables 8 and 9 show the simulation parameters for each FESact used:

FESact	Nº iter	Sim. time	Cost functions	Total cost function
0.01	359	19 min 44 sec	ControlEffort = 0.030969 MinActivations = 0.004247 StateTracking = 0.021262	0.056477
0.05	272	17 min 2 sec	ControlEffort = 0.012424 MinActivations = 0.012424 StateTracking = 0.019015	0.034232
0.10	267	16 min 22 sec	ControlEffort = 0.006077 MinActivations = 0.006077 StateTracking = 0.017713	0.027307
0.15	196	11 min 34 sec	ControlEffort = 0.008216 MinActivations = 0.008216 StateTracking = 0.008216	0.008216
0.20	377	23 min	ControlEffort = 0.007824 MinActivations = 0.006503 StateTracking = 0.016850	0.007824
0.25	233	13 min 54 sec	ControlEffort = 0.010336 MinActivations = 0.009524 StateTracking = 0.018061	0.037920
0.30	298	19 min 20 sec	ControlEffort = 0.006999 MinActivations = 0.006696 StateTracking = 0.013049	0.026744
0.35	327	19 min 31 sec	ControlEffort = 0.003652 MinActivations = 0.003168 StateTracking = 0.014777	0.021596
0.40	227	12 min 43 sec	ControlEffort = 0.005815 MinActivations = 0.005256 StateTracking = 0.016495	0.027565
0.45	274	17 min 9 sec	ControlEffort = 0.003913 MinActivations = 0.003704 StateTracking = 0.012829	0.020445
0.50	286	18 min 27 sec	ControlEffort = 0.005057 MinActivations = 0.004268 StateTracking= 0.016694	0.026018

Table 8: Case 3 Simulation Parameters (I)

FESact	Nº iter	Sim. time	Cost functions	Total cost function
0.55	310	18 min 38 sec	ControlEffort = 0.003326 MinActivations = 0.003027 StateTracking = 0.014567	0.020921
0.60	406	29 min 7 sec	ControlEffort = 0.003857 MinActivations = 0.003551 StateTracking = 0.013244	0.020653
0.65	277	16 min 58 sec	ControlEffort = 0.009524 MinActivations = 0.009330 StateTracking = 0.013427	0.032281
0.70	217	13 min 14 sec	ControlEffort = 0.019177 MinActivations = 0.018065 StateTracking = 0.023658	0.060900
0.75	220	15 min 2 sec	ControlEffort = 0.047027 MinActivations = 0.029557 StateTracking = 0.086136	0.162721
0.80	228	12 min 53 sec	ControlEffort = 0.005787 MinActivations = 0.005139 StateTracking = 0.016875	0.027801
0.85	354	22 min 39 sec	ControlEffort = 0.004880 MinActivations = 0.004103 StateTracking = 0.018918	0.027902
0.90	283	19 min 1 sec	ControlEffort = 0.019397 MinActivations = 0.018922 StateTracking = 0.016770	0.055090
0.95	419	25 min 38 sec	ControlEffort = 0.003133 MinActivations = 0.002964 StateTracking = 0.012191	0.018288
0.100	304	19 min 41 sec	ControlEffort = 0.003928 MinActivations = 0.003624 StateTracking = 0.014375	0.021926

Table 9: Case 3 Simulation Parameters (II)

The simulation number of iterations and times vary between 196 and 419 iterations, and 14 and 29 minutes, respectively. The *ControlEffort* function values vary between 0.003 and 0.047, the *MinActivations* ones between 0.003 and 0.019, and the *StateTracking* ones between 0.008 and 0.021. The total cost function value varies between 0.008 (when FESact = 0.20) and 0.163 (when FESact = 0.75).

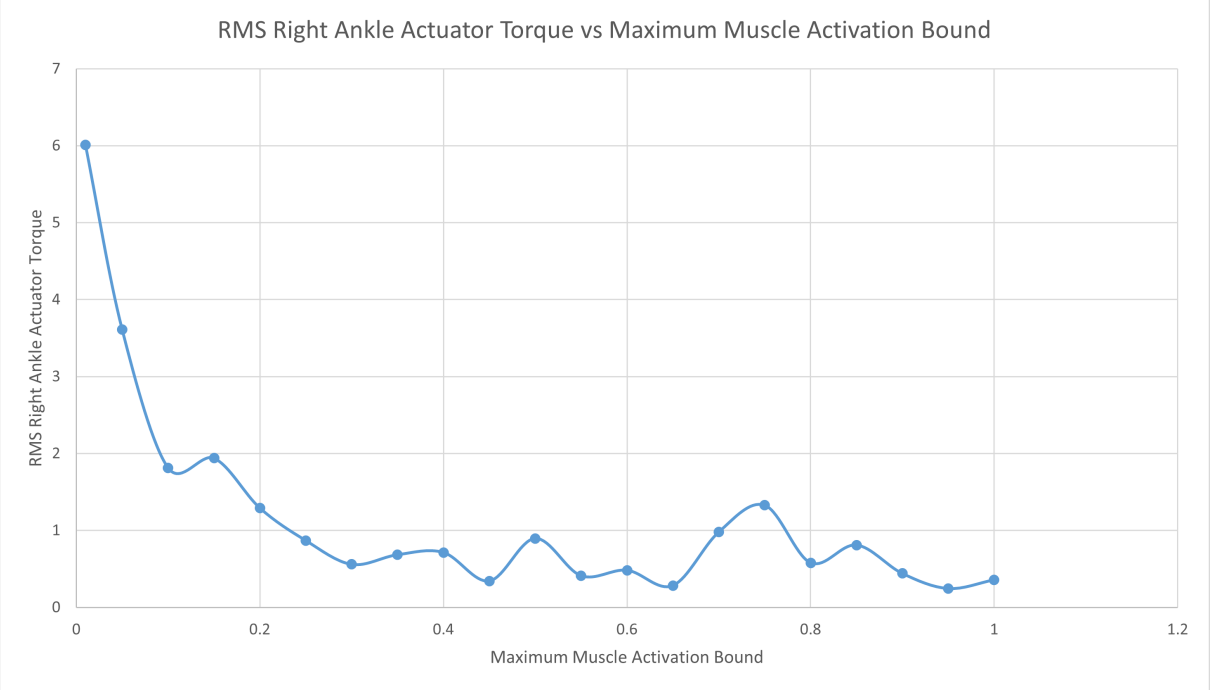

Figure 24: Case 3 Exoskeleton Torques RMS vs FES Activation Bounds

Figure 24 shows the root mean square (RMS) of the right ankle reserve actuator generated torque for every ankle muscle activation bound. The higher RMS values are of the 0.25 and 0.35 muscle activation bounds, while the lowest ones are of the 0.40 and 0.45 muscle activation bounds.

There is a visible tendency in the plot where the RMS Right Ankle Actuator Torque decreases rapidly as the Maximum Muscle Activation Bound (FESact) increases. Until the FESact reaches around 0.4, where it stays practically stable. It makes sense since more FES used should lead to needing a smaller exoskeleton torque and vice-versa. The point where the decreasing of the RMS Torque stabilizes is around the maximum activation bound reachable by using FES (section 2.6).

This plot is interesting because the relation between the exoskeleton and neuroprosthesis is shown. Depending on the characteristics of the patient, an optimal personalized hybrid device could be designed. Several studies link muscle activations to energy expenditure and fatigue ([33] for example). By computing the energy expenditure, of both the patient and the devices, and the fatigue generated on the patient, a more complex optimal problem could be considered.

Next, two of these cases are more thoroughly analyzed. They are named case 3.1 (FESact=0.1) and case 3.2 (FESact=0.3).

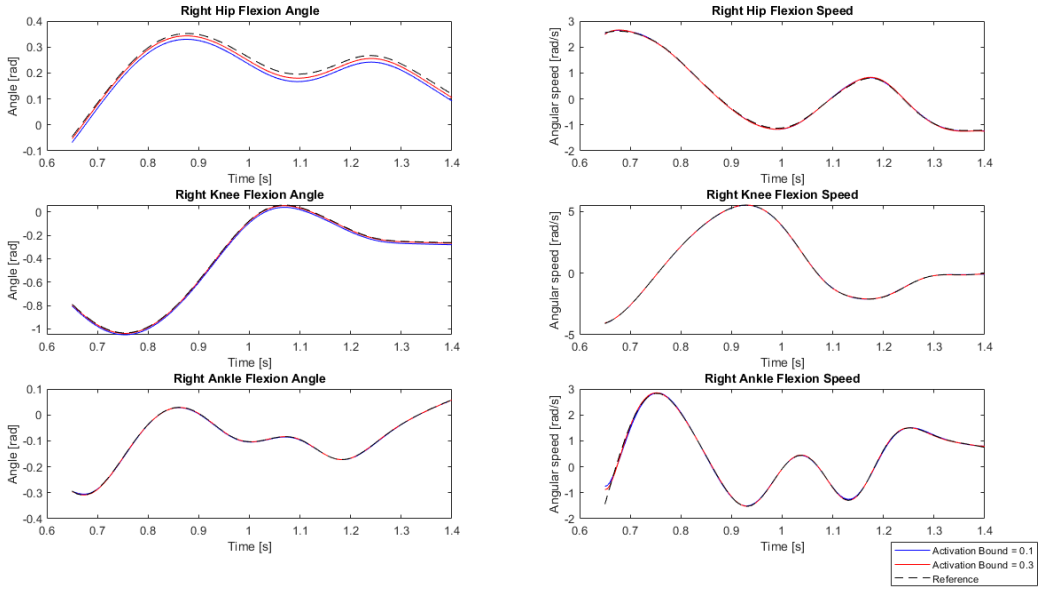

Figure 25: Case 3.1 (FESact=0.1) and 3.2 (FESact=0.3) States (generalized coordinates and velocities) for the right hip, knee and ankle

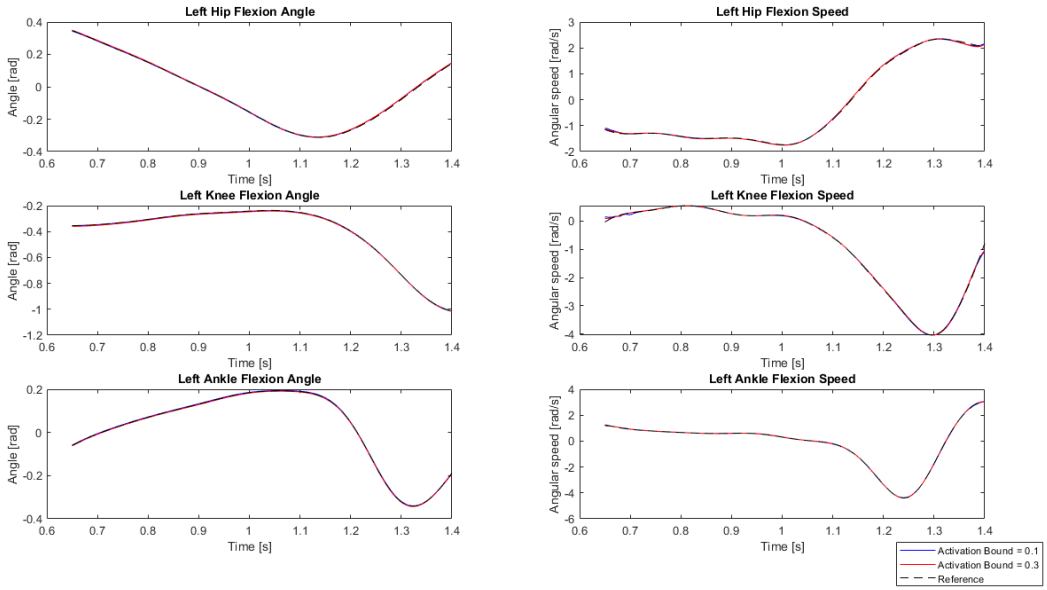

Figure 26: Case 3.1 (FESact=0.1) and 3.2 (FESact=0.3) States (generalized coordinates and velocities) for the left hip, knee and ankle

Figures 25 and 26 show the plots of the trajectories of the joints and their velocities of cases 3.1 and 3.2 and compared to the reference states. The different generalized coordinates and velocities are almost identical. The state that has the bigger offset between the cases shown is the "Right Hip Flexion Angle".

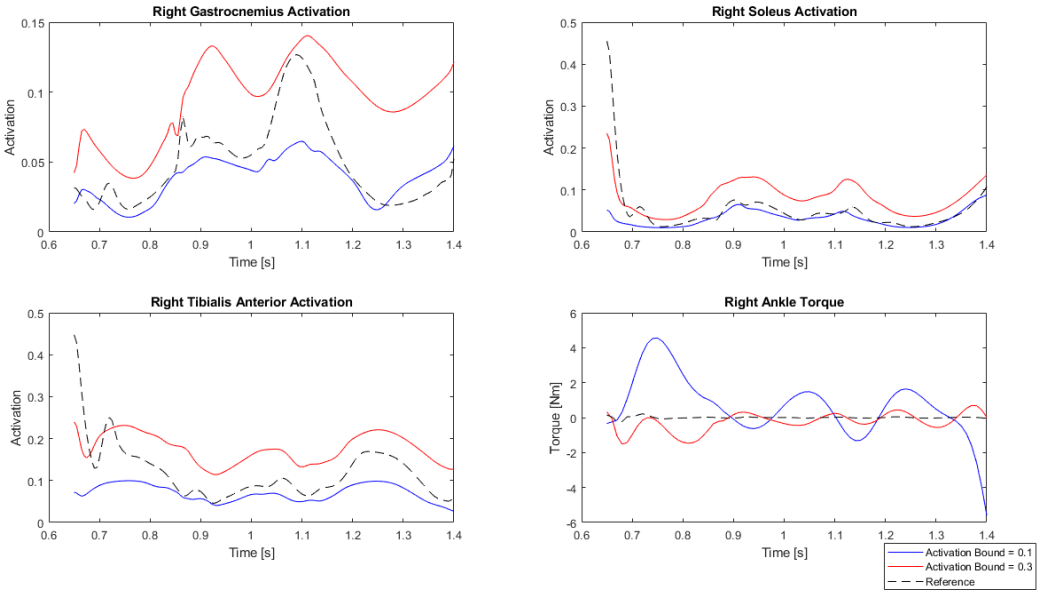

Figure 27: Case 3.1 (FESact=0.1) and 3.2 (FESact=0.3) Right Ankle FES Muscle Activation Bounds and Exoskeleton Torque

Figure 27 show the ankle muscle activations and torque generated corresponding to the hybrid strategy for cases 3.1 and 3.2, and the reference. The muscle activation curves of all cases follow a similar shape but with different offsets. Regarding the “Ankle Torque” generated by the exoskeleton, the reference one is very uniform, followed by the Case 3.2 torque. Lastly, Case 3.1 torque is the one that follows the least uniform curve. It makes sense since is the case where the FESact bound is lowest, so the exoskeleton has to generate more "work".

5 Project Impact

5.1 Social Impact

This project explores hybrid strategies for assisted walking, combining wearable robotics and neuroprostheses. Further research in this field can have a great impact on the development of assistive solutions and rehabilitation for patients suffering from motor disabilities.

5.2 Environmental Impact

This thesis has had a low environmental impact since all the research, meetings, simulations, and writing has been done using a laptop. As a result, the environmental impact has only been the electricity consumed with the laptop.

5.3 Economic Impact

The economic cost of this project takes into account the resources and time dedicated. To calculate the cost of the resources of the project, the useful life or work time and the cost per hour are estimated.

The laptop has an estimated life span of 5 years and a price of 750€. A usage of 10h per day, 5 days a week has been considered. It amounts to 13000 hours of useful life. Then, the cost per hour of the laptop is 0,0577 €/h. The estimated time referred to the project has been 300 hours.

For the software, Moco is completely free and the Matlab license lasts a year and costs 250€. The Matlab license's cost per hour amounts to 0,0286€/h while the usage time has been 150 hours.

Regarding the electric energy used by the laptop, the power is 50W and the time used has been estimated at 300 hours. The price has been considered 0,14€/kWh.

Lastly, the supervisors have dedicated 40 hours earning 50€/h, and the student has dedicated 300 hours, being a student's salary 8€/h.

The total cost of the project amounts to 4423,7€ (See Table 10):

Cost factor	Price [€]	Life span [years]	Cost per hour [€/h]	Working time [hours]	Cost related to the project [€]
Laptop	750	5	0,0577	300	17,31
Matlab license	250	1	0,0286	150	4,29
Electrical energy			0,0070	300	2,1
Supervisors			50	40	2000
Student			8	300	2400
Total Cost					4423,7

Table 10: Cost of the project

Conclusions

The present Final Project for the Bachelor's Degree in Industrial Technology Engineering has designed hybrid strategies of assisted walking combining wearable robotics and neuroprostheses to help patients with a damaged neural system. It has been achieved by programming optimal control problems of walking motion with a patient-device model. Experimental data obtained previously at the Biomechanics Laboratory of the Polytechnic University of Catalonia, and the OpenSim and Matlab software have been used.

Throughout the first part of this project, a lot of research and learning on biomechanics and optimal control has been done. Afterward, there has been a familiarization with the OpenSim Moco software toolkit in the Matlab interface. At last, different strategies of assisted walking have been simulated through optimal control problems.

For the simulations, an OpenSim model adapted with a subject's experimental data has been used. Besides, motion captures of a healthy gait cycle have been employed as a starting point and reference for predicting motion with optimal control.

The results from the strategies simulated have been analyzed and compared to each other and to a reference motion. The kinematics and dynamics of the predicted motion, along with the simulation parameters have been evaluated.

All things considered, the results have been found satisfactory. Hybrid-assisted walking strategies have been successfully obtained. The relation between the exoskeleton assistance needed and the FES used in the hybrid strategy has been analyzed. However, further research should be done to have a significant impact.

These assisted walking strategies are aimed to help recover or improve the autonomy of patients with a damaged neural system, such as spinal cord injury patients. Possible next steps of this thesis would be to analyze the energy expenditure, from the patient and the assistive devices, and the fatigue generated on the patient. From there, more personalized assistive walking strategies could be developed. Depending on the patient's condition and other circumstances, such as the assistive device potential, the solutions could be more focused on rehabilitation or just on giving autonomy to the patient.

Acknowledgments

Firstly, I would like to express my sincere gratitude towards my supervisors, Albert Peiret Gimenez and Josep Maria Font Llagunes. This thesis would have been impossible without their guiding, support, and advice. I am very grateful for the opportunity given to do this project.

I also want to thank the BIOMEC group, for the help and insights provided during the meetings.

Bibliografia

[1] ABLE, *ABLE Human Motion*, Available at: https://www.ablehumanmotion.com/es/exoesqueleto-able/ (Accessed 19 June 2021)

[2] Alizadeh Arsalan, Dyck Scott Matthew, Karimi-Abdolrezaee Soheila, *Traumatic Spinal Cord Injury: An Overview of Pathophysiology, Models and Acute Injury Mechanisms*, Frontiers in Neurology, Volume 10, 2019, Page 282, ISSN 1664-2295, DOI: 10.3389/fneur.2019.00282

[3] Andersson, J.A.E., Gillis, J., Horn, G., *CasADi: a software framework for nonlinear optimization and optimal control*, Math. Prog. Comp. 11, Pages 1–36, 2019, DOI: 10.1007/s12532-018-0139-4

[4] API: OpenSim, *InverseKinematicsTool Class Reference*, Available at: https://simtk.org/api_docs/opensim/api_docs/classOpenSim_1_1InverseKinematicsTool.html#details (Accessed 13 August 2021)

[5] CasADi, *CasADi*, Available at: https://web.casadi.org/ (Accessed 16 August 2021)

[6] Christopher L. Dembia, Nicholas A. Bianco, Antoine Falisse, Jennifer L. Hicks, Scott L. Delp, *OpenSim Moco: Musculoskeletal optimal control*, bioRxiv 839381y, 2019, DOI: 10.1101/839381

[7] De Groote, F., Kinney, A.L., Rao, A.V., *Evaluation of Direct Collocation Optimal Control Problem Formulations for Solving the Muscle Redundancy Problem*, Ann Biomed Eng 44, Pages 2922–2936, 2016, DOI: 10.1007/s10439-016-1591-9

[8] Harra R. Sandrow-Feinberg, John D. Houlé, *Exercise after spinal cord injury as an agent for neuroprotection, regeneration and rehabilitation*, Brain Research, Volume 1619, Issue 12, 2015, Pages 12-21, ISSN 0006-8993, DOI: j.brainres.2015.03.052

[9] Hartmann, Matthias Kreuzpointner, Florian Haefner, R Michels, Hartmut Schwirtz, A Haas, Johannes-Peter, *Effects of Juvenile Idiopathic Arthritis on Kinematics and Kinetics of the Lower Extremities Call for Consequences in Physical Activities Recommendations*, HINDAWI, International journal of pediatrics, 2010, DOI: 10.1155/2010/835984

[10] H.Hatze, *The meaning of the term 'biomechanics'*, ELSEVIER, Journal of Biomechanics, Volume 7, Issue 2, 1974, Pages 189-190, DOI: 10.1016/0021-9290(74)90060-8

[11] John T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Program-*

ming, 2010, ISBN 978-0-89871-688-7, DOI: 10.1137/1.9780898718577

[12] K. Mombaur, *Optimal Control for Applications in Medical and Rehabilitation Technology: Challenges and Solutions*, Advances in Mathematical Modeling, Optimization and Optimal Control, Springer Optimization and Its Applications 109, 2016, DOI: 10.1007/978-3-319-30785-5$_5$

[13] Maryam Khamar, Mehdi Edrisi, Mohsen Zahiri, *Human-exoskeleton control simulation, kinetic and kinematic modeling and parameters extraction*, MethodsX, Volume 6, 2019, Pages 1838-1846, DOI: 10.1016/j.mex.2019.08.014

[14] Michael J. DeVivo DrPhH, Bette K. Go BA Amie B. Jackson MD, *Overview Of The National Spinal Cord Injury Statistical Center Database*, The Journal of Spinal Cord Medicine, Volume 25, Issue 4, 2002, Pages 335-338, DOI: 10.1080/10790268.2002.11753637

[15] Mukund Srivastava, Mudit Srivastava, Piyush Sagar, Mamatha T.G, *Simulation of human gait for design of lower extremity exoskeletons – A review*, Materials Today: Proceedings, Volume 44, Part 6, 2021, Pages 4485-4491, DOI: 10.5772/19977

[16] Nuno A. Silva, Nuno Sousa, Rui L. Reis, António J. Salgado, *From basics to clinical: A comprehensive review on spinal cord injury*, Progress in Neurobiology, Volume 114, 2014, Pages 25-57, ISSN 0301-0082, DOI: 10.1016/j.pneurobio.2013.11.002

[17] OpenSim, *OpenSim: MocoGoal Class Reference*, Available at: https://opensim-org.github.io/opensim-moco-site/docs/0.4.0/class_open_sim_1_1_moco_goal.html (Accessed 16 August 2021)

[18] OpenSim Documentation, *Inverse Kinematics*, Available at: https://simtk-confluence.stanford.edu:8443/display/OpenSim/Inverse+Kinematics (Accessed 13 August 2021)

[19] OpenSim Documentation, *Musculoskeletal Models*, Available at: https://simtk-confluence.stanford.edu:8443/display/OpenSim/Musculoskeletal+Models/#MusculoskeletalModels-OpenSimCoreModels (Accessed 30 March 2021)

[20] OpenSim Documentation, *Scaling*, Available at: https://simtk-confluence.stanford.edu:8443/display/OpenSim/Scaling (Accessed 13 August 2021)

[21] OpenSim Moco, *DeGrooteFregly2016Muscle Class Reference*, Available at: https://opensim-org.github.io/opensim-moco-site/docs/0.2.0/class_open_sim_1_1_de_groote_fregly2016_muscle.html (Accessed 13 August 2021)

[22] OpenSim Moco, *OpenSim Moco Documentation*, Available at: https://opensim-org.github.io/opensim-moco-site/docs/0.3.0/index.html (Accessed 13 August 2021)

[23] Pallarès, R., *Optimal Control Prediction of Dynamic Consistent Walking Motions*, Available at: https://upcommons.upc.edu/handle/2117/111033 (Accessed 8 April 2021)

[24] Perry, Jacquelin, *Gait Analysis: Normal and Pathological Function*, SLACK Incorporated, 1992, ISBN 978-1-55642-192-1

[25] Phillip Correia Copley, Aimun A.B. Jamjoom, Sadaquate Khan, *The management of traumatic spinal cord injuries in adults: a review*, Orthopaedics and Trauma, Volume 34, Issue 5, 2020, Pages 255-265, ISSN 1877-1327, DOI: 10.1016/j.mporth.2020.06.002

[26] Philip E. Gill, Walter Murray, and Michael A. Saunders, *CasADi: a software framework for nonlinear optimization and optimal control*, SIAM Rev. 47(1), Pages 99–131, 2006, DOI: 10.1137/S0036144504446096

[27] Sancho-Bru, Joaquín and Pérez-González, Antonio and Mora, Marta and León, Beatriz and Vergara, Margarita and Iserte, Jose and Rodríguez-Cervantes, Pablo-Jesús and Morales, Antonio, *Towards a Realistic and Self-Contained Biomechanical Model of the Hand*, Theoretical Biomechanics, 2011, Pages 212–240, ISBN 978-953-307-851-9, DOI: 10.5772/19977

[28] Seth A, Hicks JL, Uchida TK, Habib A, Dembia CL, Dunne JJ, *OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement*, PLoS Comput Biol 14(7): e1006223, 2018, ISSN 0006-8993, DOI: 10.1371/journal.pcbi.1006223

[29] Sheperd Center, *Spinal Cord Injury Details: Levels of Injury*, 2020, Available at: https://www.spinalinjury101.org/details/levels-of-injury (Accessed 4 August 2021)

[30] SimTK, *OpenSim Moco*, Available at: https://simtk-confluence.stanford.edu:8443/display/OpenSim/Scaling (Accessed 13 August 2021)

[31] Singh A, Tetreault L, Kalsi-Ryan S, Nouri A, Fehlings M, *Global prevalence and incidence of traumatic spinal cord injury*, Clin Epidemiol, Volume 6, 2014, Pages 309—331, DOI: 10.2147/CLEP.S68889

[32] Steven C. Cramer, Lindsey Lastra, Michael G. Lacourse, Michael J. Cohen, *Brain motor system function after chronic, complete spinal cord injury*, Brain, Volume 128, Issue 12, 2005, Pages 2941–2950, DOI: 10.1093/brain/awh648

[33] UMBERGER BRIAN, KARIN G.M. GERRITSEN PHILIP E. MARTIN, *A Model of Human Muscle Energy Expenditure*, Computer Methods in Biomechanics and Biomedical Engineering, Volume 6:2, Pages 99-111, DOI: 10.1080/1025584031000091678

[34] Wächter, A., Biegler, L., *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program 106, Pages 25–57, 2006, DOI: 10.1007/s10107-004-0559-y

[35] World Health Organization, *Spinal Cord Injury*, 2013, Available at: https://www.who.int/news-room/fact-sheets/detail/spinal-cord-injury (Accessed 4 August 2021)

[36] Yi-Chung Lin, Marcus G. Pandy, *Three-dimensional data-tracking dynamic optimization simulations ofhuman locomotion generated by direct collocation*, Journal of Biomechanics, Volume 59, May 2017, DOI: 10.1016/j.jbiomech.2017.04.038