UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

MÁSTER UNIVERSITARIO EN INGENIERÍA BIOMÉDICA

TRABAJO FIN DE MÁSTER

DESIGN AND DEVELOPMENT OF AN EXOSKELETON CONTROL SYSTEM FOR A ROBOTIC PLATFORM

Pablo Romero Sorozábal

2020

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

MÁSTER UNIVERSITARIO EN INGENIERÍA BIOMÉDICA

TRABAJO FIN DE MÁSTER

DESIGN AND DEVELOPMENT OF AN EXOSKELETON CONTROL SYSTEM FOR A ROBOTIC PLATFORM

Autor
Pablo Romero Sorozábal

Tutores

Dr. D. Alvaro Gutiérrez Martín
Dr. D. Eduardo Rocon de Lima

2020

Resumen

La parálisis cerebral constituye la causa más común de discapacidad física en niños, con una prevalencia de alrededor de 2 por cada 1000 neonatos. Se trata de una condición permanente resultado de lesiones cerebrales que producen un grupo de trastornos de movimiento y de postura que comienza en las primeras fases de la infancia y persiste durante toda la vida generando costos de aproximadamente 800,000$ por paciente. Dado que esta condición es una enfermedad no degenerativa, pero que empeora durante las fases de desarrollo de los niños, el objetivo del tratamiento es proporcionar terapias tempranas para mejorar la funcionalidad y las capacidades finales de los pacientes.

El proyecto de investigación CPWalker del "Centro Superior de Investigaciones Científicas" tiene como objetivo proporcionar una solución para este problema proporcionando nuevas terapias robóticas para la temprana rehabilitación de los pacientes con parálisis cerebral.

El objetivo principal de este Trabajo Fin de Master es colaborar en este proyecto diseñando e implementando un nuevo sistema de control modular para el exoesqueleto del CPWalker. El sistema de control debe comunicarse con el resto de los módulos existentes en la plataforma robótica y proporcionar varios modos de control que permitan diferentes niveles de interacción humano-robot para las diferentes terapias de rehabilitación.

Concluido el proyecto, se ha desarrollado e implementado un nuevo módulo capaz de controlar las cuatro articulaciones del exoesqueleto del CPWalker validado con la rodilla derecha del exoesqueleto.

Palabras clave: Rehabilitation robotics, Exoskeletons, Service robots, Control-systems, Legged locomotion.

Abstract

Cerebral palsy constitutes the most common cause of physical impairment in children with a prevalence of around 2 per 1000 live births. It is a permanent condition result from brain injury that describes a group of movement and posture disorders that begins in early childhood and persists throughout the lifespan with life time costs of about 800,000$ per patient. Since this condition is a non-degenerative disease, but worsens during the developmental phases of children, the treatment goal is to provide early therapies to improve the functionality and final capabilities of patients.

The CPWalker research project of the "Centro Superior de Investigaciones Científicas" aims to provide a solution for this problem by providing novel robotic therapies for the early rehabilitation of cerebral palsy patients.

This Master Thesis collaborates in this project by designing and implementing a new modular control system for the CPWalker's exoskeleton. The control system must communicate with the rest of the existing modules and provide several control modes allowing different levels of human-robot interactions for the different rehabilitation therapies.

At the end of this project, a new robust module capable of controlling the four joints of the CPWalker exoskeleton has been developed, implemented and validated in the right knee of the exoskeleton.

Keywords: Rehabilitation robotics, Exoskeletons, Service robots, Control systems, Legged locomotion

Acknowledgements

En primer lugar, me gustaría expresar mi gratitud a mis tutores de este Trabajo de Fin de Máster, Álvaro Gutiérrez y Eduardo Rocón. Gracias Álvaro por apenas conociéndome confiar en mi ilusión y darme la oportunidad y el apoyo para realizar este proyecto en el área por la cual decidí iniciar este máster. Gracias Eduardo por aceptarme en tu grupo y hacerme sentirme como otro más del equipo, aunque la mayoría de nuestras reuniones hayan sido online el apoyo lo he sentido como si fuese en persona. Gracias a ambos por vuestra cercanía vuestro tiempo y ganas de hacer.

Gracias a mis compañeros del CAR, a Julio quien siempre me ha ayudado incluso en tiempos de aislamiento, a Ricardo por el apoyo los consejos y la ilusión que me has trasmitido y a Verónica por aguantarme todas mis dudas sobre mecánica. Gracias a todos por los cafés de las 09:00 y las conversaciones a la vuelta en el autobús. Gracias por el fantástico grupo que sois.

Gracias a mis compañeros del máster con quienes empecé esta aventura y de donde han salido grandes amistades y momentos.

Y por último y no menos importante, gracias a mi familia, a mi madre Teresa, a mi padre Ángel y a mi hermano Carlos. Gracias por escucharme, entenderme y apoyarme en todas mis decisiones. No habrá océano suficientemente grande que pueda hacernos sentir lejos.

Contents

List of Figures

Table Index

Acronym List

CAN: Controller Area Network.

CIMT: Constraint-Induced Movement Therapy.

CP: Cerebral Palsy.

CSIC: Centro Superior de Investigaciones Científicas.

DAQ: Data Acquisition Systems.

FIR: Finite Impulse Respond.

GMFCS: Gross MotorFunction Classification System.

GPIO: General Purpose Input/Output.

PBWS: Partial Body Weight-Supported.

PBWSTT: Partial Body Weight-Supported Treadmill Training.

PID: Proportional Integral Derivative.

RAGT: Robot-Assisted Gait Therapy.

ROS: Robot Operating System.

ROM: Range Of Motion.

RTOS: Real-Time Operating System.

SPI: Serial Peripherial Interface.

TCP: Transmission Control Protocol.

UDP: User Datagram Protocol.

UML: Unified Modeling Language.

Chapter 1

Introduction

This Master Thesis is a work carried out in collaboration with the "Centro de Automática y Robótica" (CAR) of the "Centro de Investigaciones Científicas" (CSIC), developed on the existing project CPWalker Robotic Platform. It encompasses the development and implementation of a modular control unit for a lower limb exoskeleton intended to support rehabilitation therapies in patients with cerebral palsy.

This chapter reviews basic concepts about cerebral palsy, current technologies designed for the support in cerebral palsy rehabilitation, the work motivation, objectives, and the document layout.

1.1. Cerebral Palsy

Cerebral palsy (CP) is a non-degenerative neurodevelopmental condition that describes a group of movement and posture disorders that begins in early childhood and persists throughout the lifespan [16]. It results from central nervous system injuries during the developing phase of the brain leading into lifelong motor disability, often accompanied by disturbances of sensation, perception, cognition, communication, and behavior.

Its overall prevalence is around 2 per 1000 live births (higher among low-birthweight) constituting the most common cause of physical impairment in children [17]. In terms of impairments, all these patients present motor impairments, and $25 - 80\%$ present additional ones such as cognitive, sensible, urogenital or endocrine pathologies like epilepsy and/or abnormal brain CT [18].

Regarding risk factors and causes, many studies conclude that multiple pathways contribute to the development of CP, each of them participating in its development in small proportions and in a multifactorial way [18]. Risk factors are categorized in prenatally, perinatally and postnatally and although many of the causes have not been already identified the best known are low birth weight (less than $2{,}500g$), intrauterine infection, multiple gestation, placental abruption and cerebral ischemia.

1.1.1. Diagnosis and classification

Cerebral palsy is a well know disease, the damage produced in the central nervous system generates primary, secondary and tertiary abnormalities that need to be detected as soon as possible to start the treatment [3]. The main primary abnormalities are: muscle control, dependence on primitive reflex patterns for ambulation, abnormal muscle tone, relative imbalance between muscle agonists and antagonists across joints and deficient equilibrium reactions. The secondary abnormalies are categorized as growth disorders because they appears during child development and define the final shape of the childs bone structure. While the tertiary abnormalities are the ones generated by the patient body to circumvent the primary and secondary abnormalities.

Since cerebral palsy covers a wide range of clinical presentations, it is needed a categorization of the patients into groups. According to [16] the classification must be divided in four dimensions: Motor abnormalities, Accompanying impairments, Anatomical and neuro-imaging findings and Causation and timing as shown in the Table 1.1.

Dimentions of classification		Description
CP Subtypes and Neurological findings	Nature and typology of the motor disorder	Observed hypertonia or hypotonia and/or spasticity, ataxia, dystonia or athetosis.
	Functional motor abilities	Limitations in motor function including orormotor and speech function.
Accompanying impairments		Presence of musculoskeletal problems and/or non-motor sensory or neurodevelopmental problems like hearing, visual or attentional impairments, cognitive deficits, behavioral and/or communicative.
Anatomical and neuro-imaging findings	Anatomic distribution	Body parts affected by the imparements.
	Neuro-imagin findings.	Biomedical imagine information (CT or MRI), e.g, ventricular enlargement, white matter loss or brain anomaly.

Causation and timing	Clearly identified causes generally in post-natal cases such as meningitis or head injury in infants.

Table 1.1: Cerebral palsy classification according to Rosenbaum et all. [16]

Also, according to [19],a more general classification depending on the type of paralysis is useful to segment in groups of patients, see Figure 1.1.

- Monoplegia: Paralysis of a single upper or lower limb.
- Hemiplegia: Paralysis of one side of the body, mainly effects the limbs.
- Diplegia: Paralysis of upper or lower limbs of both sides of the body.
- Tripegia: Paralysis of three limbs.
- Quadriplegia: Paralysis of all four limbs.

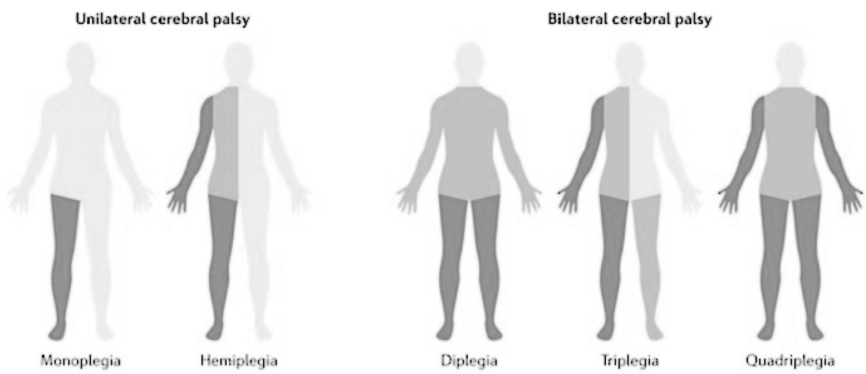


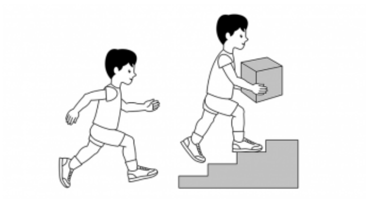
Figure 1.1: Types of paralysis. Recovered from [1]

However, children with CP experience motor function changes during growth and it is important to consider this changes in the prognostications. The Gross Motor Function Classification System for CP (GMFCS) [20] is designed to predict levels of motor funtion in CP patients and it is the most used scale for children with CP, see Table 1.2 and Figure 1.2.

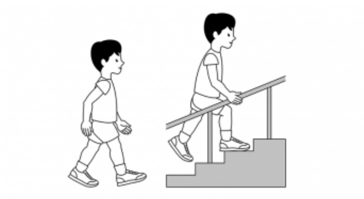
Level	Expected gross motor function
I	Children can run and jump but have limitations in speed, balance, and coordination.
II	Children can climb stairs holding onto a rail, but have limitations walking on irregular surfaces, in crowds or confined spaces, and at best minimal ability to perform gross motor skills.

III	Children may climb stairs holding onto a rail, propel a wheelchair manually, or are transported when traveling for long distances or uneven surfaces.
IV	Children may achieve self-mobility using a powered wheelchair.
V	Noticeable limitations in the control of limb movements, trunk postures, and ability to maintain anti-gravity head. Sitting and standing functional limitations. Children are transported and may achieve self-mobility using powered devices.

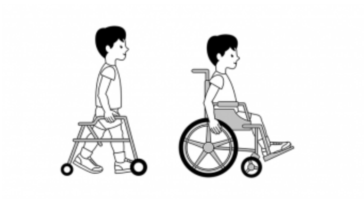
Table 1.2: Expected gross motor function.



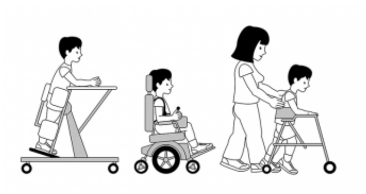
(a) Level I



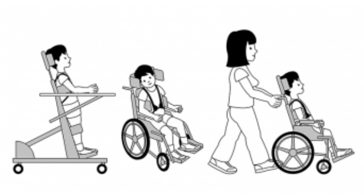
(b) Level II



(c) Level III



(d) Level IV



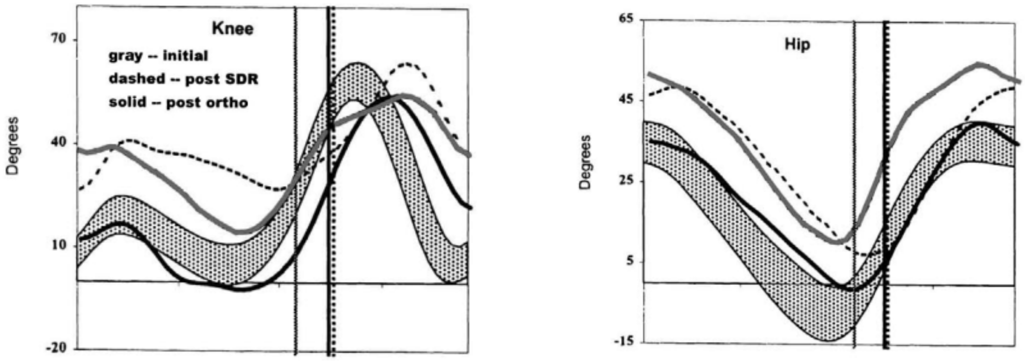
(e) Level V

Figure 1.2: Gross Motor Function Classification System Levels.Recovered from [2].

1.1.2. Gait rehabilitation therapies in patients with Cerebral Palsy

Currently there is no cure for CP, although, several treatments are used to maintain and improve the quality of life of the patients and prevent complications. These treatments depend on the patient's specific symptoms and are classified in: general strategies, physical therapy, medications, surgical treatments and external aids [21]. A review of the main general approaches for rehabilitation and treatment in patients with CP showed in the Table 1.3 and based on [22].

In terms of CP gait rehabilitation, nowadays dynamic gait analysis is mandatory for an optimal treatment [3]. The systems used to do these analysis are kinematic and kinetic studies of gait, Figure 1.3. These studies give us information about the treatment outcomes by comparing presurgery and postsurgery kinematics and kinetics. Oxygen consumption and electromiography are also used for these analysis.



(a) *Knee* (b) *Hip*

Figure 1.3: Sagittal plane preoperative and postoperative kinematics of a child [3].

Between the mentioned treatments, physical therapy is one of the most common interventions and usually a mandatory component in therapy programs [23]. Physical therapy is mainly given during the developing phases of the CP patients when their fundamental abilities and skills are developed, increasing the rehabilitation success rate in accordance with the tensity and repetition of the therapy as well as the patient's motivation [24]. The design of the therapy depends on the objective followed and the different types are divided in: Normalization of the quality of movement (e.g. physical therapy); and Development of skills necessary for the performance of activities of daily living (e.g. occupational therapy).

Summarizing, since the present work focuses on gait rehabilitation therapies for CP patients, a review of the main treatments for gait rehabilitation is provided [25]:

1. Physical and occupational therapy: Focused in walking, standing, stretching exercises, and flexibility.
2. Medication: Generalized to spasticity treatment.
3. Orthoses: Prevent deformities, contractures,and pain in children with CP.
4. Botulinum toxin: Treat localized spasticity.
5. Ferule and plaster: Avoid moderate contractures.
6. Multilevel orthopedic surgery: Two or more soft-tissue ans/or bony surgical procedures, at two or more anatomical levels during one unique operative procedure.
7. Neurosurgical procedures.
8. Partial Body Weight-Supported Treadmill Training (PBWSTT) and Constraint-Induced Movement Therapy(CIMT): Motor learning techniques that promotes the standardization of gait pattern involving sensory information and reflection components of gait.
9. Robot-Assisted Gait Therapy (RAGT): Effective tool to compensate and/or rehabilitate the functional skills of the patients.

Rehabilitation	Medical therapy	Therapeutic team approach	
Motor weakness	Strengthening, task-specific intensive therapy, functional neurostimulation	Assessment of joint alignment and avoidance of joint overuse and tendon strain and rupture, preventive approach against painful syndromes later in life	Educate patient and family to promotefunctional activities at home and in the community.
Tone abnormalities	Stretching, task-specific practice, functional neurostimulation, serial casting, orthotic devices	Oral medicines (baclofen and tizanidine), botulinum toxin injections, nerve blocks, intrathecal agents, treatment of disorders such as scoliosis and tendon contractures byorthopaedic surgery	Educate patient and family aboutpositive and negative effects of treatments and interventions, elicit feedback to monitor success, encourage self-directed stretching and use of prescribed devices.
Cognitive impairments	Neuropsychology and speech or language therapy after assessment of extent and precise nature of deficits (eg, attention vs memory)	Consider pharmacological treatment of attention deficits, monitor effects of other medications on cognition	Work with family and school on an individualised educational plan and adaptive communication device
Seizure disorder	-	Diagnose and treat as for patients without cerebral palsy	Educate patient and family about complications and risks
Psychosocial disorders	Neuropsychology, speech therapy, occupational therapy	Psychiatric assessment of affective and social disorders and counselling and pharmaco-therapy as appropriate	Empower family to use techniques for engaging patient and advocating for appropriate measures in school and in the community
Oromotor impairment	Speech therapy, occupational therapy	Nutritional and enteral supplements	Family engagement and education

Table 1.3: Multidisciplinary management of CP

1.1.3. State of art of robotic devices for rehabilitation

In recent years, many authors have increasingly emphasized promoting active therapies in CP rehabilitation that include intensive, repetitive, and specific training to improve neuroplasticity and rehabilitation [26]. Conventional therapies have tried to archive the mentioned goal without good results due to complexity of the exercises, lack of patient engagement and therapist's physical fatigue. Although, recently, several studies suggest that wearable rehabilitation robots can increase intensity, repetition and control of the therapies increasing the benefit of the sessions [24].

A therapy in which a robot is used in a session is called "Robot-assisted therapy". It may be defined as a form of physical therapy that uses a robotic device to help patients with impaired functional abilities to recover their function [27]. Since rehabilitation robots usually hold the patients weight and controlled its movements, physical therapist can easily focus in the patients, making it possible to perform therapies more active, intense and repetitive.

Furthermore, rehabilitation robots generally introduce different types, modes and levels of therapies, increasing patient-robot interaction, their participation in therapies, and therefore motivation during sessions, leading to an increase in duration and frequency of these.

Several devices for CP gait rehabilitation are currently in use. They are generally composed of a holding system and control movement system. The holding system is designed to hold in the weight of the patients in order to reduce the effects of gravity in case the patients cannot support their own weight and perform the rehabilitation exercises. There are several types of holding systems, walkers and arnesses the most common are the walkers. Regarding control movement systems, these systems are designed to help make the desired movement. Several types of movement control systems are currently in use, the most used are the powered orthosis.

Powered orthosis, also called robotic exoskeletons are wearable robots composed by mechanical end electrical modules designed to biomimetic human limbs and to couple them in some extend. These devices are used to empower human abilities and when used for rehabilitation they replicate the desired limb movements designed by the therapist. The use of sensors and actuators in their system architecture makes it possible to obtain quantitative information about the user performance, control in a precise way the desired amplitude and velocity of the movements and post process the kinematic and kinetic information to analyze the outcomes.

A revision of the commercially available rehabilitation devices is showed hereafter:

- **NFWalker**: Hybrid assistive device that gives weight and dynamic support to standing position and gait. Created by "Made for Movement", Norway, its design allows children suffering from CP to move around even if they have severe gait impairments [28], see Figure 1.4.

Figure 1.4: NFWalker hybrid assistive device. Recovered from [4].

- **Innowalk and Innowalk-Pro**: Also developed by "Made for Movement", these available robots are Partial Body Weight-Supported (PBWS) devices, static devices that generate gait patterns movements for disable people. They also allow a correct sitting and standing leg movements. See Figure 1.5.

(a) Innowalk *(b) Innowalk-Pro*

Figure 1.5: Innowalk and Innowalk-Pro, PBWS assistive devices. Recovered from [5]

- **Locomat**: Developed by Hocoma AG, it is a Robot-Assisted Gait Therapy (RAGT) device for adults and children suffering from different movement pathologies. It is the most widely used hospital rehabilitation robot. It is composed by several modules: a treadmill, a harness and an exoskeleton. These set of modules make it possible to adjust to almost all different patient sizes and perform repetitive movements mirroring the physiological walk. It also incorporates different intensities and gait velocities so the patient can evolve during the sessions adjusting these, it is an ideal way to approach a rehabilitation session. See Figure 1.6.

Figure 1.6: Locomat, the most used RAGT device worldwide. Recovered from [6].

- **Gait Trainer GT II RehaStim**: It is a gait therapy device designed to improve patient's ability to walk through repetitive training. It lifts the body of the patient with an harness reducing strain on therapists and hold the feet of the patient on two footplates. The movement of these footplates simulate the gait and induce low limbs gait movement. See Figure 1.7.

Figure 1.7: Gait Trainer GT II RehaStim. Recovered from [7].

- **Walkbot**: South korean rehabilitation medical device for adults and pediatric with neurological and musculoskeletal impairments (e.g. stroke, spinal cord injury or CP). It is a robot-assisted gait training system composed by a tradmile and a exoskeleton, really similar approach as the Locomat. It uses Hip/Knee/Ankle joint drive motors to give an accurate gait pattern for the patients. See Figure 1.8.

Figure 1.8: Walkbot. Recovered from [8].

1.2. CPWalker Robotic Platform

The CPWalker Robotic Platform is a rehabilitation robotic platform that enables a top down approach in the rehabilitation of CP patients(primarily children) developed by the *Centro Superior de Investigaciones Científicas* (CSIC). This robotic platform is made up of two main structures, a smart walker for patient ambulation and a portable exoskeleton for joint control, allowing successful robot-based therapies to be performed. A review of the system, base on [29], is given in this section.

The smart walker of the CPWalker, see Figure 1.9, is a ambulatory and weight holding system designed to give the optimal support and balance in the therapy sessions allowing the movement of the patient around the room. It resists a total of $80kg$ (exoskeleton+patient) and its subdivided in the following modules:

- **Drive system**: Module located in the back wheels, is responsible for generating the movement of the platform. The main contribution of this module is that it provides the necessary support for ambulation over-ground treatments in real rehabilitation environments, without needing a treadmill. It is composed of two independent gear motors coupled in each back wheel and other two encoders for the translation speed control.

- **Partial body weight support system**: This module is designed for the control of the discharge of the patients's body weight. This module allows a partial discharge of the user's weight during gait improving the rehabilitation [30]. It is composed by an electric linear actuator which compression and decompression controls the patient weight (allowing up to 45 kg unloading respect to the ground), a pontentiometer for the fine control of the patient's weight discharge and a load cell to measure the current weight supported.

- **System for the adaptation of hip height**: This module is designed for adjusting the platform for different patient' sizes by adjusting the hip joint of the exoskeleton at different heights. It is composed by a linear actuator $E21BX300 - U - 001$ (Bansbach easylift, Germany) composed of a hydraulic pump and two cylinder-pistons able to elevate the patient and support bending movements and a potentiometer to control the height.

The CPWalker's exoskeleton is a robot with a similar kinematic configuration as the lower human limbs, composed of four active joints (right and left knees and hips) and two passive (ankles). Its design is based on the passive NFWalker platform of "Made for Movement" in which various sensors and actuators have been implemented. The structure is mainly made up of $Aluminum7075$ [31] resulting in a lightweight,rigid and robust construction. The dimensions of the structure can also be adjusted to the different anthropomorphic sizes of the patients. The mechanical movement of the exoskeleton is limited to some ranges to adjust to over-ground walking and ovoid hazard to the patient, hip ranges: [60°, -40°]; knee ranges [90°, 0°].

The exoskeleton is composed by four active joints each of them resents a groups of actuators and sensors that allow to perform an optimal control of the rehabilitation. Each joint presents a harmonic drive coupled with a maxon's brushless flat DC motor allowing a gear reduction of $1 : 160$ and an average torque of $35Nm$. In addition, the joints have two types of sensors, potentiometers and force sensors, that allow the control and performance of the different modes of rehabilitation. The potentiometers are placed concentrically to the joint axis so their measurements are used for the localization of the joints positions and the gait control. The force sensors are strain gauges placed on the links of the exoskeleton and designed so that they measure the torque applied by the patient to each joint.

Regarding basic functions of the CPWalker. The set of sensors and actuators presented in the robotic platform allows the implementation of novel therapies that may vary according to the level of disability, increasing the patient participation and motivation during the therapy. These strategies are based on two modes of control: trajectory control and impedance control.

- Trajectory control: This strategy is based on the principle of guiding the lower limbs of the user following fixed trajectories based on the patients height and gait speed. To accomplish this, control uses the referent positions to move the patients limbs according to the potentiometer measurements.

- Impedance control: This strategy is based on the principle of impedance. The objective is allowing an angle deviation from the reference trajectory by inducing a spring-dumper-inertia behavior in the joints, so the patient is no longer guided through the fixed trajectories but need to collaborate.

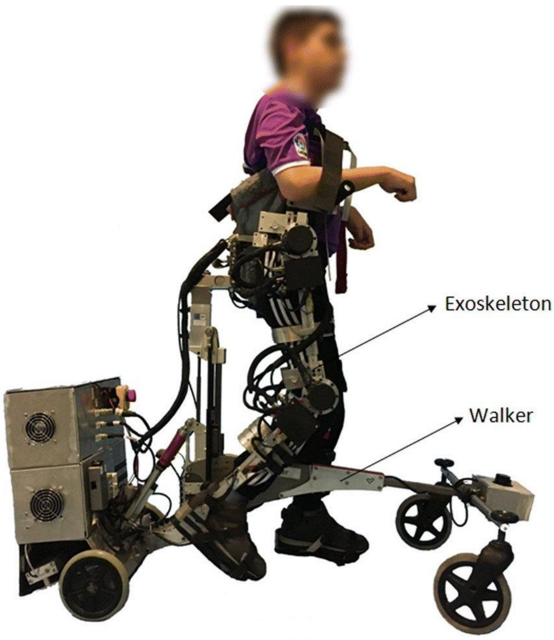


Figure 1.9: CPwalker, exoskeleton and smart walker [9].

1.3. Motivation and objectives

As stated in Section 1.1.1 cerebral palsy is the most common motor disability in childhood with around 2 per 1000 cases in live births. This disability is a non-degenerative but permanent movement disorder. Although, if it is treated from the early stages of life, it is possible to reduce the final effects and improve the quality of life of the patients. This is the objective followed by the CPWalker project, the incorporation of novel robotic therapies in the early treatment of CP patients. At this point is where thus present Masther's Thesis emerges aiming to support the CPWalker project and their future iterations.

The objective of this Master's Thesis is to design, implement and evaluate a modular control system for the CPWalker robotic exoskeleton. The exoskeleton must move along specific trajectories and interact with the patient, so the control must be made up of two control subsystems: trajectory control and impedance control. In addition, the architecture of the system must present high modularity and speed-performance, so it will be based on the "Robot Operating System" (ROS) and the software will be written in the programming language `c++`. To reach the main objective several secondary objectives are defined:

- Design and software development of the communications with the rest of the robotic platform using ROS.
- Design and software development of the hardware communications and data processing to control the exoskeleton.
- Design and software development of the trajectory and impedance control modes.

- Implementation and technical validation of the control system in the four joints of the CPWalker.

1.4. Document Layout

This document describes the several steps performed to carry out the present Master Thesis. The document is organized in the following chapters:

- **Introduction, Chapter 1**: In this chapter it is explained the problem statement a review of the solution and the objectives.
- **Electronic conceptual design, Chapter 2**: In this chapter it is introduced the concept design of the system this Master Thesis has worked with.
- **System Implementation, Chapter 3**: In this chapter it is explained the develop and implemented system in the robotic platform.
- **Thechnical Validation, Chapter 4**: In chapter it is explained the tests performed in the implemented system to validate the system.
- **Conclusions, Chapter 5** In this Chapter are exposed the conclusions and future challenges drawn from this Master Thesis.

Chapter 2

Electronic conceptual design

2.1. Introduction

In this section, a description of the exoskeleton system architecture is provided. The objective is to review the whole system and describe the different modules, mechanical models and control principles that play a relevant roll in this Masther Thesis.

2.2. System architecture

The exoskeleton is made up of several modules that communicate with each other within a network, see Figure 2.3 and 2.4. Each of the four joints of the exoskeleton integrates two sensors and actuators controlled with two small single-board computers (Raspberry Pi 4 Model B [32]) and one external PC. The communication between the actuators, sensors and controllers is accomplish with hardware systems and data buses, and between machines via an internal network based in Robotic Operating System (ROS) [33]. In the following subsections an introduction of ROS and a review of the different connections and networks is given.

2.2.1. Robot Operating System (ROS)

In the last iterations of the CPWalker project, Robot Operating System (ROS)[33] has been implemented as part of the communication system of the robotic platform.

Robot Operating System or ROS is an open source robotic middleware under BSD license developed by Willow Garage ,California EEUU, used to facilitate the development of complex software control systems on different robotic platforms. It is one of the most widely used framework for the development of applications for robotics in the world, providing libraries, tools and a huge community of users to facilitate the development of robotic projects.

ROS is easily implemented in modern languages such as `c++` (roscpp), `Phython` (rospy), although there are also experimental libraries in `Java` or `MATLAB`. It officially supports Ubuntu Linux and experimentally others OS such as Microsoft Windows, OS-X, Debian or Raspbian.

Communications are divided in three layers. A first and application layer composed by the different processes (e.g. nodes), a second layer or middleware layer composed by the libraries and the communication protocols UDPROS and TCPROS based on the standard UDP datagram packets to transport serialized messages and standard TCP/IP sockets for transporting data , finally a third Layer referred to the Operation System as showed in Figure 2.1. It should be noted that although ROS presents low latency and real-time code can be implemented, it is not an RTOS (Real-Time Operating System) [34].

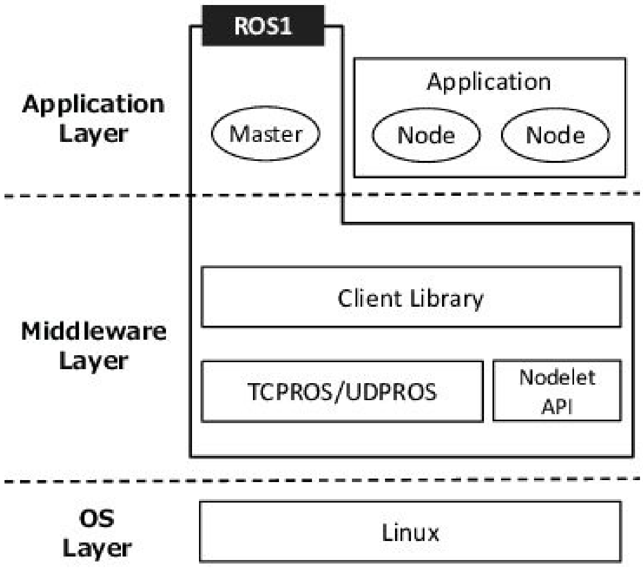


Figure 2.1: Robotic Operating System logo communication layers. Recovered from: [10].

ROS processes are represented as nodes connected by edges called topics in a graph structure [35]. The nodes send and receive messages through these topics or make service calls to other nodes. This network is controlled by the ROS Master process which registers, names the nodes and sets-up the topic and service communications [36].

- **Nodes**: The nodes represent processes in the ROS environment. Several nodes can exist within the same namespace and all nodes must be registered by the ROS Master to have access to topics and services [35].
- **Topics**: Topics are the channels through which the nodes send and receive messages. To exchange information, the nodes publish messages in topics to send data and/or subscribe to topics to receive data. It is an anonymous information exchange system managed by the ROS Master [37].
- **Services**: Services are punctual exchanges of call-response type information, actions with a defined start and end. The nodes call the services or receive the call from them [38].

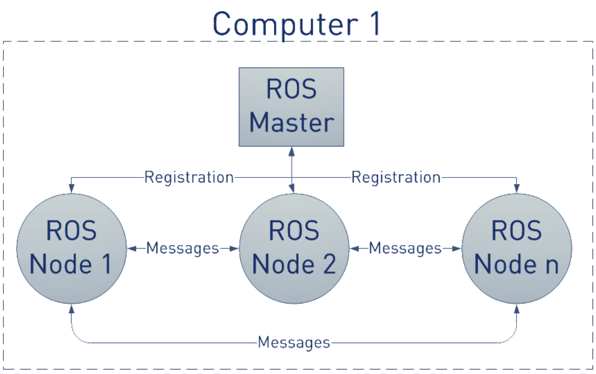


Figure 2.2: Example of a ROS Network.

ROS looks for a distributed design so the nodes do not know where in the network they are running. This means that, when ROS is deployed in various machines/computers, several nodes can be running in different computers/machines and still communicate between each other in the same network.

This approach allows an easy multimachine communication. To obtain said communication several facts must be taken into account:

- ROS must be deployed and running in all the computer/machines.
- There must be bi-directional connectivity between the machines and advertise itself by a name that all other machines can resolve [39].
- Only one master node is needed. And one machine must be selected to run it on and will be the Master machine who will manage the network.
- The rest of the nodes are configured to recognise said master via ROS_MASTER_URI [40].

2.2.2. Communications

The exoskeleton presents three main controller. Two Raspberries Pi 4 Model B called, "Pi Worker" and "Pi Master" integrated inside the exoskeleton that collects and process data and control the exoskeleton, see Figure 2.3 and 2.4. And one external PC used as interface between the physiotherapist and the exoskeleton control. To allow a proper information flow, the architecture of the system is based on several communication systems.

Aforementioned, the communication system may be divided in two differentiated types of networks:

- **Internal Network**: The internal network allows communications between the machines (e.g. Raspberries and portable computer) based on ROS and Internet connections. It uses the two Raspberries as clients of a LAN Network and sets the multi-machine environmental variables to enable ROS communications. Static IPs are defined for the Raspberries in the internal LAN, the Pi

Worker acts as an access point and the Pi Master is configured to command the ROS System. The ROS environment variables ROS_MASTER_URI, ROS_HOSTNAME and ROS_IP are defined in all machines so all can interact with the Master machine. An overall diagram of the internal network is presented in Figure 2.4.

- **Peripheral network**: The preripheral network connects the machines with the rest of sensors and actuators using data-buses and hardware components. The exoskeleton is also composed of electronic components to read, translate information and control actuators such as: sensors (e.g. potentiometers and strain gauges), analog/digital converters (A/D), data acquisition devices (DAQs), motor drivers and motors. In the communication system, the Raspberries perform two clearly different roles: The Raspberry Pi Master communicates with the sensors and the Raspberry Pi Worker controls the actuators. Two types of data-buses are used for the communication between these devices: Serial Peripheral Interface (SPI) for the actuators and Controller Area Network (CAN) for the sensors. See Figure 2.3

As explained in Figure 2.3, the information coming from the sensors is obtained with the Pi Master. This information needs to be sent to the Pi Worker which controls with the motors. To accomplish this task, the messages travel through the internal network and more precisely as ROS messages. Some ROS nodes has being already implemented, the node in charge of receiving and publishing the sensors data is the exo_sensor_acquisition_node. This node obtains the data from the sensors and publishes it through topics to which other nodes can subscribe. The published information is the bits transformed by the DAQ referred to the analog measurements of potentiometers and meters of each one of the four joints.

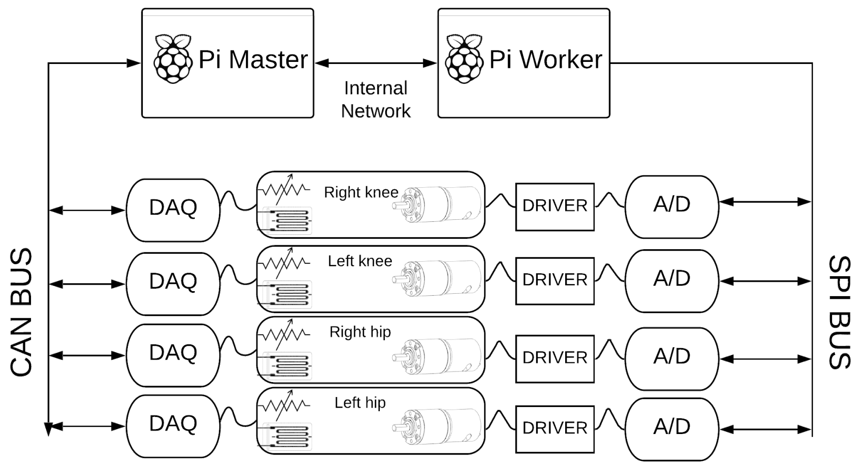


Figure 2.3: Peripheral Network. The Data Acquisition devices (DAQ) translate the gauge and potentiometer analog information to digital data and sent through the CAN bus to the PiMaster. The PiWorker sends digital information through SPI bus to the digital/analog converters (D/A) which translate it to analog information arriving to the servo drivers which controls the motors.

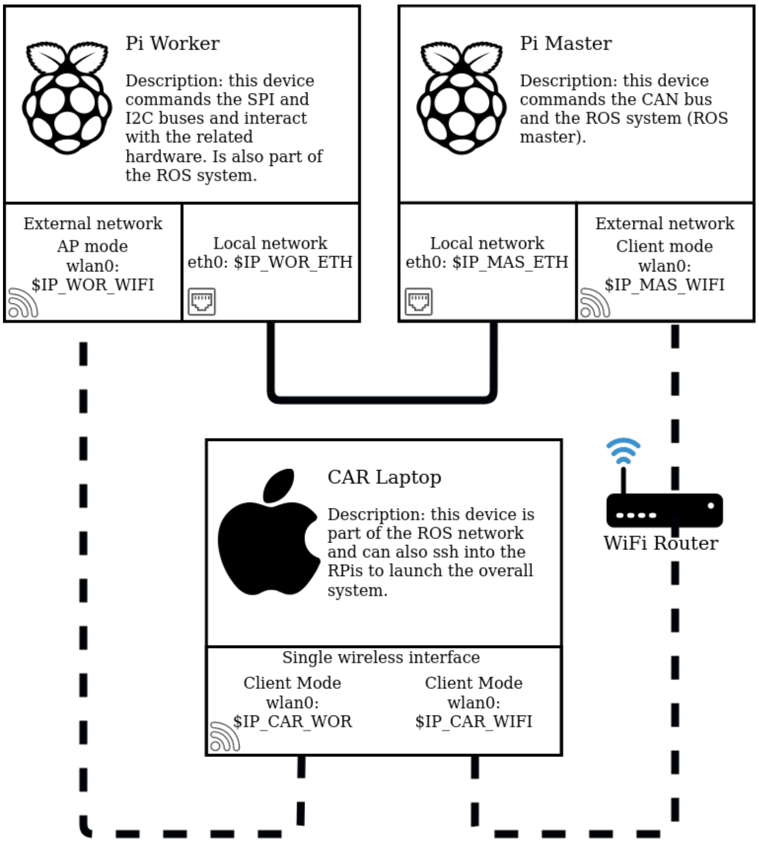


Figure 2.4: Internal Network.The two Raspberries (Pi Worker and Pi Master) and the PC all interconnected via ethernet and LAN. Recovered from [11].

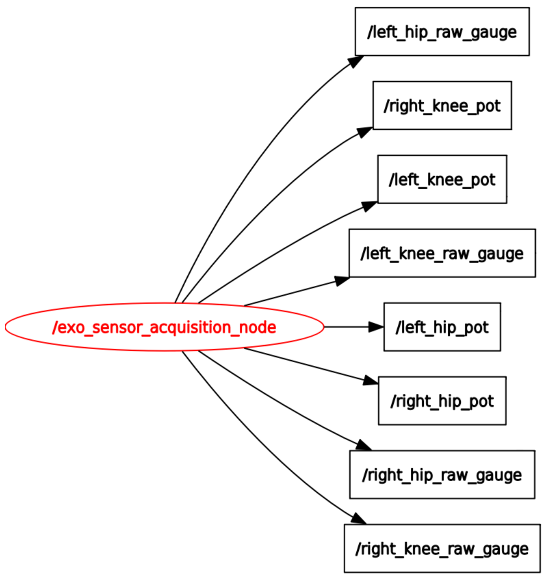


Figure 2.5: The acquisition node receives all the joints sensor data (strain gauges and potentiometers) and publish them in their respected channels (topics) so other topics can use them. (Figure obtained with the ROS tool rqt_graph.)

2.2.3. Data processing

Once the data is received, it presents certain level of noise that need to be filtered before using it to control of the device. An specific ROS node inside the Raspberry Master is in charge of processing the data and publish the new processed information.

The data obtained comes from the potentiometers attached to the axis of the joints measuring the angular position of the joints and the strain gauges located in the metal rods configured in a Wheatstone bridge circuit measuring the torque generated by the human-robot interaction.

The controllers receive two sources of data with noise so to prevent undesired performance of the controllers FIR digital filters are used:

Finite Impulse Respond (FIR): Used for the gauge measurements. FIR digital filters are filters with a finite response duration[41]. The output is represented as a weighted combination of past input values, see Figure 2.6 and Equation 2.1.

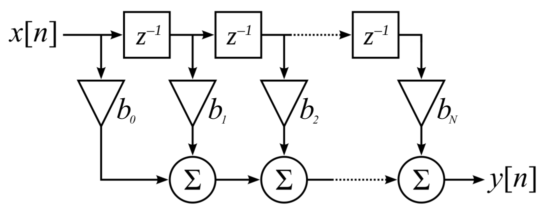


Figure 2.6: Block diagram of a FIR filter. Recovered from [12].

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] \tag{2.1}$$

Where $y[n]$ represents the output, $x[n]$ the inputs, b_k represents the coefficients of the filter and M the number of filter coefficients.

Once the data has been filtered, specific transfer functions are applied so the units of the values change from bits to degrees in case of the potentiometer and to Nm incase of the strain gauges (more details will be explained in the section 3. After applying the filters and transfer functions, the sensed data is published by the node in charge of the data processing so the control node can use them for the control.

2.3. Mechanical model of the exoskeleton

In terms of mechanical structure, the exoskeleton is made up of four links and one hip orthosis, all connected with four joints as exhibited in Figure 2.8. Each of the links presents two leg restraints to ensure the proper movement of the patients' legs, configurable dimensions to adapt to the size of the patients. As mentioned in Section 1.2, all links are made of *Alluminium*7075 [31] resulting in a lightweight and robust

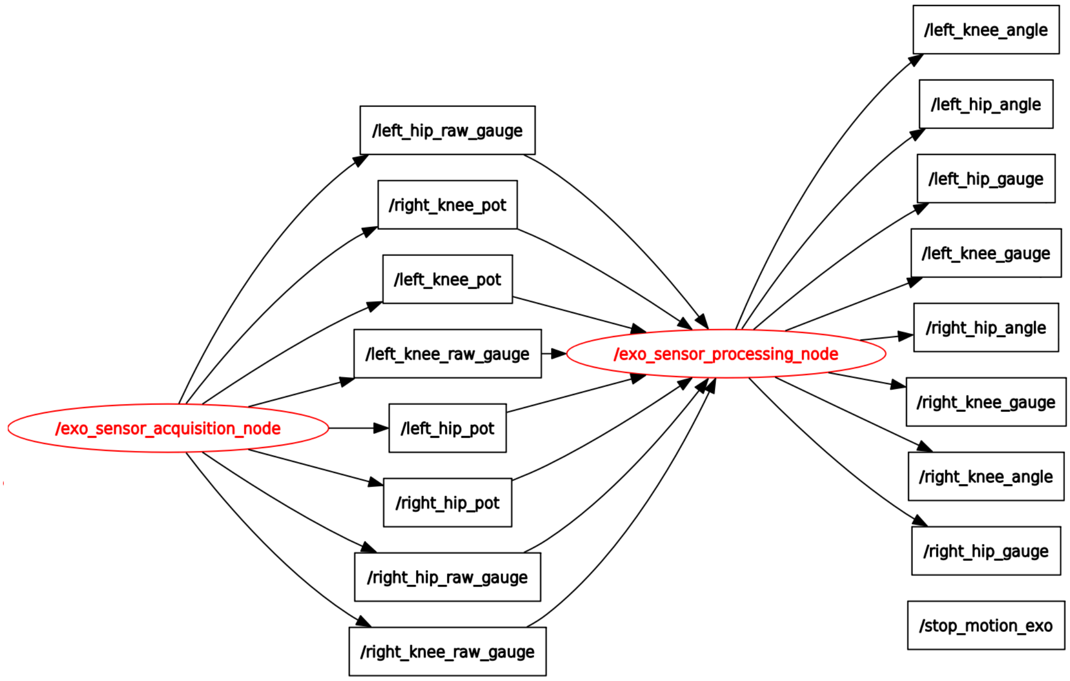


Figure 2.7: The data processing node subscribes to the topics through which the data acquisition node to receive the sensors data, processes and publish it in new topics.(Figure obtained with the ROS tool rqt_graph.)

structure.

According to the mentioned structure and in a similar way as the `Lokomat` approach [42], each of the exoskeleton limbs can be modeled as a double pendulum with distributed masses using the dynamic equation derived by the Lagrange Principle shown in Equation 2.2:

$$\boldsymbol{\tau_{exo}} = \boldsymbol{M}(\theta)\ddot{\theta} + \boldsymbol{c}(\theta, \dot{\theta}) + \boldsymbol{g}(\theta) \tag{2.2}$$

where $\boldsymbol{\tau_{exo}} \in \mathbb{R}^2$ is the torque vector representing the two joints of one exoskeleton's limb, $\boldsymbol{M} \in \mathbb{R}^{2\times2}$ is the symmetric mass matrix representing inertia, the $\boldsymbol{c} \in \mathbb{R}^2$ is the Coriolis term or velocity product term and $\boldsymbol{g} \in \mathbb{R}^2$ is the gravity term representing the force of gravity and other accelerations of other components.

The torque vector can be also expressed as an equation of the torques involved in the exoskeleton dynamics as represented in Equation2.3.

$$\boldsymbol{\tau_{exo}} = \boldsymbol{\tau_{motor}} + \boldsymbol{\tau_{pat}} - \boldsymbol{\tau_{frict}} - \boldsymbol{\tau_{floor}} \tag{2.3}$$

where all
boldsymboltau are vectors in the $\mathbb{R}^2$ space. The $\boldsymbol{\tau_{motor}}$ component represents the torque generated by the motor, including the gear and reduction ratio and joint frictions, $\boldsymbol{\tau_{pat}}$ the torque generated with the patient-robot interaction, $\boldsymbol{\tau_{frict}}$ the static and dynamic friction of the joints and $\boldsymbol{\tau_{floor}}$ the torque generated withe the contact of the patients with the floor while walking.

Regarding the control models of the active components of the structure. The motors used to move the joints of the exoskeleton are the DC brushless motors `maxon`

(a) *Structure.* (b) *Restraints.*

Figure 2.8: Current structure of the exoskeleton and restraints uses to hold the patients legs.

`EC 60 flat Ø68 mm, 100W` model `408057` and a gear reducer `Harmonic Drive model CSD-20-160-2A-GR, Ø20 mm`.

Parameter	**Value**	**Units**
Nominal voltage U_N	24	V
Terminal resistance phase to phase R_m	0.307	Ω
Rotor inductance phase to phase L_m	188×10^{-6}	H
Rotor inertia J_m	121×10^{-6}	kgm^2
Mechanical time constant t_m	13×10^{-3}	s
Electrical time constant t_e	612×10^{-6}	s
Torque constant k_m	53.4×10^{-3}	Nm/A
Back-EMF constant k_b	5.58×10^{-3}	rpm/V

Table 2.1: `maxon EC 60 flat Ø68 mm, 100W` model `408057` characteristics. Recovered from [43].

According to the characteristics mentioned in the Table 2.1, the transfer function of the motor ($G_{\dot{\theta}_m}(s)$) is calculated as followed according its mechanical model as expressed in Equation2.4.

$$G_{\dot{\theta}_m} = \frac{k_m}{(sJ_m + B_m)(sL_m + R_m) + k_b k_m} \tag{2.4}$$

where B_m is the damping constant of the motor calculated as: $B_m = \frac{J_m}{t_m} - \frac{K_b k_m}{R_m}$

Substituting Equation2.4 with the values of Table 2.1 and factorizing the solution, it is obtained the transfer function:

$$G_{\dot{\theta}_m} = \frac{2.32 \times 10^6}{(s+77.18)(s+1605.42)} \tag{2.5}$$

Since the distance from dominant pole of the transfer function $(s + 77.18)$ to the non-dominant pole $(s+1605.42)$ is much higher than the distance from the dominant pole to the origin of the s plane, the effects of this non-dominant pole are considered negligible and can be eliminated obtaining. The module and frequency behaviour of the obtained model is represented in Figure 2.9.

$$\tilde{G}_{\dot{\theta}_m} = \frac{1445.10}{s+77.18} \tag{2.6}$$

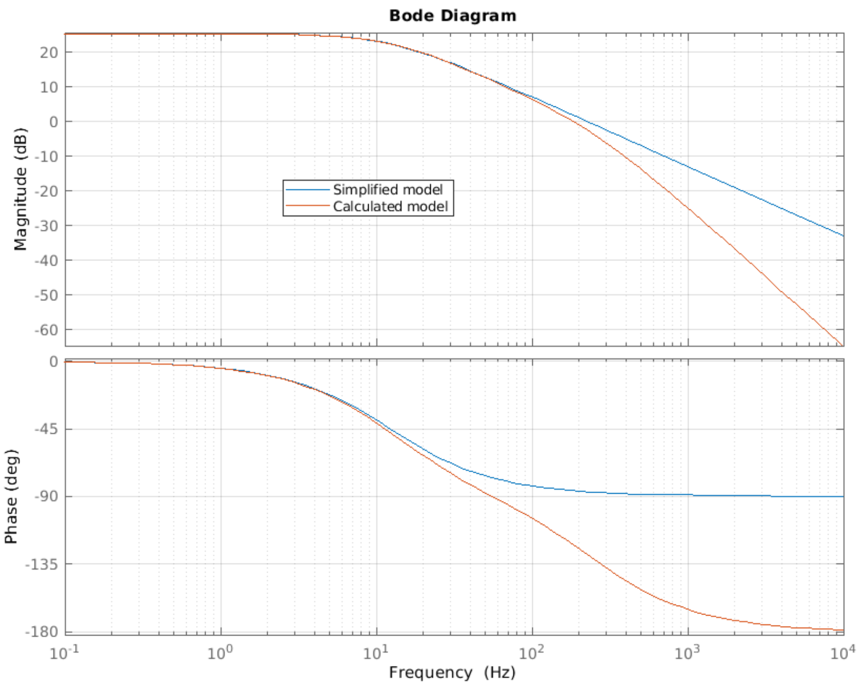
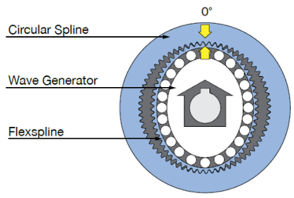


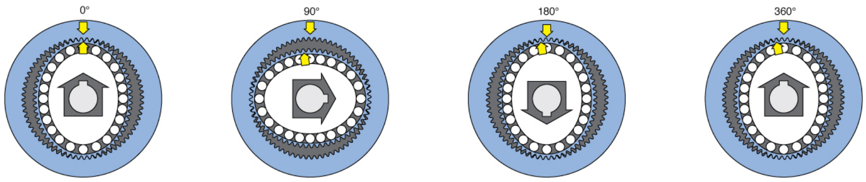
Figure 2.9: Bode representation of the motor frequency response.

Aforementioned, the motors are attached to a Harmonic Drive model `CSG-20-160-2A-GR, Ø20 mm` to increment the output torque allowing the system to move the patient legs, see Figure 2.10. These kind of mechanical gears are known for an almost-zero backlash, high torque, compact size, and excellent positional accuracy due to their operating principle, see Figure 2.10. So, they are commonly implemented in robotic systems [44]. The efficiency (η) of the Harmonic Drives can arrive near to 90% but varies depending on the velocity, [45], so the final input and output torque (τ_{in}, τ_{out}) and velocities ($\dot{\theta}_{in}$, $\dot{\theta}_{out}$) relation of the model are:

$$R = \eta \frac{\dot{\theta}_{in}}{\dot{\theta}_{out}} = \frac{1}{\eta} \frac{\tau_{out}}{\tau_{in}} \tag{2.7}$$



(a) HD Components.



(b) HD operating principle

Figure 2.10: Components and operating principle of a Harmonic Drive gear [13]

Paramenter	**Values**	**Units**
Backlash	$\simeq 0$	-
Ratio (R)	$1 : 160$	-
Starting Torque	3.2	Ncm
Standard Accuracy	1	$arc\,min$
Limit for Average Torque	64	Nm
Limit for Repeated Peak Torque	120	Nm
Limit for Momentary Peak Torque	191	Nm

Table 2.2: `Harmonic Drive model CSD-20-60-2A-GR, Ø20 mm` characteristics.

To ensure a correct response of the system when controlled in the real world, it has been needed a more precise mathematical model of the system including inertia, friction forces and the rest of the components (motor driver, DC motor, motor gear, potentiometer). This has being obtained using the "System Identification" tool of `Matlab` which allows constructing mathematical models of dynamic systems from measured input-output data [46]. Using voltage sent to the system as input-data and the position obtained with the potentiometer as output-data it has been obtained the following transfer function of the hole system:

$$G_{\theta_{exo}}(s) = \frac{9193}{s^3 + 31.53s^2 + 314.5s} \tag{2.8}$$

The close loop system with unitary feedback presents a band width of $3.97Hz$ defined as the frequency when the gain is 3db lower than the static gain, see bode in Figure 2.11.

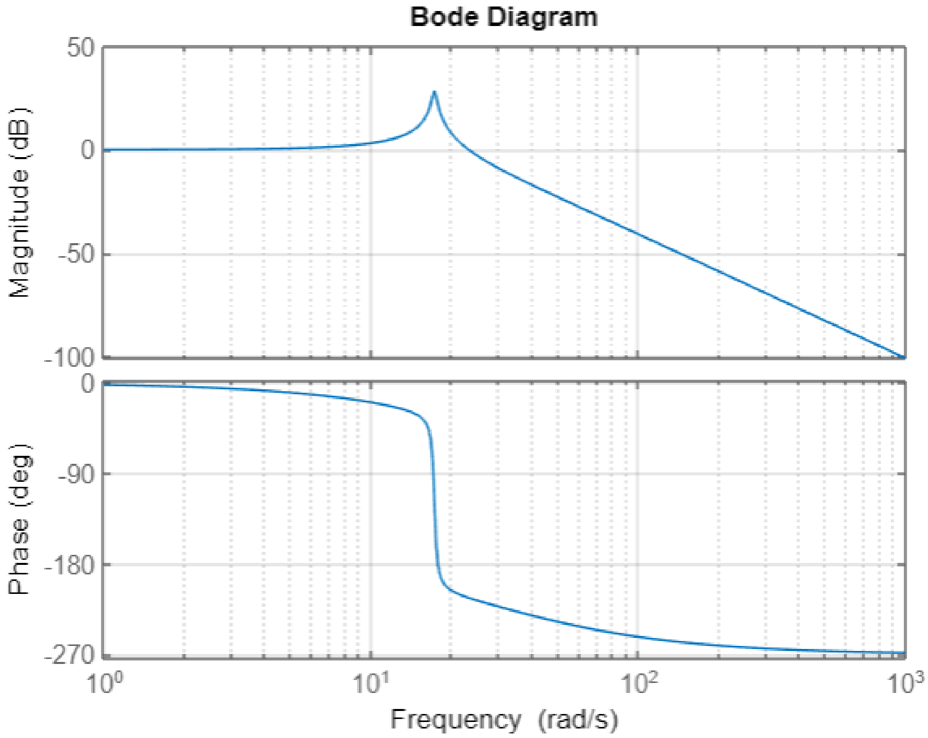


Figure 2.11: Close loop frequency response of the system.

Said transfer function can be rearranged as shown in Equation 2.9 and understood as an integrator (i.e. potentiometer) multiplied by a second order system (i.e. rest of the system) as shown in Equation 2.10.

$$G_{\theta_{exo}}(s) = \frac{1}{s}\frac{9193}{s^2 + 31.53s + 314.5} \tag{2.9}$$

$$G(s) = \frac{Kw_n^2}{s^2 + 2\zeta w_n s + w_n^2} \tag{2.10}$$

The resultant second order system presents a natural frequency (ω_n) of 17.73, a damped frequency (ω_d) of 8.118 and a damping ratio (ζ) of 0.889, with the transient and steady state characteristics of a step response of table 2.3.

Transient state	**Value**	**Units**
Settling time (t_s)	0.2584	s
Overshoot (M_p)	0.2247	%
Steady state	**Value**	**Units**
Static/Position gain(K_p)	29.23	–

Table 2.3: Transient state and steady state characteristic of the second order system of Equation 2.9.

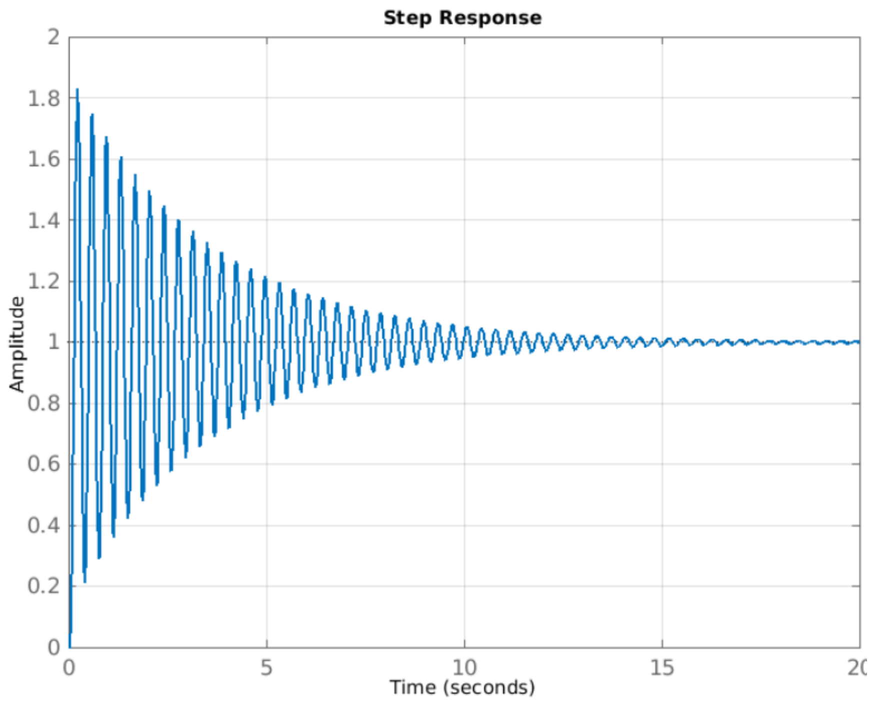


Figure 2.12: Step response of the close loop system. Settling time of 13.1762s and overshoot of 82.8116%.

2.4. Trajectory control approach

As aforementioned in Section 1.2, the control modes of the exoskeleton are trajectory and impedance control. Trajectory control or trajectory tracking is based on the principle of guiding the patient's limbs on fixed reference gait trajectories and has been proven to result in significant improvements in step length, endurance and walking speed in neurologically impaired patients [47].

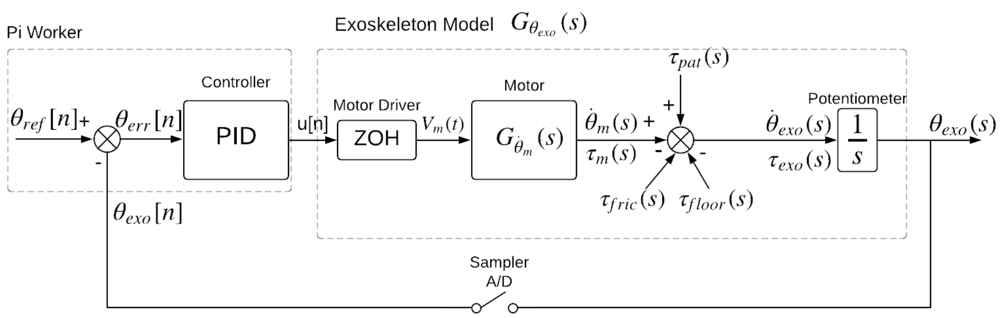


Figure 2.13: Block diagram of the trajectory control system of the exoskeleton.

The block diagram of the trajectory control is showed in Figure 2.13. As observed, the trajectory control, uses the potentiometers information represented as $\boldsymbol{\theta_{exo}} \in \mathbb{R}^4$, to represent the joints angular position. This position is compared with a reference trajectory position of the a reference matrix $\boldsymbol{\theta_{ref}} \in \mathbb{R}^{4\times N}$, see Equation 2.11.

$$\boldsymbol{\theta_{exo}} = \begin{bmatrix} \theta_{right\_knee} \\ \theta_{left\_knee} \\ \theta_{right\_hip} \\ \theta_{left\_hip} \end{bmatrix} \quad ; \boldsymbol{\theta_{ref}} = \begin{bmatrix} \theta_{ref\_right\_knee_1} \cdots \theta_{ref\_right\_knee_N} \\ \theta_{ref\_left\_knee_1} \cdots \theta_{ref\_left\_knee_N} \\ \theta_{ref\_right\_hip_1} \cdots \theta_{ref\_right\_hip_N} \\ \theta_{ref\_left\_hip_1} \cdots \theta_{ref\_left\_hip_N} \end{bmatrix} \tag{2.11}$$

where N represents the number of samples of reference position vectors generated during gait.

This reference matrix is composed of position vectors (matrix columns) that the patient must move along during therapy. It is generated by recording the position vectors ($\boldsymbol{\theta_{exo}}$) of healthy individuals, a common approach in exoskeletons trajectory control systems for rehabilitation therapies [48, 49]. The gait duration was of four seconds (two seconds per step), and the sampled of $10ms$, so the generated matrix is of size $\boldsymbol{\theta_{ref}} \in \mathbb{R}^{4\times 200}$.

By comparing the angular position vector $\boldsymbol{\theta}$ at a certain moment i of the gait with the corresponding sample vector of the reference trajectory matrix $\boldsymbol{\theta_{ref}}$, the angular position error vector $\boldsymbol{\theta_{err}} \in \mathbb{R}^4$ is obtained it represents the angular error of every joints that needs to be corrected by the controller (see Equation 2.12).

$$\boldsymbol{\theta}_{err} = \begin{bmatrix} \theta_{ref\_right\_knee_i} \\ \theta_{ref\_left\_knee_i} \\ \theta_{ref\_right\_hip_i} \\ \theta_{ref\_left\_hip_i} \end{bmatrix} - \begin{bmatrix} \theta_{right\_knee} \\ \theta_{left\_knee} \\ \theta_{right\_hip} \\ \theta_{left\_hip} \end{bmatrix} \tag{2.12}$$

The angular position error $\boldsymbol{\theta}_{err}$ inputs the PID controller implemented in the Pi Worker. The PID generates a control signal for the Motor Drivers which actuates the DC motors, interacting with the exoskeleton mechanical model and generating a final angular speed of the joints of the exoskeleton. The potentiometers obtain the resultant position of the exoskeleton joints and the control loop is re-feed.

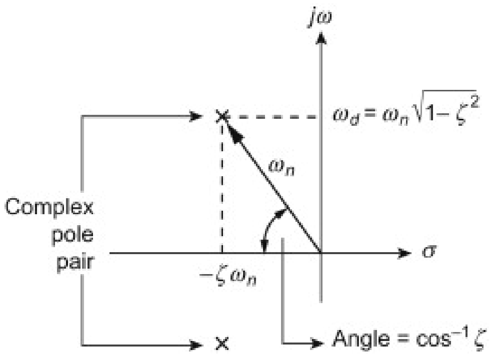
Regarding the PID tuning, several considerations has been taken into account. A first attempt of designing the controller has been done using the root locus method [50]. This method is based on the modification of the poles location in the close loop system by introducing a regulator ($R(s)$) in the open loop. The objective is to modify the root locus of said open loop system so it passes through defined positions (dominant poles) to satisfy the desired transient and steady state requirements of the system.

The therapy can be given at several velocities, the maximum velocity is limited to the four seconds gait duration of the recorded θ_{ref} matrix. According to this, the requirement for transient state of the system are defined to satisfy the fastest possible control velocity (i.e. fastest settling time): ($\boldsymbol{t_s}$) of $10ms$, with an estimated overshoot of 15%. Regarding the steady state requirements, since the transfer function of the system already presents an integrator $\frac{1}{s}$, the position error(e_p) to an step reference is already 0 and no additional steady state requirements are needed. According to this, the designed dominant poles are represented in Equation 2.13:

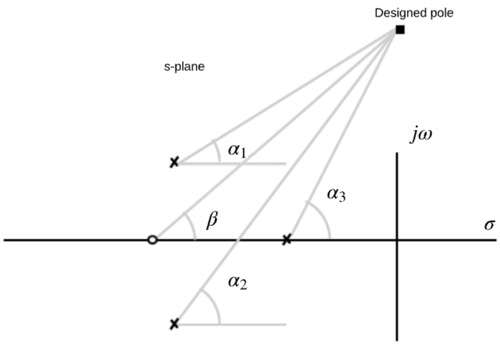
$$t_s = \frac{\pi}{\sigma} \quad ; M_p = e^{-\frac{\pi}{tg(\theta)}} \quad ; \omega_d = \frac{\sigma}{tg(\theta)}$$

$$s_{designed} = \sigma \pm j\omega_b \tag{2.13}$$

$$s_{designed} = -314.15 \pm 511.477j$$



(a) *Defined poles.*



(b) *Argument criteria.*

Figure 2.14: Dominant poles representation in the s-plane.

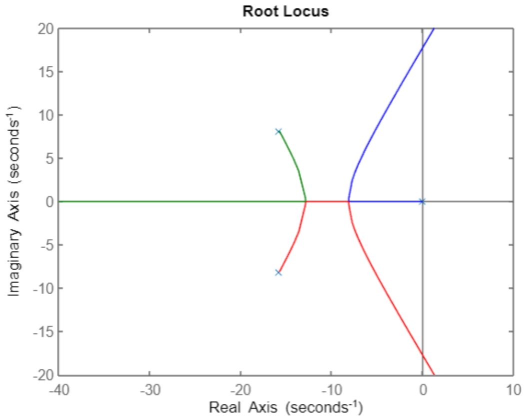
The root locus of the system ($G(s)$) is plotted, see Figure 2.15, and it is observed that it does not pass through the designed poles. According to root locus method [50], in this cases it is needed to introduce a zero in the real axis to modify the root locus.

The location of said zero is determined with the argument criteria, see Equation 2.14 and Figure 2.14:

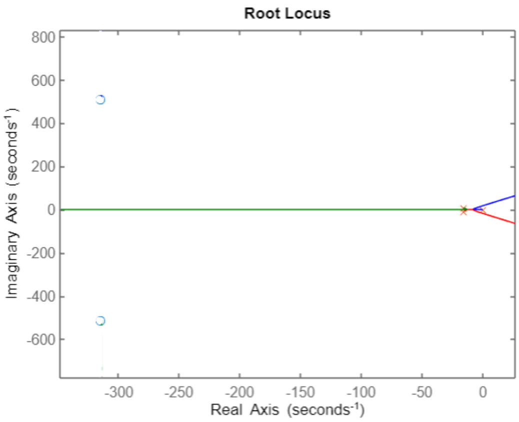
$$(2q+1)\pi = \sum_{i=0}^{N} \alpha_i - \sum_{k}^{M} \beta_k \tag{2.14}$$

where α is the angle from the to the designed pole, β is the angle from the zeros to the designed pole, N is the number of poles of the open loop system, M the number of zeros of the open loop system and q is a vector $q = [0, 1, \ldots, N-M-1]$.

Obtaining an angle of $\beta = 3.18rad$. Since the calculated angle would place the zero higher than the designed poles, meaning it is needed the participation of more than one pole to move the root locus to the desired poles and so the transient state requirements are not satisfied. According to this, a `Simulink` model of the system was generated, see Figure 2.17, and the PID controller was empirically tuned to adjust in the best way to the desired reference trajectory. The final selected controller was a proportional controller with a K_p of 0.17. Obtaining a smooth controlled signal with $200ms$ of dephase, see Figures 2.18 and 2.19.



(a) Root locus of the openloop system.



(b) Root locus including the dominant poles.

Figure 2.15: Root locus of the open loop system. Where the two circles are the designed poles.

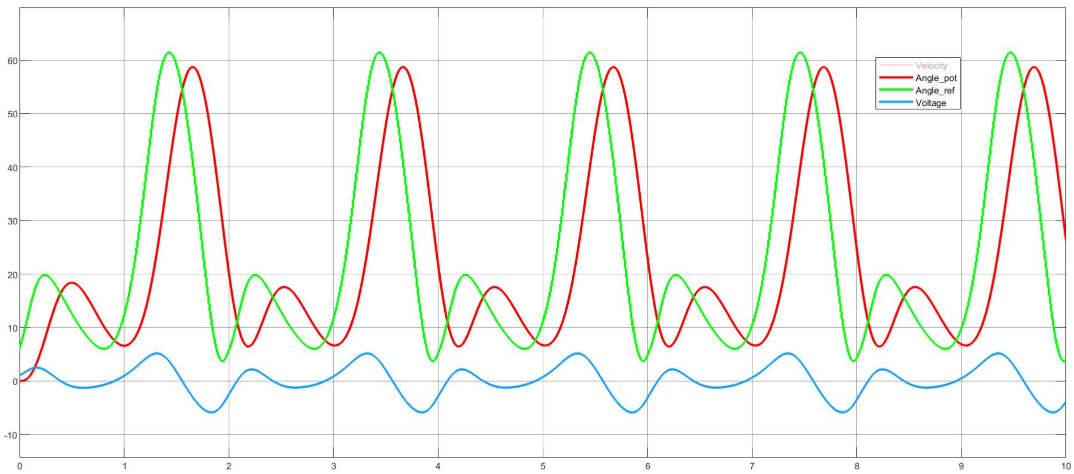


Figure 2.18: Simulation of the trajectory control of the knee joint with a proportional controller of $K_p = 0.17$.

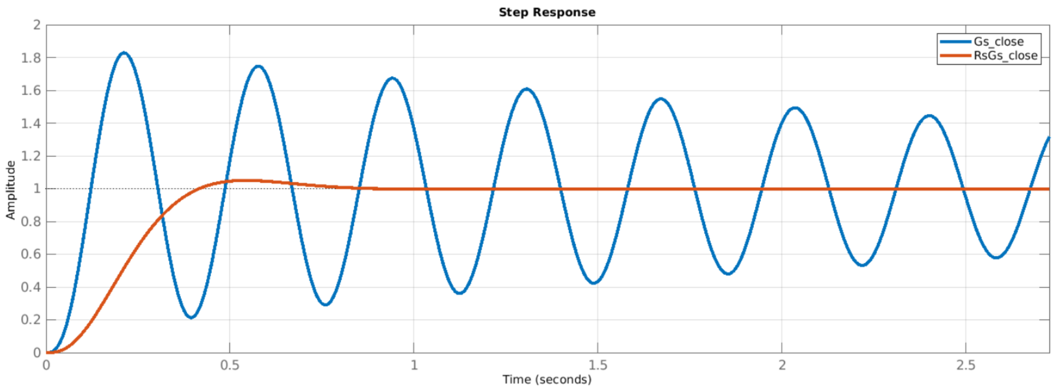


Figure 2.16: Step response of the close loop system when no regulator is used ("Gs_close", blue) and when the regulator is used ("RsGs_close", orange). Settling time of 0.7250s and overshoot of 4.9908%.

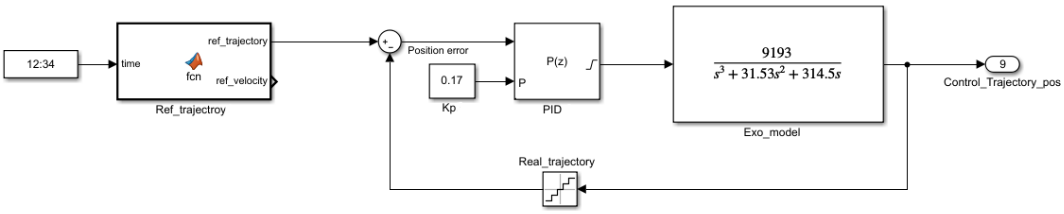


Figure 2.17: Simulink model of the exoskeleton system.

2.5. Impedance control approach

Although the trajectory control systems have been proved to be effective for patients in their rehabilitation [51][52], the objective followed with the impedance control mode is promoting the patient-robot interaction to increase the motivation and participation of the patient during the rehabilitation session. It is proved that an active participation and involvement in the robotic gait rehabilitation is important to develop neuroplasticity, motor control and improve rehabilitation outcomes [47].

The impedance control concept was introduced by Neville Hogan in 1985 to facilitate the application of robots and/or prostheses to tasks involving static and dynamic interactions between the manipulator and its environment. This concept describes the environment as an admittance and the manipulator as an impedance, i.e., the manipulator always impress force in the environment changing its inherent stability, the level of change is controlled with the level of impedance. In other words, it gives a "disturbance response" for deviations from the manipulator desired motion which has the form of an impedance [53].

The impedance can also be understood as a stiffens-dumping-inertia system that interacts with the environment so it can be defined in those three terms. The lowest term is the stiffness, it represents the static relation between the output force and

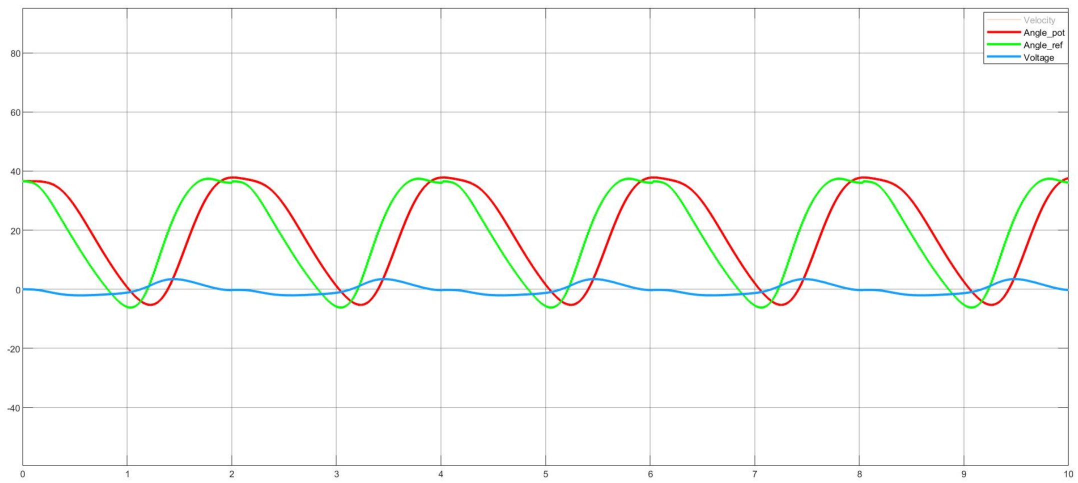


Figure 2.19: Simulation of the trajectory control of the hip joint with a proportional controller of $K_p = 0.17$.

input displacement [53], represented by Equation2.15. The second term is related with the dumping which describes the relation between force and velocity [54], represented by Equation2.16. And the last term deals with the inertia effects of the interaction, represented by Equation2.17.

$$\boldsymbol{F}_{int}(t) = K[\boldsymbol{X}_0(t) - \boldsymbol{X}(t)] \tag{2.15}$$

$$\boldsymbol{F}_{int}(t) = B[\boldsymbol{V}_0(t) - \boldsymbol{V}(t)] \tag{2.16}$$

$$\boldsymbol{F}_{int}(t) = \boldsymbol{M}\frac{d\boldsymbol{V(t)}}{dt} \tag{2.17}$$

where $\boldsymbol{F}_{int}$ represents the end-point force, K the stiffness constant, $\boldsymbol{X}_0$ the desired end point position and $\boldsymbol{X}$ the real position, B the dumping constant or viscosity, $\boldsymbol{V}_0$ the desired end point velocity and $\boldsymbol{V}$ the real velocity and $\boldsymbol{M}$ represents the inertia tensor of a rigid body.

Summarizing, the non linear feedback law for impedance control is given by the Equations 2.18 and 2.19 [54]:

$$\boldsymbol{F}_{int}(t) = K[\boldsymbol{X}_0(t) - \boldsymbol{X}(t)] + B[\boldsymbol{V}_0(t) - \boldsymbol{V}(t)] + \boldsymbol{M}\frac{d\boldsymbol{V(t)}}{dt} \tag{2.18}$$

$$Z(s) = \frac{F(s)}{X(s)} = Is^2 + Bs + K \tag{2.19}$$

In the present work, the exoskeleton represents the manipulator and the environment is represented as the patient-robot forces and the rest involved forces in the exoskeleton dynamics, see Equation 2.3. Aforementioned, the level of impedance can be modified so the interaction forces between the manipulator and the environment can vary. This approach is used to set different levels of impedance, i.e., different levels if intensity in the therapy. High impedance levels implies the manipulator (i.e., exoskeleton) impress more force to the environment (i.e., the patient). Low impedance levels implies the patient is in charge of the exoskeleton motion.

Inspired in the `Lokomat` approach, the impedance control is accomplished using a cascaded impedance and force control system using the position and the torque values obtained by the potentiometers and the strain gauges, see Figure 2.22.

The external impedance loop generates the desired impedance according to the K, B and $\boldsymbol{M}$ selected values and the $\boldsymbol{\theta_{err}}$ respect the reference trajectory; in case the environment does not generate any forces $\boldsymbol{\tau_{exo}} = 0$, the controller behaves as a trajectory controller. The internal force control loop (PID) tracks the relation between the interaction torques of the exoskeleton and the desired impedance torque calculated; in the case that $\boldsymbol{\theta_{error}} = 0$, it tracks the environment forces $\boldsymbol{\tau_{exo}}$ (i.e., the patient forces) with zero set point. The controller can be understood as a physical equation of torque interaction where the impedance generator produces the desired torque ($\boldsymbol{\tau_{impedance}}$) and the torque tracking generates the resultant torque according to the torque applied by the patient ($\boldsymbol{\tau_{exo}}$), where $\boldsymbol{\tau_{resultant}}$ is the final torque applied to the system.

$$\boldsymbol{\tau_{resultant}} = \boldsymbol{\tau_{impedance}} - \boldsymbol{\tau_{exo}} \tag{2.20}$$

Regarding the controller tuning of the system, see Figure 2.22, three different levels of impedance have being selected (high, medium and low) to allow an evolution in the rehabilitation therapies of the patients. High impedance means high torques are applied to the patients so they follow the trajectory (more similar to a trajectory control approach) and low impedance means low torques are applied to the patients so they are in charge of the movement. The values of the variables of the controllers have been empirically selected, taking into account that all (the impedance generator variables and the force tracking variables) influence in the final applied torque ($\boldsymbol{\tau_{resultant}}$). The final selected values has been the following:

- **High impedance:**
  - Impedance generator: $K = 2$, $B = 0.05$ and $\boldsymbol{M} = 0$.
  - PID Force taking: $Kp = 0.085$
- **Medium impedance:**
  - Impedance generator: $K = 1.4$, $B = 0.025$ and $\boldsymbol{M} = 0$.
  - PID Force taking: $Kp = 0$.
- **Low impedance:**
  - Impedance generator: $K = 1$, $B = 0$ and $\boldsymbol{M} = 0$.
  - PID Force taking: $Kp = 0.17$

For the validation of the simulated control, the three levels of impedance were implemented in the `Simulink` model using as input torque ($\boldsymbol{\tau_{exo}}$) a real measured torque signal obtained with the gauge sensors. The simulation results are shown in Figure 2.20 and Figure 2.21, which shows that, depending on the impedance level, the

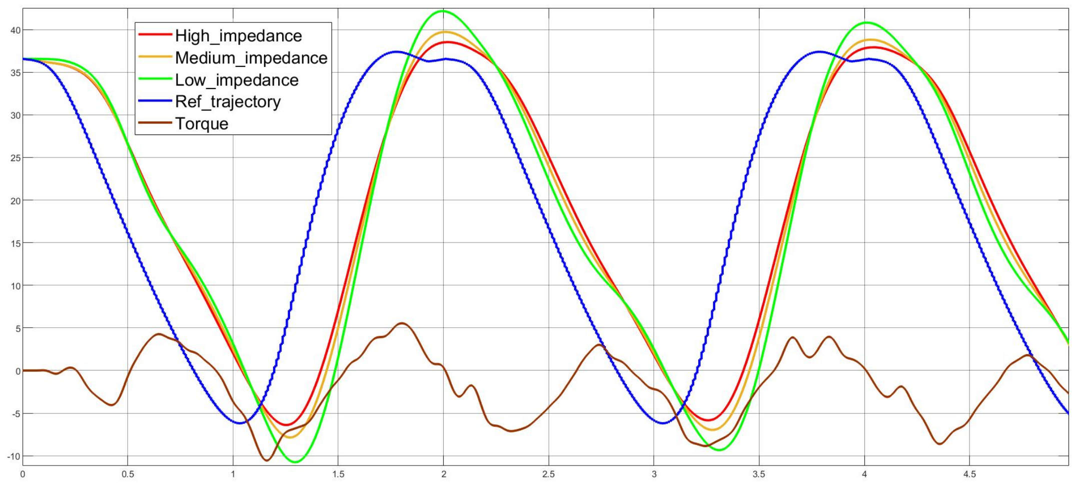


Figure 2.20: Simulation of the impedance control of the hip joint of the exoskeleton.

real trajectory of the exoskeleton is partially modified according to the toque applied by the patient-robot interaction.

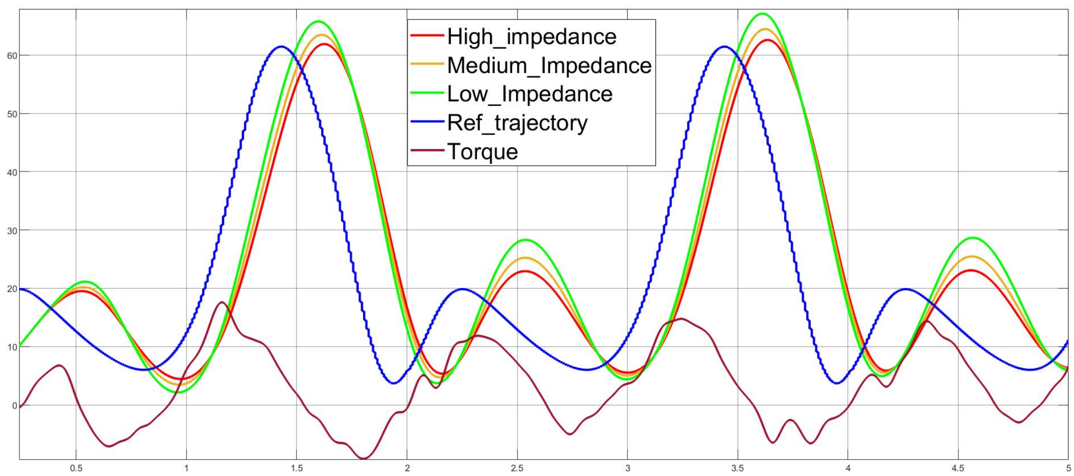


Figure 2.21: Simulation of the impedance control of the knee joint of the exoskeleton.

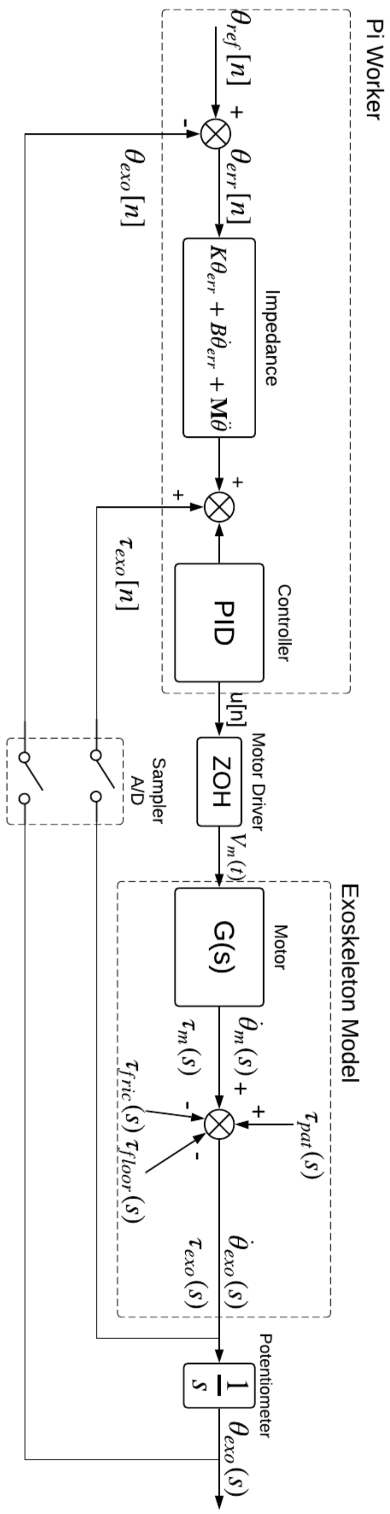
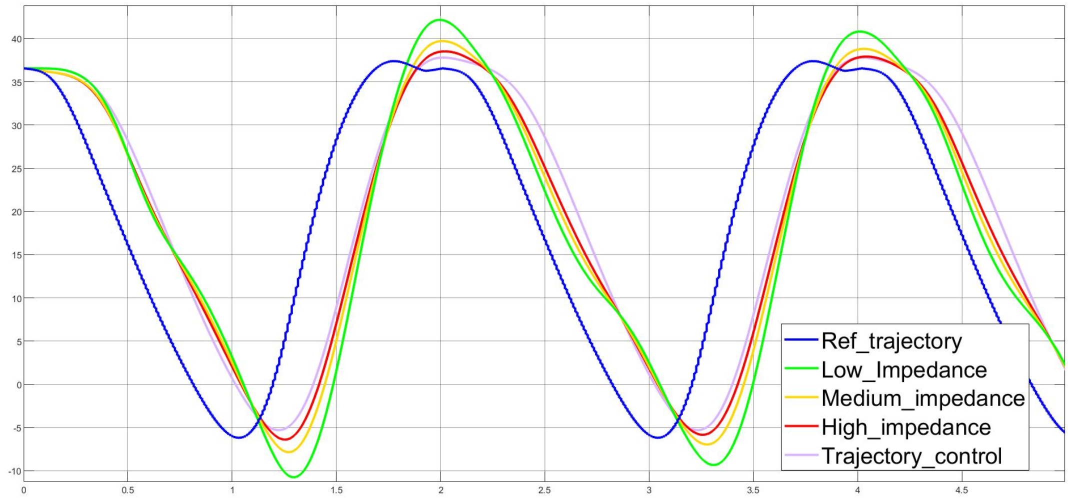


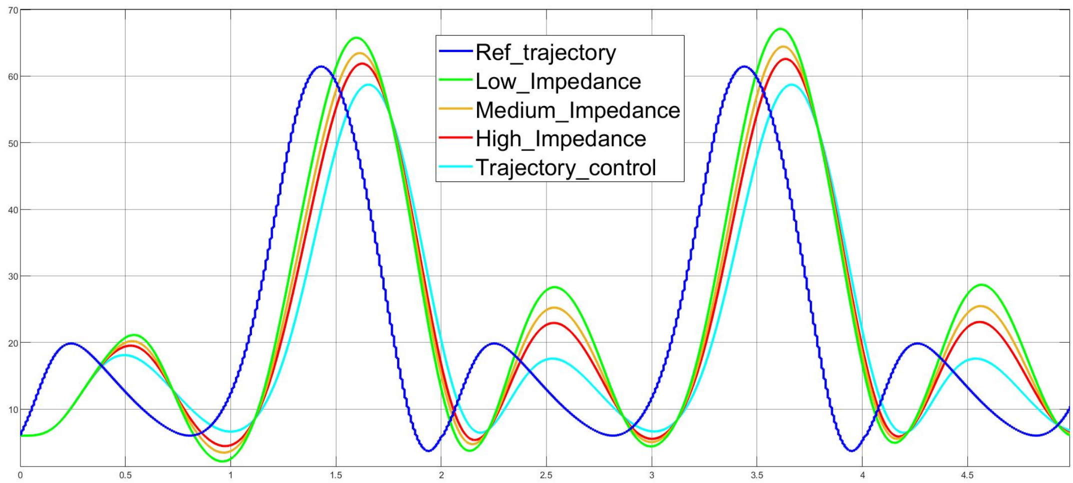
Figure 2.22: Block diagram of the impedance control system.

2.6. Conclusions

This chapter has explained exoskeleton control design including the communication systems, the data processing methods, structure and actuators models and the control principles and methods used in the exoskeleton tested under a simulation, see results in Figure 2.23.



(a) Trajectory control and Impedance control of the hip joint.



(b) Trajectory control and Impedance control of the knee joint.

Figure 2.23: Simulation of the control systems (trajectory control and impedance control) in the *(a)*hip and *(b)* knee.

Chapter 3

System implementation

3.1. Introduction

As aforementioned, the system is composed of two raspberries and one computer. The Raspberry Master (PiMaster) runs the acquisition and processing ROS nodes collecting and processing the information. The Raspberry Worker(PiWorker) controls the motors and the trajectory and impedance control modes.

The present work implementation encompasses modifications in the processing ROS node to filter the torque data obtained with the strain gauges and the development of a new ROS node (control node) which includes the communication with the actuators, with the other ROS nodes, and contains the control algorithms.

This chapter is divided in two parts, "low-level control" which encompasses the low-level communications with the motor, data filtering and the development of a new ROS node and a "high-level control" which includes the algorithms used for the control of the exoskeleton.

3.2. Low-level control

In this section it is explained the implemented actuator low-level control of the system. Encompasses the communication with the digital-analogue converters that inputs the motor drivers and control the motors movements, as well as the data processing of the gauge measurements.

3.2.1. Actuator control

The actuators are controlled with an analog motor driver, `AZBH12A8 PWM servo driver` of ADVANCED motion control [55], controlled in "duty cycle mode" which means: the internal duty cycle (which controls the motor supply current) is directly proportional to an input command of range $[-10V\ ,\ 10V]$ [56]. Since the Raspberry Pi output voltage is limited to $5V$ this control is accomplish using the Digital Analog Converter `AD5570` of Analog Devices [57].

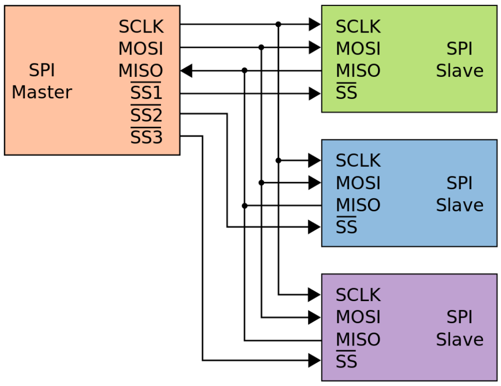


Figure 3.1: Typical SPI bus: master and three independent slaves. Recovered from [14].

The communications with the converter are done via the Serial Peripheral Communication bus (SPI), [58]. The SPI bus communication is a synchronous serial communication used in short-distance communication primarily in embedded systems. It uses a slave-master communications architecture with a single master wired to several slaves so it can ask for information of a specific slave, see Figure 3.1. Several channels are used for the communication:

- **SCLK**: Serial Clock (output from master)
- **MOSI**: Master Output Slave Input, or Master Out Slave In (data output from master)
- **MISO**: Master Input Slave Output, or Master In Slave Out (data output from slave)
- **SS**: Slave Select (often active low, output from master)

The SPI is implemented in the "Independent Slave Configuration" [58], with four Slave Select (SS) channels, one per joint. Since the Raspberry Pi does not have four SS pins it has been used the `bcm2835` library version 1.62 [59], which allows the user to configure the SPI bus to use free GPIO pins as SS pins. The selected PINs and GPIOs of the Raspberry Pi for the SPI communication are listed in Table 3.1.

According to the `AD5570` requirements, it needs to receive $16bits$ data to generate the desired voltage, see `AD5570` transfer function of Equation 3.1. This information is transmitted once per control cycle to the four converters, a total of $4 \times 16bits = 64bits$ per control cycle.

$$V_{out} = 30215.3 + D \times 3364.2 \tag{3.1}$$

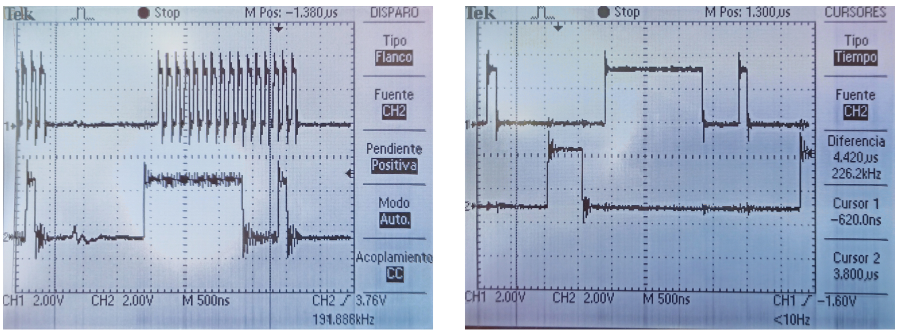
where D represents the decimal equivalent of the code

The system works at a frequency of $500Hz$, so the SPI has been configured to send $16bits$ per $3.74\mu s \rightarrow 267.4kHz$, a total of $14.96\mu s \rightarrow 66.84kHz$ for the hole 64bits, to avoid any latency problems.

PiWorker		
SPI	**GPIO**	**PIN**
MISO	19	35
MOSI	20	38
SCLK	21	40
SS_right_knee	5	29
SS_left_knee	6	31
SS_right_hip	16	36
SS_left_hip	26	37

Table 3.1: GPIO and PIN configuration of the SPI.

The design of the SPI communication software has been done according to the "Standalone" timing characteristics of the `AD5570` obtaining the following satisfactory results, see Figure 3.2.



(a)SCLK up, MOSI down. *(b)MOSI up, SS down.*

Figure 3.2: Raspberry Pi SPI test plotted in an oscilloscope.

3.2.2. Data acquisition and processing

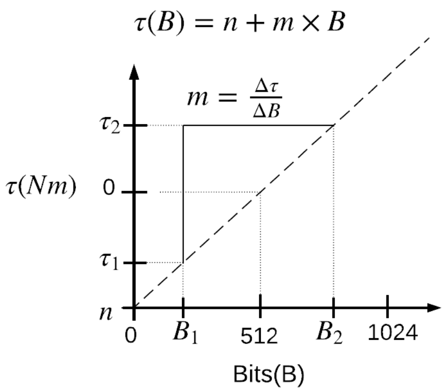
The Raspberry Pi Worker is in charge of receiving the sensor messages obtained and processed with the Raspberry Pi Master to use them for the exoskeleton control.

The exoskeleton uses gauge sensors to receive information about the torque experienced in its joints. These sensors were already implemented in an analog electronic circuit with a four active strain gauges mounted in a Wheatstone bridge configuration which increases sensibility and reduces temperature noise. The mentioned electronic montage transforms deformation of the exoskeleton links to voltage. This voltage is read by a 1024*bits* Data Acquisition module (DAQ) and converted into digital information obtained by the Pi Master via the CAN bus.

A calibration and linearization of the gauge readings is needed to obtain the *bit*-torque (Nm) relation, see Figure 3.3. The procedure for the calibration is the following:

- When no torque is applied to the joints, the offset of the analog electronic circuit amplifier is tuned to obtain a logic zero ($512 bits$) in the DAQs output.

- The transfer function is obtained placing several weights at a certain distance from the joints axis generating different torques (τ_1,τ_2), taking note of the DAQ out values (D_1, D_2) and linerizing the measurements.



$$\tau = -155.488 + 0.30333 \times B$$

Figure 3.3: Linearisation of the bit-torque gauge measurements.

Regarding the gauge data processing. As observed in *(a)* and *(b)* of Figure 3.5, the gauge measurements present low amplitude noise in all frequencies which results in unstable control of the robot. By analysing the frequency domain of the recorded signals it is concluded that the frequencies of interest where between $0Hz$ and $10Hz$ so a low pass filter of cut frequency equal to $10Hz$ was implemented.

The mentioned low-pass filter was designed with the form of a digital Finite Impulse Response filter (FIR) since these kind of filters present stable responses and are easy to implement. The FIR filter was implemented using the window method with a Hann window of order 100. This window was selected because it presents an accurate cut frequency and high attenuation ($-100dB$) in frequencies higher than $77Hz$, see *(c)* and *(d)* of Figure 3.4 and Figure 3.5.

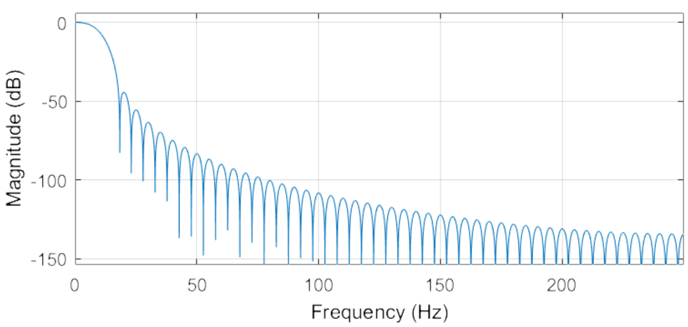
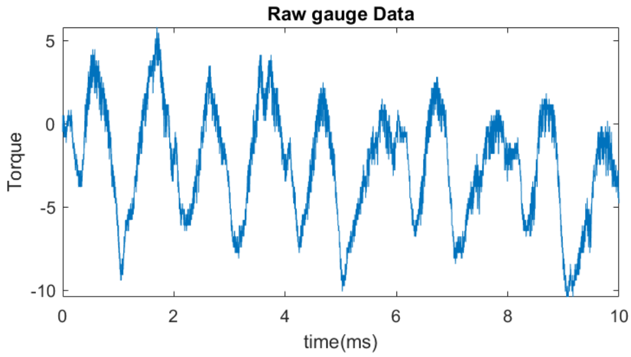
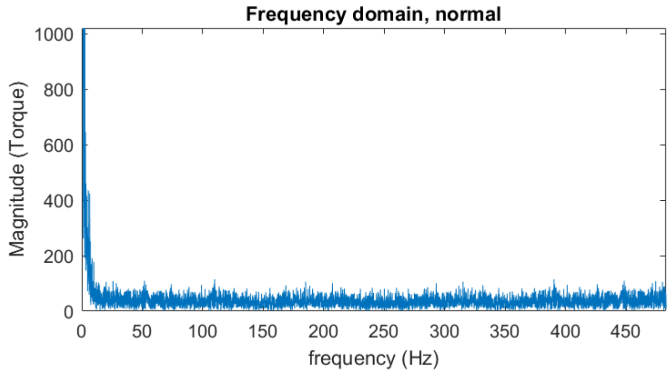


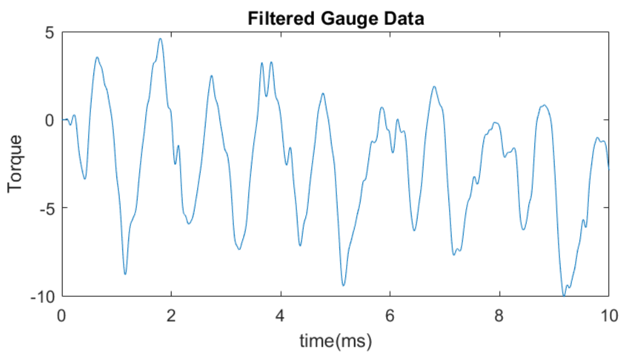
Figure 3.4: Hann filter with cut frequency of $10Hz$ and order 100.



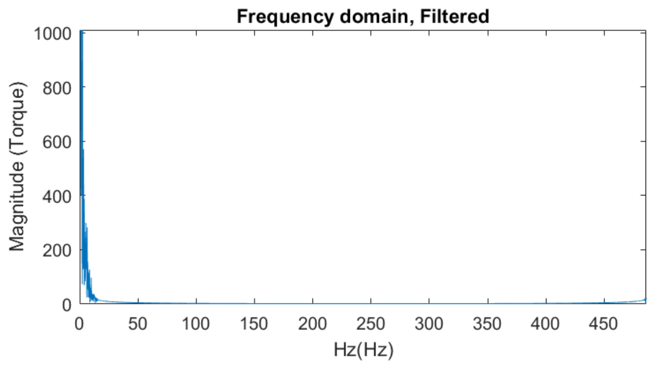
(a) Time domain raw measurements.



(b) Frequency raw measurements.



(c) Filtered torque measurements represented in the time domain.



(d) Filtered torque measurements represented in the frequency domain.

Figure 3.5: Raw and filtered data of the torque measurements obtained with the gauge senors of the system.

3.3. High-level control architecture

This section explains the architecture of the high-level software implemented in the exoskeleton.

The software runs in the Raspberry Pi worker and has been developed based on ROS and using `c++` programming language. The main control loop of the program runs inside a new node called "control node", see Figure 3.6. The node automatically subscribes to topics according to the predefined hardware configuration listed in a `.jaml` file, to a "stop_motion_exo" topic representing a panic button that stops the motion on the exoskeleton and obtains the trajectory reference matrix ($\boldsymbol{\theta_{ref}}$) from other `.jaml` as ROS parameters so it can be used by the controllers.

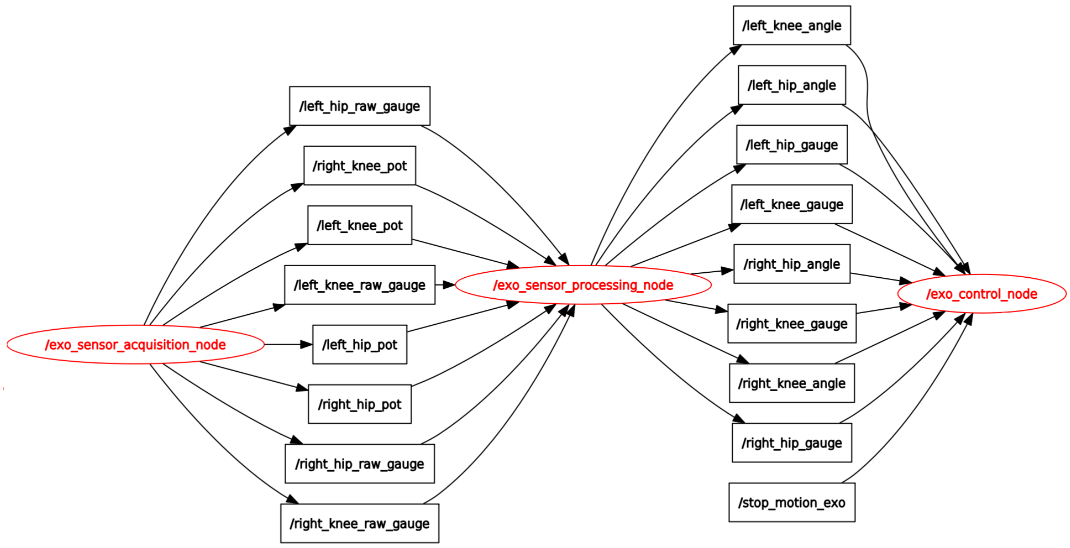


Figure 3.6: ROS communications between the three nodes preset in the exoskeleton. (Figure obtained with the ROS tool rqt_graph.)

The node uses three classes to perform the control, see Figure 3.9:

- **JointController**: This class represents a controller of a joint. The control node generates as JointController objects as available joints in the system. The JointController objects contain the designed PIDs and Impedance generators for the different modes of control.

- **SPI**: This class represents an SPI communication bus of the Raspberry. Since the Raspberry Pi only uses one SPI port for the communication, the control node generates just one SPI object which contains the aforementioned designed SPI bus.

- **PID**: This class represents a PID discrete implementation. Four PID objects are generated with each JointController object (one for the trajectory control and three for the impedance control levels).

When the node is executed, it checks if the ROS network and communication with the ROS Master are available. Then it subscribes to defined topics according to the hardware available listed in a hardware configuration `.jaml` file and then enters in the ROS loop. In each loop cycle, the node first checks for the network status and then performs the desired control. The loop runs according to the selected ROS loop characteristics at a sampling frequency of $500Hz$ (see Figure 3.7).

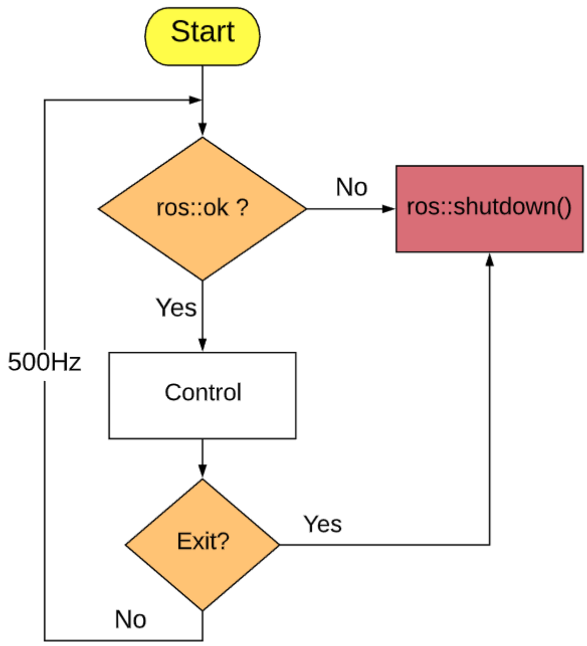


Figure 3.7: ROS loop running the control node of the exoskeleton.

In Figure 3.8, the flow of the program is represented. The program first checks if the motors are stopped to avoid hazard situations. Then it asks to the user to start a new the therapy. When the therapy starts the program, it checks if the joints are in their start positions. In case they are not it moves them till they reach the starting points. Then, the duration of the step, number of steps, and the type of control mode are asked. In case the impedance control is selected, the level of impedance is also asked. Once obtained this information the program ask to begin the therapy and generates the desired control. The program finishes anytime the boolean messages of the "stop_motion_exo" topic are "true" or at the end of a each therapy session when the user decides to exit.

Small adjustments to the controllers values have been done to improve the exoskeleton performance in real world environment. This adjustments have been done under empirical tests performed in the right knee of the exoskeleton. The final control values are the following:

- **Trajectory control:**
  - $K_p = 0.14, K_i = 0,\ K_d = 0$
- **High impedance:**

  - Impedance generator: $K = 2$, $B = 0.05$ and $\mathbf{M} = 0$.
  - PID Force taking: $Kp = 0.07$

- **Medium impedance:**
  - Impedance generator: $K = 1.4$, $B = 0.03$ and $\mathbf{M} = 0$.
  - PID Force taking: $Kp = 0.1$, $K_d = 1.5$,

- **Low impedance:**
  - Impedance generator: $K = 0.7$, $B = 0.01$ and $\mathbf{M} = 0$.
  - PID Force taking: $Kp = 0.14$

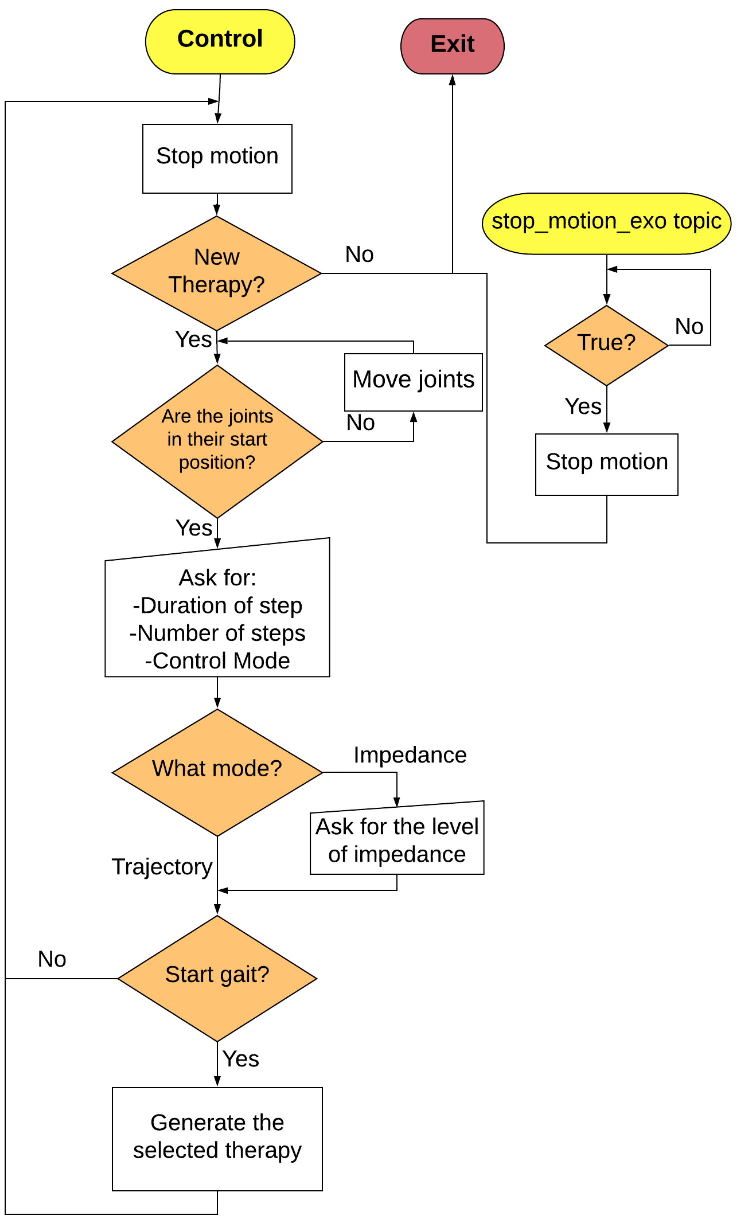


Figure 3.8: Organigram of the exoskeleton normal operation.

3.4. Conclusions

This chapter has explained the implementation of the concept design, explained in Chapter 2, in the real exoskeleton including the low-level communications, data filtering and software architecture.

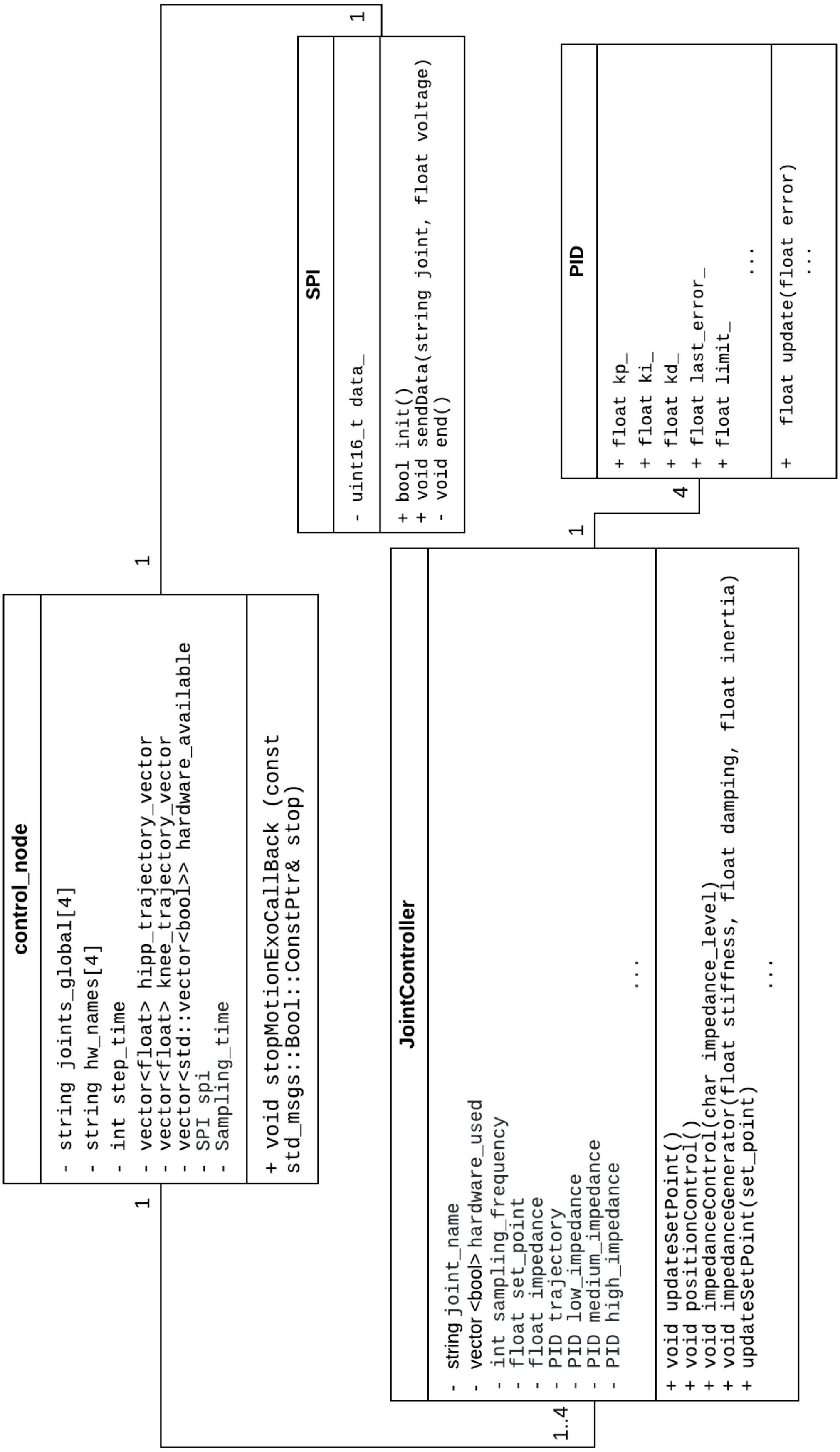


Figure 3.9: UML of the software system.

Chapter 4

Technical Validation

4.1. Introduction

In this section the several tests performed to validate the system are explained. The objective of the validation is to ensure that each control mode performs in the desired way, i.e., to obtain a system with four differentiated modes of control. The desired system performance is that the joints always follow the pattern of the reference trajectory, deviating to some degree according to the control mode and torque applied to the joint. These deviations will be greater when the low impedance control is used (i.e. user in charge) and smaller when the high impedance or the trajectory control is used.

The tests consists in the application of torques in different magnitudes and directions during therapy to test the performance of the exoskeleton under different scenarios. The designed scenarios are:

- **No torque**: This test simulates the scenario in which no torque is applied to the exoskeleton (i.e. no one is using the exoskeleton).
- **Passive Torque**: This test simulates the scenario in which the patient leaves the legs soft during therapy.
- **Favorable Torque**: This test simulates the scenario in which the patient applies a high torque favorable to the desired movement.
- **Opposite Torque**: This test simulates the scenario in which the patient applies a high torque opposite to the desired movement.

Each of these tests is applied during the four control modes, measuring the angle reference ($\boldsymbol{\theta_{ref}}$) and the angle measured with the potentiometer ($\boldsymbol{\theta}$). The obtained data is plotted, compared and analyzed.

The comparison is done by calculating the Range Of Motion (ROM). The Range Of Motion is a term that represents the linear or angular distance that a moving object may normally travel while properly attached to another [60]. In this case, the moving object of study is the segment from the knee to the ankle, and the angular distance is the distance the knee must move to match the reference trajectory (θ_{ref}).

The technical validation has been carried out in the right knee of the exoskeleton. The knee kinematics function can be understood as a time function with two relative and absolute minimums and maximums repeated periodically as shown in Figure 4.2 and 4.1.The ROM calculation consists of the comparison between the angular path of the reference trajectory (that is, increments between a maximum and minimum value) and the real trajectory generated with each control mode.

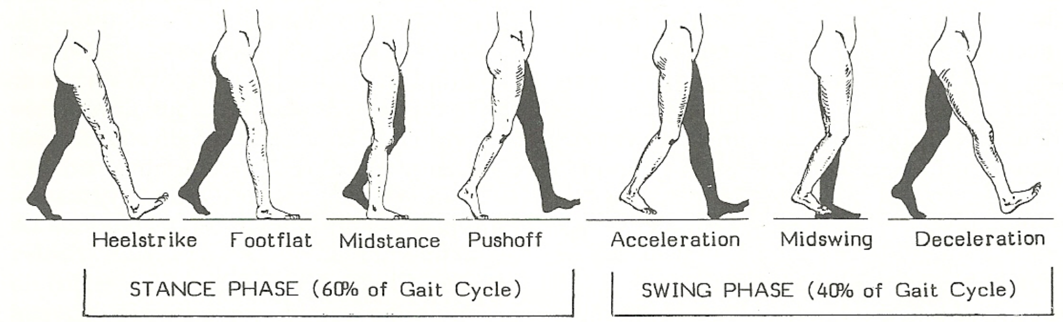


Figure 4.1: Bipedal gait cycle [15]

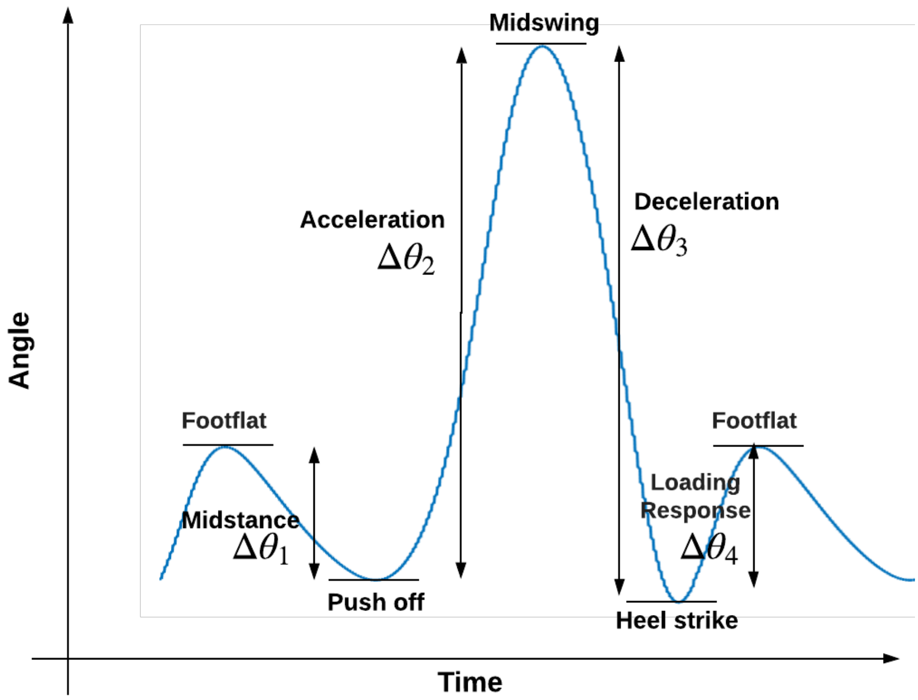


Figure 4.2: Knee kinematics relation with bipedal gait cycle

Since the knee motion consists on two flexions (acceleration and loading response) and two extensions (midstance and deceleration), the ROM is divided in two flexion movements ROM_A and ROM_L) and two extension movements (ROM_D and ROM_M).

Calculated as the percentage of gait accomplished in comparison with the desired gait pattern (θ_{ref}):

$$ROM_M(\%) = \frac{\sum_{n=0}^{M} \Delta\theta_1(n)}{\sum_{n=0}^{M} \Delta\theta_{1\ ref}(n)} \times 100 \tag{4.1}$$

$$ROM_A(\%) = \frac{\sum_{n=0}^{M} \Delta\theta_2(n)}{\sum_{n=0}^{M} \Delta\theta_{2\ ref}(n)} \times 100 \tag{4.2}$$

$$ROM_D(\%) = \frac{\sum_{n=0}^{M} \Delta\theta_3(n)}{\sum_{n=0}^{M} \Delta\theta_{3\ ref}(n)} \times 100 \tag{4.3}$$

$$ROM_L(\%) = \frac{\sum_{n=0}^{M} \Delta\theta_4(n)}{\sum_{n=0}^{M} \Delta\theta_{4\ ref}(n)} \times 100 \tag{4.4}$$

With these values it is calculated a ROM for the extensions and flexions movements (ROM_F and ROM_E)by averaging the previous values:

$$ROM_F(\%) = \frac{ROM_A + ROM_L}{2} \tag{4.5}$$

$$ROM_E(\%) = \frac{ROM_D + ROM_M}{2} \tag{4.6}$$

The final ROM of the performed therapy (ROM_P) is calculated by averaging ROM_F and ROM_E:

$$ROM_P(\%) = \frac{ROM_F + ROM_E}{2} \tag{4.7}$$

4.2. Results

All the tests has been carried out in the right knee of the exoskeleton in a tree steeps duration therapy using the maximum velocity per step ($2s$), a total testing time of $6s$ per test.

The obtained results for each test are shown in Table 4.1 and the following Figures. Summarizing, the obtained results show that the patient always follows the pattern of the desired reference trajectory, but when different control modes are used, the trajectory can be considerably modified. When the low impedance control mode is used a high variability of the ROM values is appreciated arriving to values such as 157.5% or 39.5% depending on the scenario. On the other hand, trajectory control mode ensures a much more control scenario in which the patient will always maintain between the ROM values between 111.6% and 66.57% in the worst of the scenarios. High and medium impedance control modes are intermedium modes that stay between ROM ranges of [115.3% − 50.63%] and [130.3% − 47.27%] respectively.

- **No torque**: See Figures 4.3, 4.4 and 4.5. The results under this test show this is the most controlled scenario with a ROM_P of 94.32% for the trajectory control, 86.2% for the high impedance control, 75.54% for the medium impedance control and 67.52% for the low impedance control. It is observed a reduction in the ROM_P while reducing the level of impedance even if the patient is not interacting with the exoskeleton. This occurs due to joint friction forces and structure inertia that generate undesired torques.

- **Passive torque**: See Figures 4.6, 4.7 and 4.8. Since in this test it is simulated when he patient leaves the legs soft an opposite torque to the movement is produced by the legs of the patient during therapy. Due to this event the results show a reduction of the ROM_P. Obtaining with the trajectory control mode a ROM_P of 75.1%, 62.29% with the high impedance control, 51.59% with the medium impedance control and 40.12% with the low impedance control. A reduction respect when no torque is applied of a 20.37% in the case of the trajectory control, 27.73% with the high impedance control, 31.7% for the medium impedance control and 40.58% whith the low impedance control.

- **Favorable force**: See Figures 4.9, 4.10 and 4.11. In this test it is simulated when the patient generates a positive favorable force applied in direction of the motion. Due to this event the results show a increase of the ROM_P. Obtaining with the trajectory control mode a ROM_P of 111.6%, 115.3% with the high impedance control, 130.7% with the medium impedance control and 157.5% with the low impedance control. A ROM_P increase respect when no torque is applied of a 18.32% in the case of the trajectory control, 33.75% with the high impedance control, 73.02% for the medium impedance control and 137.7% with the low impedance control.

- **Opposite force**: See Figures 4.12, 4.13 and 4.14. In this test it is simulated when the patient generates a force applied in opposite direction of the motion. Due to this event the results show a high decrease of the ROM_P. Obtaining with the trajectory control mode a ROM_P of 66.57%,50.62% with the high impedance control, 47.27% with the medium impedance control and 39.5% with the low impedance control. A ROM_P decrease respect when no torque is applied of a 29.42% in the case of the trajectory control, 41.27% with the high impedance control, 37.42% for the medium impedance control and 41.5% with the low impedance control.

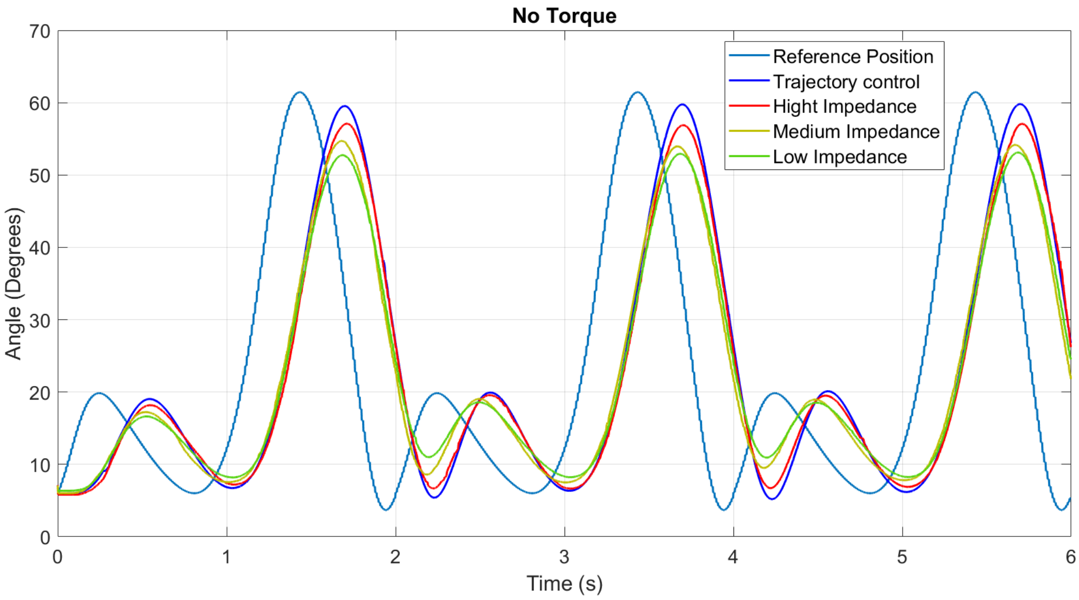


Figure 4.3: Performance when no torque is applied during therapy.

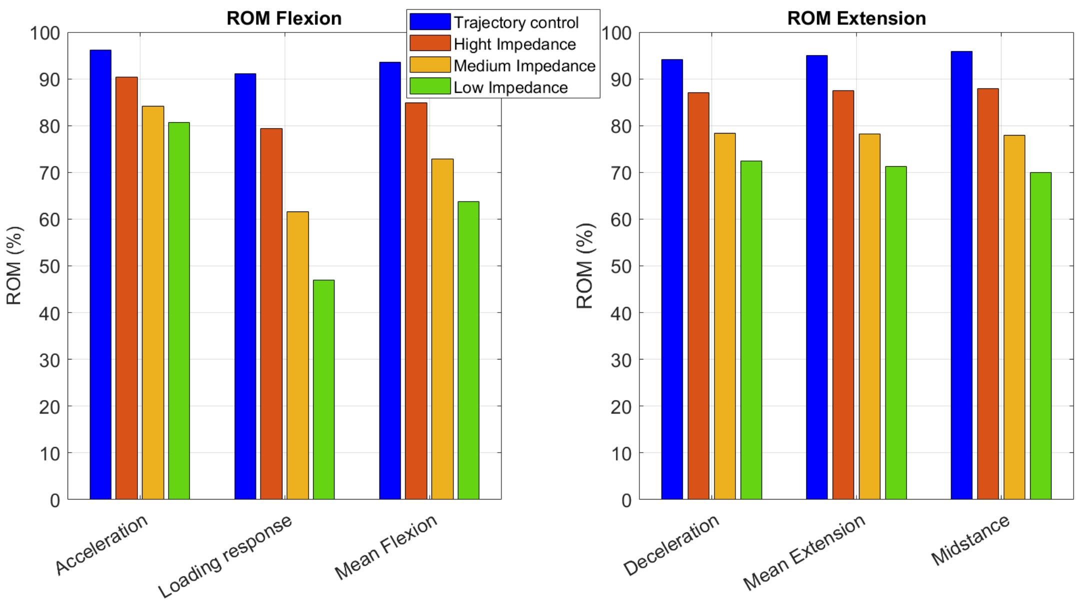


Figure 4.4: Flexions and extensions ROMs under no torque.

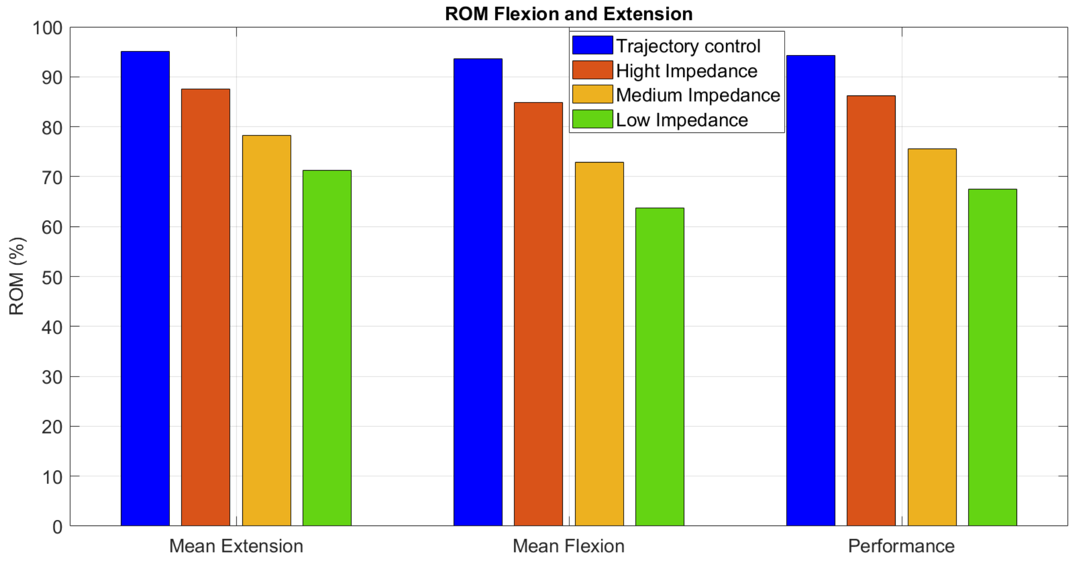


Figure 4.5: Mean flexion, mean extension and final performance ROMs under no torque.

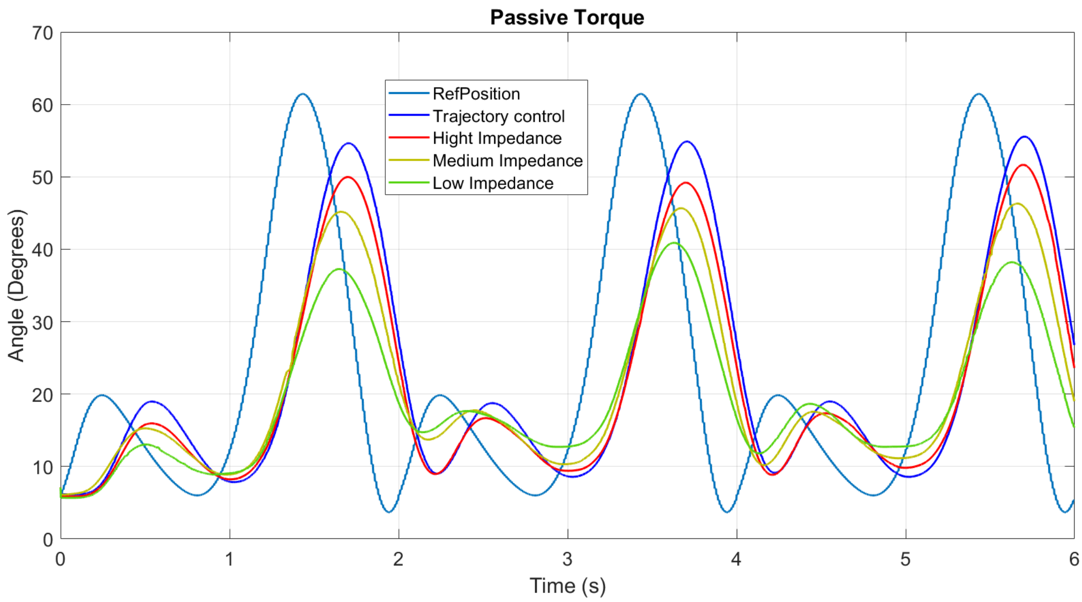


Figure 4.6: Performance when passive torque is applied during therapy.

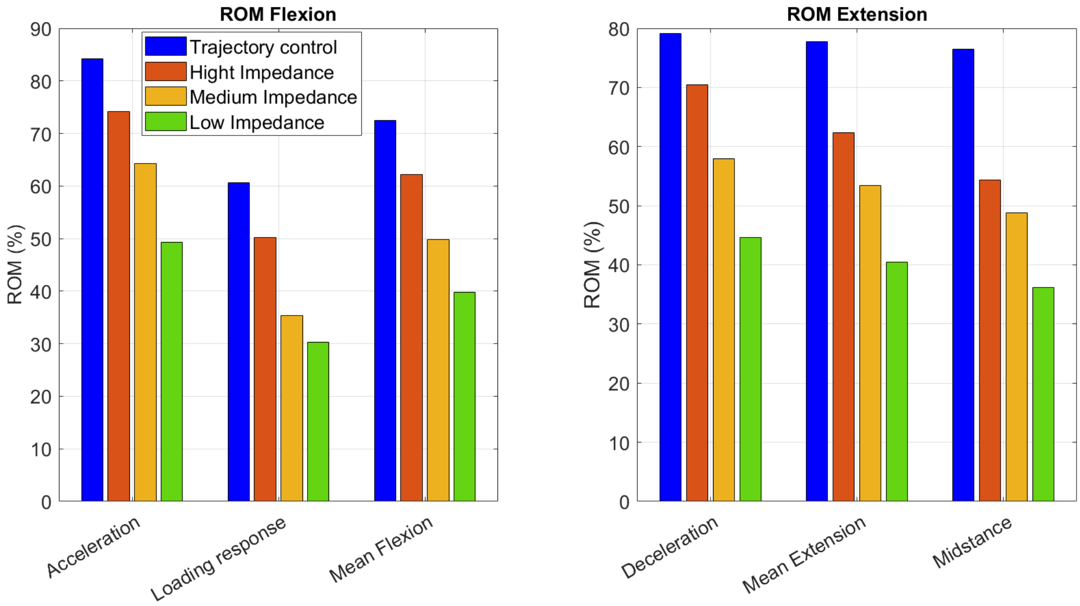


Figure 4.7: Flexions and extensions ROMs under passive torque.

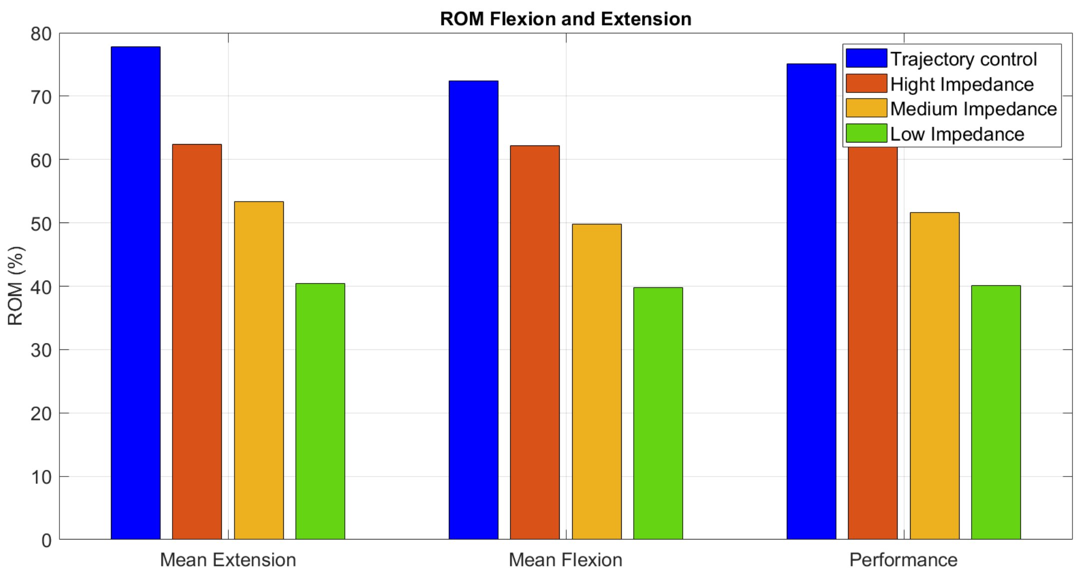


Figure 4.8: Mean flexion, mean extension and final performance ROMs under passive torque.

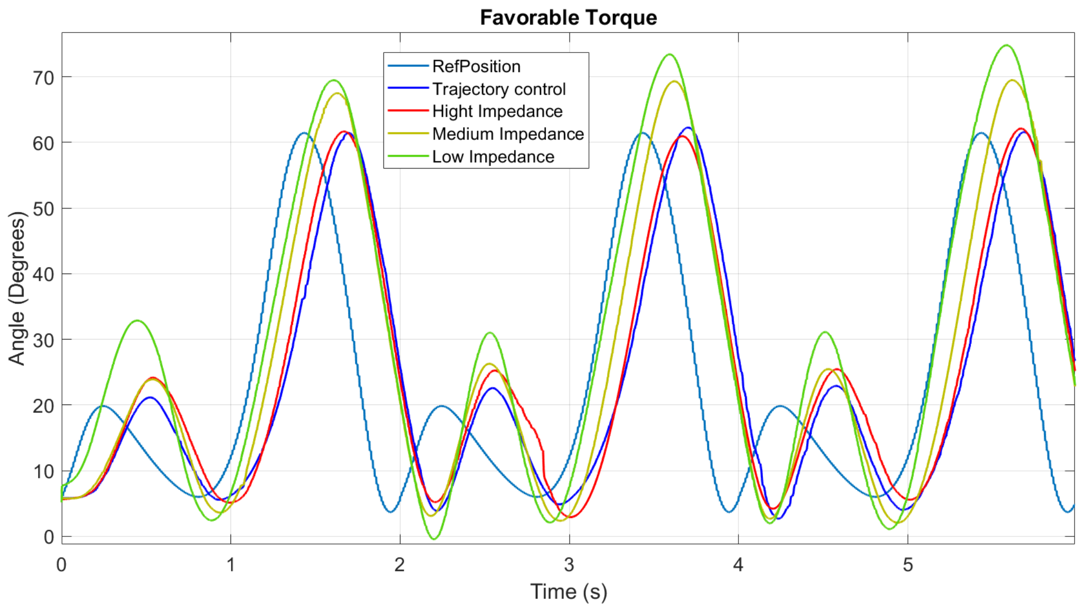


Figure 4.9: Performance when favorable torque is applied during therapy.

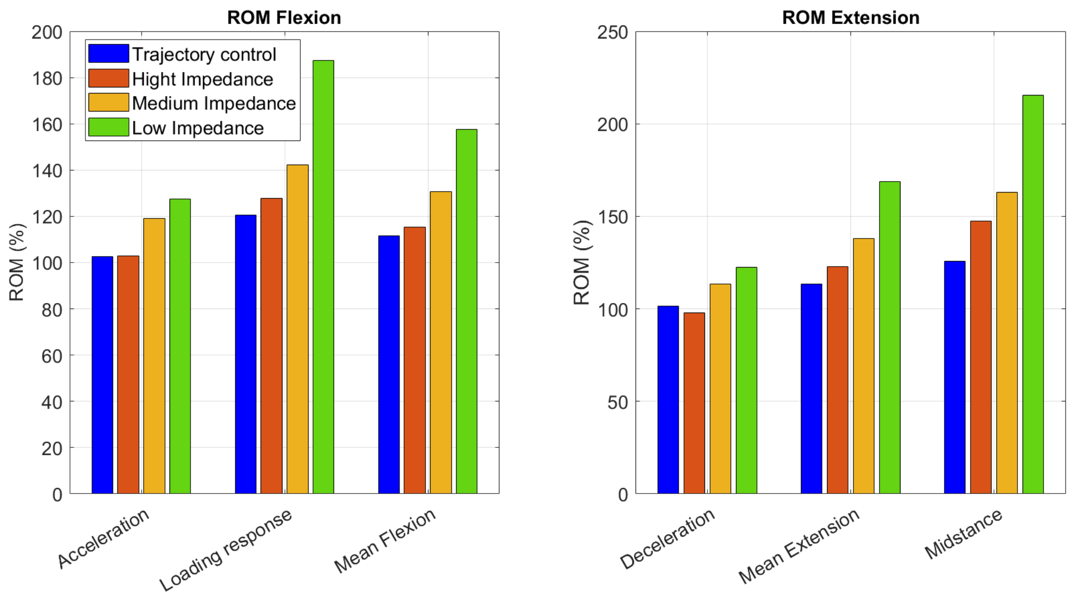


Figure 4.10: Flexions and extensions ROMs under passive torque.

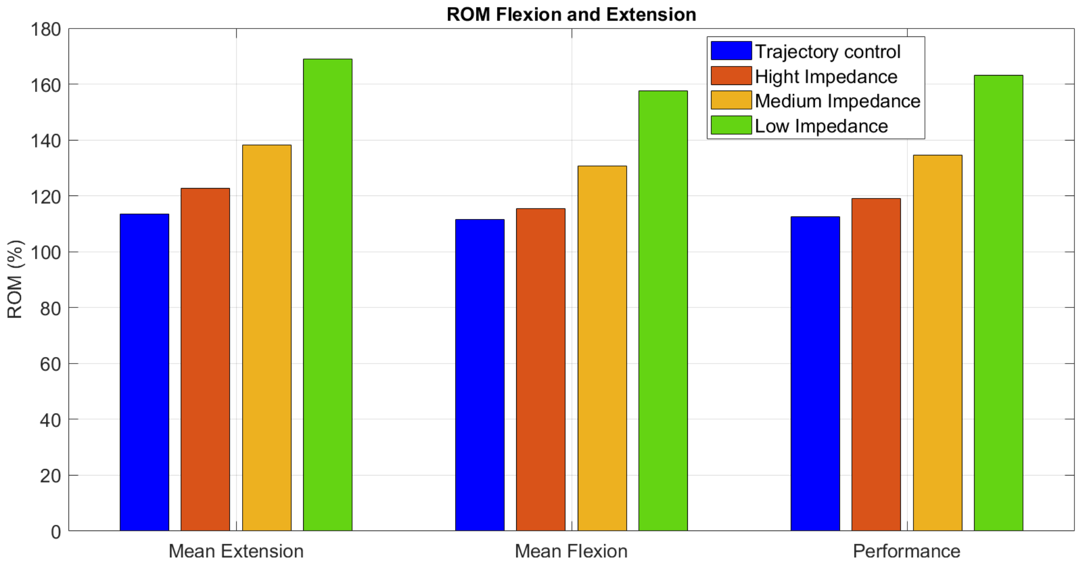


Figure 4.11: Mean flexion, mean extension and final performance ROMs under favorable torque.

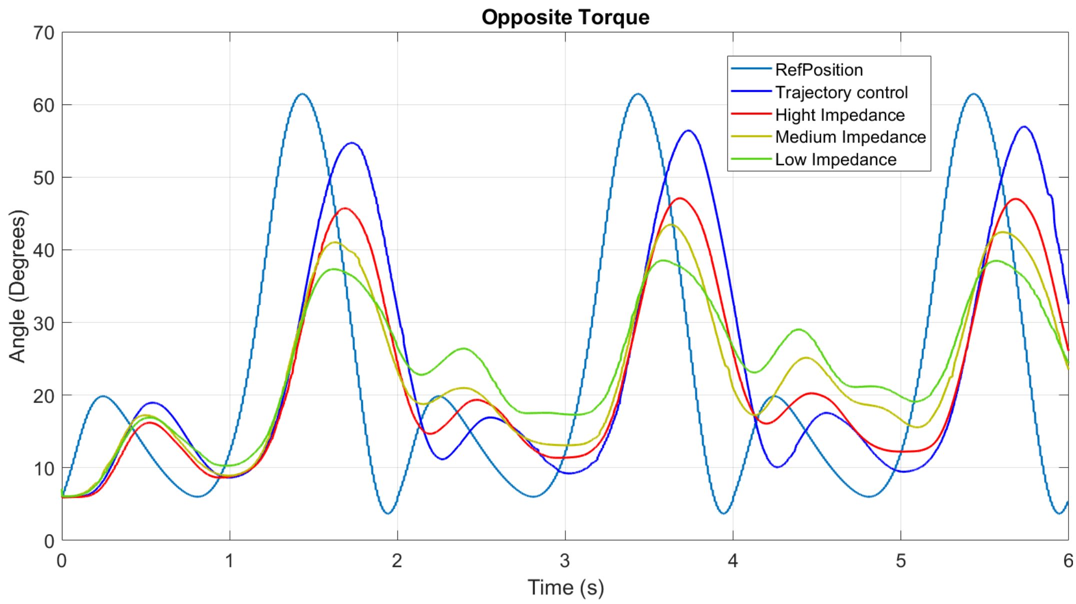


Figure 4.12: Performance when opposite torque is applied during therapy.

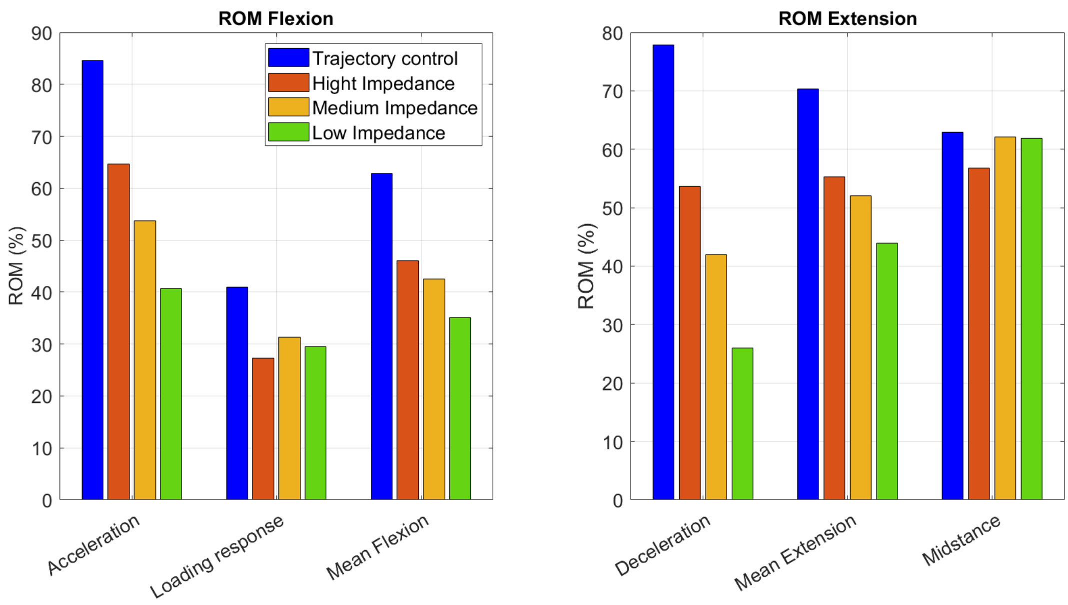


Figure 4.13: Flexions and extensions ROMs under opposite torque.

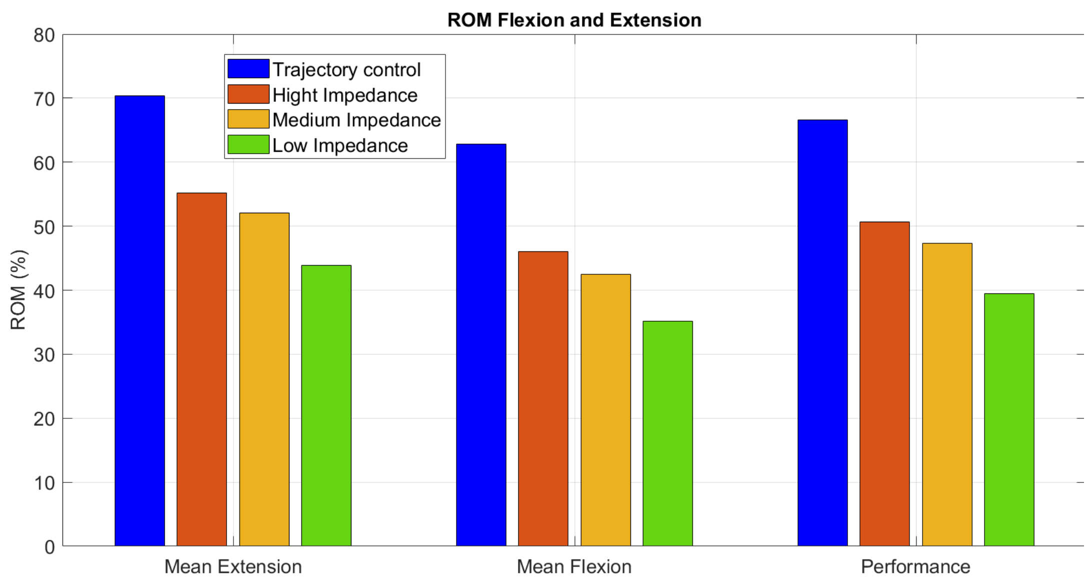


Figure 4.14: Mean flexion, mean extension and final performance ROMs under opposite torque.

TORQUE	MODE	ROM FLEXION (%)			ROM EXTENSION(%)			ROM(%)
		Acceleration	Loading	Mean	Deceleration	Midstance	Mean	PERFORMANCE
NO TORQUE	Trajectory control	96.12	91.13	93.62	94.1	95.94	95.02	**94.32**
	High Impedance	90.38	79.43	84.9	87.07	87,91	87.49	**86.2**
	Medium Impedance	84.17	61.57	72.87	78.4	78	78.2	**75.54**
	Low Impedance	80.65	46.92	63.79	72.49	70.01	71.25	**67.52**
PASSIVE	Trajectory control	84.23	60.64	72.44	79.11	76.41	77.26	**75.1**
	High Impedance	74.19	50.22	62.2	70.43	54.32	62.38	**62.29**
	Medium Impedance	64.24	35.36	49.8	57.97	48.81	53.39	**51.59**
	Low Impedance	49.33	30.26	39.79	44.67	36.22	40.45	**40.12**
FAVORABLE	Trajectory control	102.7	120.5	111.6	101.4	125.7	113.6	**111.6**
	High Impedance	102.9	127.8	115.3	98.01	147.6	122.8	**115.3**
	Medium Impedance	119.2	142.3	130.7	113.4	163	138.2	**130.7**
	Low Impedance	127.5	187.5	157.5	122.4	215.4	168.9	**157.5**
OPPOSITE	Trajectory control	84.6	40.59	62.69	78.81	62.88	70.35	**66.57**
	High Impedance	64.68	27.35	46.01	53.71	56.75	55.23	**50.62**
	Medium Impedance	53.72	31.29	42.51	41.96	62.12	52.04	**47.27**
	Low Impedance	40.68	29.54	35.11	25.94	61.84	43.89	**39.5**

Table 4.1: ROM results for the performed tests.

Chapter 5

Conclusions and future work

5.1. Goals achieved

The main objective of this Master Thesis was to design, implement and evaluate a modular control system for the CPWalker robotic exoskeleton. With the designed and implemented system it has been obtained a modular node based on ROS that communicates with the other modules of the CPWalker System and performs the desired control of the four joints of the exoskeleton.

After the technical validation, it has been observed that the designed system implemented in the right knee of the exoskeleton works in the desired way under several scenarios. Concluding that the designed control provides different levels of human-robot interaction that will be used in the evolution of the rehabilitation therapies of the patients.

Therefore, according to the secondary objectives aforementioned in Chapter 1, the archived objectives are the following:

- Design and software development of the communications with the rest of the robotic platform using ROS.
- Design and software development of the hardware communications and data processing to control the exoskeleton.
- Design and software development of the trajectory and impedance control modes.

Since the technical validation has just been performed in the right knee of the exoskeleton, the last secondary objective "*Implementation and technical validation of the control system in the four joints of the CPWalker*" is considered partially archived.

5.2. Challenges and Future developments

This Master Thesis has designed a control system that allows the control of the CPWalker robotic platform and communicates with the rest of the system modules. This initial work leaves some future work lines emerge:

- The most immediate future line of work is the implementation and technical validation of the control system in the four joints of the CPWalker Platform.
- A clinical validation of the control system with real patients and physiotherapists has to be undertaken to obtain feedback from the users experiences and clinical professionals suggestions.
- Design a user interface to allow easy control of the system.
- Dynamic calculation of the ROM during therapies so the clinical professionals can obtain analytic information from the measured data.

References

[1] H. Kerr Graham, Peter Rosenbaum, Nigel Paneth, Bernard Dan, Jean-Pierre Lin, Diane L. Damiano, Jules G. Becher, Deborah Gaebler-Spira, Allan Colver, Dinah S. Reddihough, Kylie E. Crompton, and Richard L Lieber. Cerebral palsy.

[2] Cerebralpalsy.org. Gross motor function classification system (gmfcs). `https://cerebralpalsy.org.au/our-research/about-cerebral-palsy/what-is-cerebral-palsy/severity-of-cerebral-palsy/gross-motor-function-classification-system/`. [Online; accessed 05-11-2020].

[3] James R Gage, Tom F Novacheck, et al. An update on the treatment of gait problems in cerebral palsy. *Journal of Pediatric Orthopaedics Part B*, 10(4):265–274, 2001.

[4] Madeformovement. Nf-walker. `https://www.madeformovement.com/nf-walker`. [Online; accessed 05-11-2020].

[5] Madeformovement. Innowalk. `https://www.madeformovement.com/innowalk?hsCtaTracking=85b700d2-c13e-488b-869e-58afb7b460d6%7C6ac1ed1c-e2a6-4ee9-9620-d93f26d86d46`. [Online; accessed 06-11-2020].

[6] Hocoma. Locomat. `https://www.hocoma.com/solutions/lokomat/`. [Online; accessed 05-11-2020].

[7] REHA STIM. Gait trainer gt ii. `https://reha-stim.com/gt-ii/`. [Online; accessed 05-15-2020].

[8] WALKBOT. Walkbot. `http://walkbot.co.kr/introduction/`. [Online; accessed 05-15-2020].

[9] Cristina Bayón, Teresa Martín Lorenzo, Beatriz Moral, Óscar Ramírez, Álvaro Moreno, Sergio Lerma Lara, Ignacio Caballero, and Eduardo Rocon. A robot-based gait training therapy for pediatric population with cerebral palsy: goal setting, proposal and preliminary clinical implementation. 01 2018.

[10] Yuya Maruyama, Shinpei Kato, and Takuya Azumi. Exploring the performance of ros2. pages 1–10, 10 2016.

[11] Ricardo Mello. Cpwalker internal network. `https://github.com/ricardocmello/cpwalker-docs/wiki/CPWalker-internal-network`. [Online; accessed 05-20-2020].

[12] BlanchardJ. Finite impulse respond. `https://en.wikipedia.org/wiki/Finite_impulse_response#/media/`. [Online; accessed 05-21-2020].

[13] Harmonic Drive. Strain wave gear principle. `https://www.harmonicdrive.net/technology`. [Online; accessed 05-25-2020].

[14] Cburnett. Typical spi bus: master and three independent slaves. `https://en.wikipedia.org/wiki/Serial_Peripheral_Interface#/media/`. [Online; accessed 06-05-2020].

[15] Lindy N M Gommans, Annemieke T Smid, Marc R. M. Scheltinga, Ernst Cancrinus, Frans A M Brooijmans, Kenneth Meijer, and Joep A. W. Teijink. Prolonged stance phase during walking in intermittent claudication. *Journal of Vascular Surgery*, 66:515–522, 2017.

[16] Peter Rosenbaum, Nigel Paneth, Alan Leviton, Murray Goldstein, Martin Bax, Diane Damiano, Bernard Dan, and Bo Jacobsson. A report: The definition and classification of cerebral palsy april 2006. *Developmental medicine and child neurology.Supplement*, 109:8–14, 03 2007.

[17] Organisation mondiale de la santé, World Health Organization, and World Health Organization Staff. *International classification of functioning, disability and health: ICF*. World Health Organization, 2001.

[18] Else Odding, Marij E. Roebroeck, and Hendrik J. Stam. The epidemiology of cerebral palsy: Incidence, impairments and risk factors. *Disability and Rehabilitation*, 28(4):183–191, 2006. PMID: 16467053.

[19] Confederación ASPACE. Descubriendo la parálisis cerebral. *Confederación Aspace*, pages 1–69, 2014.

[20] Ellen Wood and Peter Rosenbaum. The gross motor function classification system for cerebral palsy: a study of reliability and stability over time. *Developmental Medicine Child Neurology*, 42(5):292–296, 2000.

[21] Karen W Krigger. Cerebral palsy: an overview. *American family physician*, 73(1):91–100, 2006.

[22] Mindy Lipson Aisen, Danielle Kerkovich, Joelle Mast, Sara Mulroy, Tishya AL Wren, Robert M Kay, and Susan A Rethlefsen. Cerebral palsy: clinical care and neurological rehabilitation. *The Lancet Neurology*, 10(9):844 – 852, 2011.

[23] Frederick B. Palmer, Bruce K. Shapiro, Renee C. Wachtel, Marilee C. Allen, Janet E. Hiller, Susan E. Harryman, Barbara S. Mosher, Curtis L. Meinert, and Arnold J. Capute. The effects of physical therapy on cerebral palsy. *New England Journal of Medicine*, 318(13):803–808, 1988. PMID: 3280999.

[24] Andreas Meyer-Heim and Hubertus J.A. [van Hedel]. Robot-assisted and computer-enhanced therapies for children with cerebral palsy: Current state and clinical implementation. *Seminars in Pediatric Neurology*, 20(2):139 – 145, 2013. Update on Cerebral Palsy: Diagnostics, Therapies and the Ethics of it All.

[25] Cristinia Bayon, Rafael Raya, S Lerma Lara, Oscar Ramírez, J Serrano, and Eduardo Rocon. Robotic therapies for children with cerebral palsy: a systematic review. *Transl Biomed*, 7(1):44, 2016.

[26] Andreas Meyer-Heim, Corinne Ammann-Reiffer, Annick Schmartz, Jan Schaefer, Felix H Sennhauser, Florian Heinen, Beat Knecht, E Dabrowski, and Ingo Borggraefe. Improvement of walking abilities after robotic-assisted locomotion training in children with cerebral palsy. *Archives of disease in childhood*, 94(8):615–620, 2009.

[27] Susan E Fasoli, Barbara Ladenheim, Joelle Mast, and Hermano Igo Krebs. New horizons for robot-assisted therapy in pediatrics. *American journal of physical medicine rehabilitation*, 91(11):S280–S289, 2012.

[28] N Smania, M Gandolfi, V Marconi, A Calanca, C Geroin, S Piazza, P Bonetti, P Fiorini, A Consentino, C Capelli, et al. Applicability of a new robotic walking aid in a patient with cerebral palsy. *Eur J Phys Rehabil Med*, 147(2):135–40, 2011.

[29] Cristina Bayon, Oscar Ramírez, M Dolores Del Castillo, José Ignacio Serrano, Rafael Raya, José M Belda-Lois, R Poveda, F Mollà, Teresa Martin, Ignacio Martínez, et al. Cpwalker: Robotic platform for gait rehabilitation in patients with cerebral palsy. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3736–3741. IEEE, 2016.

[30] Herman van der Kooij, Bram Koopman, and Edwin HF van Asseldonk. Body weight support by virtual model control of an impedance controlled exoskeleton (lopes) for gait training. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1969–1972. IEEE, 2008.

[31] MakeltFrom.com. 7075 (alzn5.5mgcu, 3.4365, 2l95, a97075) aluminum. `https://www.makeitfrom.com/material-properties/7075-AlZn5.5MgCu-3.4365-2L95-A97075-Aluminum`. [Online; accessed 05-11-2020].

[32] Raspberrypi.org. Raspberry pi 4. `https://www.raspberrypi.org/products/raspberry-pi-4-model-b/`. [Online; accessed 05-30-2020].

[33] ROS.org. Robot operating system. `https://www.ros.org/`. [Online; accessed 05-30-2020].

[34] ROS.org. Ros introduction. `http://wiki.ros.org/ROS/Introduction/`. [Online; accessed 05-20-2020].

[35] ROS.org. Nodes. `http://wiki.ros.org/Nodes`. [Online; accessed 05-20-2020].

[36] ROS.org. Understanding ros master. `http://wiki.ros.org/Master`. [Online; accessed 05-20-2020].

[37] ROS.org. Topics. `http://wiki.ros.org/Topics`. [Online; accessed 05-20-2020].

[38] ROS.org. Services. `http://wiki.ros.org/Services`. [Online; accessed 05-22-2020].

[39] ROS.org. Network setup. http://wiki.ros.org/ROS/NetworkSetup. [Online; accessed 05-20-2020].

[40] ROS.org. Multiple machnes. http://wiki.ros.org/ROS/Tutorials/MultipleMachines, note = [Online; accessed 05-20-2020].

[41] L. Litwin. Fir and iir digital filters. *IEEE Potentials*, 19(4):28–31, 2000.

[42] S. Jezernik, G. Colombo, and M. Morari. Automatic gait-pattern adaptation algorithms for rehabilitation with a 4-dof robotic orthosis. *IEEE Transactions on Robotics and Automation*, 20(3):574–582, June 2004.

[43] Maxon. *EC 60 flat 68 mm, brushless, 100 Watt*, 5 2018.

[44] Z. Li, W. W. Melek, and C. Clark. Decentralized robust control of robot manipulators with harmonic drive transmission and application to modular and reconfigurable serial arms. *Robotica*, 27(2):291–302, 2009.

[45] Harmonic Drive SE. Engineering data csg-2a component sets. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj_2KCBo9vpAhWPOeAKHXAyB5sQFjAAegQIAhAB&url=https%3A%2F%2Fharmonicdrive.de%2Ffileadmin%2Fuser_upload%2F2014_11_ED_1019654_CSG-2A.pdf&usg=AOvVaw0RrccJsocTF3f-4Ag53ka8. [Online; accessed 05-30-2020].

[46] MathWorks. System identification toolbox. https://es.mathworks.com/products/sysid.htm. [Online; accessed 05-20-2020].

[47] Shahid Hussain, Sheng Quan Xie, and Guangyu Liu. Robot assisted treadmill training: mechanisms and training strategies. *Medical engineering & physics*, 33(5):527–533, 2011.

[48] Gery Colombo, Matthias Joerg, Reinhard Schreier, Volker Dietz, et al. Treadmill training of paraplegic patients using a robotic orthosis. *Journal of rehabilitation research and development*, 37(6):693–700, 2000.

[49] Daisuke Aoyagi, Wade E Ichinose, Susan J Harkema, David J Reinkensmeyer, and James E Bobrow. A robot and control algorithm that can synchronously assist in naturalistic motion during body-weight-supported gait training following neurologic injury. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(3):387–400, 2007.

[50] W. R. Evans. Control system synthesis by root locus method. *Transactions of the American Institute of Electrical Engineers*, 69(1):66–69, 1950.

[51] Andreas Meyer-Heim, Corinne Ammann-Reiffer, Annick Schmartz, Jan Schaefer, Felix H Sennhauser, Florian Heinen, Beat Knecht, E Dabrowski, and Ingo Borggraefe. Improvement of walking abilities after robotic-assisted locomotion training in children with cerebral palsy. *Archives of disease in childhood*, 94(8):615–620, 2009.

[52] Britta Husemann, Friedemann M̈uller, Carmen Krewer, Silke Heller, and Eberhardt Koenig. Effects of locomotion training with assistance of a robot-driven gait orthosis in hemiparetic patients after stroke: a randomized controlled pilot study. *Stroke*, 38(2):349–354, 2007.

[53] Neville Hogan. Impedance control: An approach to manipulation: Part i—theory. 1985.

[54] Neville Hogan. Impedance control: An approach to manipulation: Part ii—implementation. 1985.

[55] ADVANCED MOTION CONTROLS. *AZBH12A8 AxCent^{TM} pcb mount servo motor drive*, 04 2016.

[56] ADVANCED MOTION CONTROLS. Duty cycle (open loop). `https://www.a-m-c.com/experience/technologies/operating-mode/duty-cycle-open-loop/`. [Online; accessed 05-30-2020].

[57] ANALOG DEVICES. *True Accuracy, 16-Bit $\pm$ 12 V/ $\pm$ 15 V, Serial Input Voltage Output DAC.*

[58] Wikipedia.org. Serial peripheral interface. `https://en.wikipedia.org/wiki/Serial_Peripheral_Interface/`. [Online; accessed 06-05-2020].

[59] Mike McCauley. C library for broadcom bcm 2835 as used in raspberry pi. `https://www.airspayce.com/mikem/bcm2835/`. [Online; accessed 01-12-2020].

[60] Wikipedia. Range of movement. `https://en.wikipedia.org/wiki/Range_of_motion`. [Online; accessed 06-07-2020].

[61] David Feil-Seifer and Maja J Mataric. Defining socially assistive robotics. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 465–468. IEEE, 2005.

[62] D. Feil-Seifer and M. J. Matarić. Socially assistive robotics. *IEEE Robotics Automation Magazine*, 18(1):24–31, 2011.

[63] Marie Kruse, Susan Ishøy Michelsen, Esben Meulengracht Flachs, HENRIK BRØNNUM-HANSEN, Mette Madsen, and Peter Uldall. Lifetime costs of cerebral palsy. *Developmental Medicine & Child Neurology*, 51(8):622–628, 2009.

Appendix A

Impact

A.1. Ethical

This work developed in this Master Thesis is included in the framework of "Assistive Robotics" (AR). AR describes a group of robot that assist people with physical disabilities through physical interaction [61]. These robots aims to address areas and gaps in care by automating supervision, motivation, and companionship aspects of one-on-one interactions with individuals from various large and growing populations [62]. Specifically, this work is included under the area of health care wearable robotics (exoskeleton) used for rehabilitation.

AR and health care robots have started several ethical, social and philosophical discussions, the most relevant are exposed:

- **Labour replacement**: Mainly related with the well known social discussion about if robots are designed to solve problems or save money by replacing human care. In this case the robot is a tool of the clinical professional to facilitate the session and the clinical professional are always needed.

- **Moral**: This issue refers to the modifications in the quality of the given services when a robot undertake the labor of a human ("cold-care"). Also there are discussions about if robots are capable of moral reasoning and deal with ethical problems. Since the clinical professional is needed during the hole therapy it is continuously checking the correct performance of the system and interactuating with the patient.

- **Responsibility and trust**: This discussion questions who is responsible for the robot actions, the robot or the human? Since robots get more and more autonomous, human care givers are less in charge of the processes and so patients must trust the devices.

- **Privacy and data protection**: Issue related with what data is collected during therapies, how is stored, who has access to it and how is it used. The collected data from the therapies is only used by the engineers and clinical professionals to analyze it and keep track of the patient evolution.

A.2. Social and Economical

Cerebral palsy constitutes the most common cause of physical impairment in children with an overall prevalence of around 2 per 1000 live births [17]. The life time costs associated with cerebral palsy in the United States in the 2000s was about 11500 million$, 800,000$ per patient. This estimation underscores the need for effective primary and secondary prevention measures to prevent and reduce the impact of the disease [63].

Aforementioned, the present Master Thesis is part of the CPWalker robotic platform project included in the area of health care robotics for rehabilitation. CPWalker's objective is to improve quality of life and autonomy of cerebral palsy patients by performing early and novel robotic therapies to prevent patients from worsening. These early therapies during the development stages of patients' life allows a reduction of end impairments produced by cerebral palsy increasing quality of life and reducing life time costs.

Appendix B

Economical Budget

This Master Thesis has been developed in collaboration with the "Consejo Superior de Investigaciones Científicas" (CSIC) which has provided the necessary resources to carry out this project. A budget is calculated based on the needed human resources and materials to carried out the project.

- **Human Resources**: This item considers the salary of the engineering student, author of this Master Thesis, see Table B.3.

	Cost per hour (€)	Hours	Total (€)
Engineering student	20	450	9,000
TOTAL			**9,000**

Table B.1: Human resources costs.

- **Materials**: This item considers the costs of the used materials during the development of this Master Thesis (see Tabla B.2).

	Lifespa (years)	U.	Cost (€)	Depreciation (€/month)	Time used (months)	Total (€)
Structure	10	1	1000	8.33	3	25
HarmonicDrive	10	1	1230	10.25	3	30.75
Maxon-408057	10	1	207	1.72	3	5.18
DriverAZBH12A8	10	1	191,91	1.6	3	4.79
PCM3910DCDC	5	1	130.19	2.16	3	6.5
DAC-5570	10	1	34.17	2.6	3	7.8
dsPIC30F4011	5	1	4.84	0.08	3	0.24
Raspberry Pi4	1	2	31	0.25	4	2
Sensors	10	1	70	0.58	3	1.75
Power supply	10	1	60	0.5	3	1.5
MATLAB	1	1	2000	166.67	3	500
TOTAL						**585.51**

Table B.2: Meterial costs

Therefore, taking into account human resources costs and materials costs the final economical budget is the following:

	Coste
Human Resources	9,000€
Material cots	585.51€
Subtotal	9,585€
IVA	2,012.85€
Total	11,597.85€

Table B.3: Total costs.

Appendix C

Developer manual

This Appendix attempt to help to understand the work carried out in this Master Thesis so new collaborators can continue with future work.

- **Connection with the Raspberry Pi and PC**: The operations with the raspberries are done using the Secure Shell (SSH) network protocol so it is suggested to use a linux Operating System in the PC controller. Since the raspberries' IPs are static it is just needed to connect the PC to the same local internet network as the raspberries to enable the communication.

- **ROS**: To continue this work it is needed to install ROS 1 in your operating system and to have a basic knowledge about it and the programming lenguage `c++` to understand the sytem and the communications between the different nodes, it is suggested to take a look at `http://wiki.ros.org/ROS/Tutorials`.

  - **Control node**: Since the developed ROS node (control node) runs in the Raspberry Pi Worker all the hardware controlled with this node must be attached to it not to the other Raspberry (PiMaster).

  - **Environment variables**: To enable the ROS network it is needed to set the ROS environmental variables "ROS_MASTER_URI and ROS_HOSTNAME and ROS_IP in the " /.bashrc" of all machines. The PiMaster works as ROS MASTER of the network so the network will not work if the PiMaster is not running or the other machines does not recognize it. The Raspberry IPs are fixed, and it is suggested not to change them, in case you, do the ROS Network variables must also be changed accordingly to the new IPs.

- **SPI**: Aforementioned, the system uses the SPI bus to communicate with the actuators. The Raspberry Pi is equipped with two SPI busses disable by default, to enable them ensure the line "dtparam=spi=on" is not commented out in "/boot/config.txt" file. The SPI of the raspberry has up to three CS channels, since this system needs four CS (one for each joint) the bcm2835 library is used to control normal GPIOs as CS channels. Take into account that the developed SPI object satisfies the SPI requirements of the used hardware (DAC5570) but may not satisfy others form other devices.

- **Executing the program**: Since the program is written in the control node, to execute the program it is needed to run the node by using the command "roslaunch cpwalker joint_control.launch" in the shell. Be sure the acquisition node and processing node are already running in the PiMaster before launching the control node, since the PiMaster acts as the ROS MASTER the network will not work if it is not working. Also, since the program uses physical ports of the Raspberry Pi (SPI) it requires from specific permissions to control this ports, be sure to give the necessary privileges to the executed program by using: "$sudo chown root::root name_of_executable", "$sudo chmod a+rx name_of_executable" and "$sudo chmod u+s name_of_executable".

- **Data collected**: The program collects the potentiometer, torque and voltage ata measured and sent of each session. This information is overwritten in text files in the "/Documents" folder of the PiWorker.